Predictive Modeling for Spatio-Temporal Analysis of Urban Sensing Data

Thesis submitted

by

Asif Iqbal Middya

DOCTOR OF PHILOSOPHY (Engineering)

Department of Computer Science and Engineering

Faculty Council of Engineering and Technology

Jadavpur University

Kolkata, West Bengal-700032, India

2024

Predictive Modeling for Spatio-Temporal Analysis of Urban Sensing Data

Thesis submitted

by

Asif Iqbal Middya

DOCTOR OF PHILOSOPHY (Engineering)

under the supervision of

Prof. Sarbani Roy

Department of Computer Science and Engineering

JADAVPUR UNIVERSITY

Kolkata, West Bengal-700032, India

2024

Index No: 102/19/E

1. **Title of the Thesis:**

   PREDICTIVE MODELING FOR SPATIO-TEMPORAL ANALYSIS OF URBAN SENSING DATA

2. **Name, Designation and Institution of the Supervisor:**

   Dr. Sarbani Roy
   Professor
   Department of Computer Science and Engineering
   Jadavpur University
   Kolkata-700032

3. **List Of Publication:**

   (a) **Journal:**

      i. **A.I. Middya**, S. Roy, J. Dutta, and R. Das, 2020. Jusense: a unified framework for participatory-based urban sensing system. Mobile Networks and Applications, 25, pp.1249-1274. (DOI: https://doi.org/10.1007/s11036-020-01539-x)

      ii. **A.I. Middya**, S. Roy, S. Mandal and R. Talukdar, 2021. Privacy protected user identification using deep learning for smartphone-based participatory sensing applications. Neural Computing and Applications, 33, pp.17303-17313. (DOI: https://doi.org/10.1007/s00521-021-06319-6)

      iii. **A.I. Middya**, and S. Roy, 2024. Truthful double auction based incentive mechanism for participatory sensing systems. Peer-to-Peer Networking and Applications. (DOI: https://doi.org/10.1007/s12083-024-01681-3)

      iv. **A.I. Middya**, and S. Roy, 2021. Spatial interpolation techniques on participatory sensing data. ACM Transactions on Spatial Algorithms and Systems, 7(3), pp.1-32. (DOI: https://doi.org/10.1145/3457609)

      v. **A.I. Middya**, and S. Roy, 2022. Spatio-temporal variation of Covid-19 health outcomes in India using deep learning based models. Technological Forecasting and Social Change, 183, p.121911.
      (DOI: https://doi.org/10.1016/j.techfore.2022.121911)

      vi. **A.I. Middya**, and S. Roy, 2021. Geographically varying relationships of COVID-19 mortality with different factors in India. Scientific reports, 11(1), p.7890. (DOI: https://doi.org/10.1038/s41598-021-86987-5)

   (b) **Conference:**

      i. **A.I. Middya**, and S. Roy, 2022, December. Predicting Missing Value for Ozone Data Based on Optimal Site-Selection Strategy. In 2022 IEEE Calcutta Conference (CALCON) (pp. 288-292). IEEE.
      (DOI: https://doi.org/10.1109/CALCON56258.2022.10060138)

(c) **Book Chapter:**

i. Middya, A.I., Dey, P. and Roy, S., 2022. IoT-based crowdsensing for smart environments. In Internet of Things for Smart Environments (pp. 33-58). Cham: Springer International Publishing. (DOI: https://doi.org/10.1007/978-3-031-09729-4_3)

4. **List of Patents:**

(a) Patent filed for "High-granular pollution mapping based less polluted path recommendation using Participatory sensing". Patent Application No. 202331064959 dated 27-09-2023.

5. **List of Presentations in International Conference:**

(a) **A.I. Middya**, and S. Roy, 2022, December. Predicting Missing Value for Ozone Data Based on Optimal Site-Selection Strategy. In 2022 IEEE Calcutta Conference (CALCON) (pp. 288-292). IEEE.
(DOI: https://doi.org/10.1109/CALCON56258.2022.10060138)

Statement of Originality

I, **Asif Iqbal Middya** registered on **30th May 2019**, having registration number- **1021904007**, do hereby declare that this thesis entitled "**Predictive Modeling for Spatio-Temporal Analysis of Urban Sensing Data**" contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 4%.

Asif Iqbal Middya.

Signature of Candidate

Date: 02/04/2024

Sarbani Roy 02/04/2024

Professor
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032

Dr. Sarbani Roy,

Professor,

Department of Computer Science and Engineering,

Jadavpur University

(Supervisor)

Certificate from the Supervisor

*This is to certify that the thesis entitled "**Predictive Modeling for Spatio-Temporal Analysis of Urban Sensing Data**" submitted by **Asif Iqbal Middya**, who got his name registered on 30th May 2019, for the award of Ph.D. (Engineering) degree of Jadavpur University, is absolutely based upon his own work under the supervision of Dr. Sarbani Roy and that neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.*

Sarbani Roy 02/04/2024

Professor
Computer Sc. & Engg. Department
Jadavpur University
Kolkata-700032

Dr. Sarbani Roy,
Professor,
Department of Computer Science and Engineering,
Jadavpur University
(Supervisor)

Dedicated

To Sarbani Roy Madam and My Family

Acknowledgements

This thesis is the eagerly anticipated result of my perseverance and that of numerous others who helped me finish my Doctor of Philosophy in Engineering degree at Jadavpur University. Without the expert guidance of my supervisor, assistance from my fellow researchers, and support from my family, who have been an essential part of this journey, I would never have been able to complete my dissertation.

Foremost, I extend my heartfelt gratitude to my mentor and supervisor, Professor Sarbani Roy, from the Department of Computer Science and Engineering at Jadavpur University. My appreciation for her knows no bounds, as she not only paved the way for this doctoral journey but also offered steadfast support, unwavering encouragement, exceptional guidance, nurturing care, and boundless patience. Her mentorship has been pivotal in steering me towards the successful completion of this thesis. Professor Roy's insightful observations and constructive feedback have not only set the course for my research but have also propelled me to delve deeper into my investigations. I have been incredibly fortunate to have an advisor who encouraged independent exploration while offering invaluable guidance to refine my ideas.

I wish to express my sincere appreciation to the members of the Research Advisory Committee (RAC), Professor Nandini Mukherjee and Dr. Chandreyee Chowdhury, who diligently assessed my progress every six months during my doctoral studies.

I extend my deep thanks to the Heads of the Department, Professor Mahantapas Kundu and Professor Anupam Sinha, along with Professor Nandini Mukherjee, from the Department of Computer Science and Engineering at Jadavpur University, for granting me access to the DST-FIST lab. I am also grateful for the unwavering support of the entire faculty and non-teaching staff within our department.

I sincerely acknowledge the UGC-NET JRF Fellowship scheme (UGC-Ref. No.: 3684 / (NET-JULY 2018)) for providing the financial support that enabled me to pursue my Ph.D. from May 2019 to February 2024.

It has been a delightful and enriching experience to work within the DSTFIST Lab at the Department of Computer Science and Engineering, Jadavpur University. The congenial working atmosphere has been a great boon, and I wish to express my sincere appreciation to all my labmates and friends from DST-FIST lab. In particular, I would like to extend my

gratitude to Benay da, Tanmoy da, Joy da, Joyeeta di, Moumita di, Priya di, Paramita di, Srimoyee di, Sounak, Subhayan, Rituparna, Anwesha, Manjarini, Anrarlina, Soumyadeep, and Rohit for their camaraderie, cooperation, and valuable discussions during our collaborative efforts.

I am deeply thankful to all my esteemed teachers, from my school years to my university tenure, for their guidance during critical junctures in my academic journey. My friends, spanning from school to university, have been a wellspring of inspiration and unwavering support.

Last but not least, I extend my heartfelt gratitude to my family members who have stood by my side through all the challenges in my life. In particular, I am indebted to my sister, Sona, for her constant support, which has allowed me to focus more intently on my work.

Asif Iqbal Middya.

Jadavpur, Kolkata — Asif Iqbal Middya

Date: 02/04/2024

Contents

1 Introduction **1**
1.1 Overview . . . 1
1.2 Research Issues . . . 2
1.3 Motivation . . . 4
1.4 Objective . . . 5
1.5 Contribution . . . 6
1.6 Organization of the Thesis . . . 8

2 Literature Review **13**
2.1 Urban Dynamics and Societal Applications . . . 13
2.2 Traditional Sensing . . . 14
2.3 Participatory Sensing . . . 14
2.4 Challenges of Participatory Sensing . . . 16
2.4.1 Privacy Preservation . . . 16
2.4.2 Task Allocation and Incentive Mechanism Design . . . 16
2.4.3 Data Quality . . . 18
2.5 Predictive Techniques for Spatio-temporal Analysis . . . 19
2.6 Summary . . . 20

3 Unified Framework Design for Participatory Sensing Applications **23**
3.1 Introduction . . . 23
3.1.1 Objective . . . 24
3.1.2 Contribution . . . 24
3.2 Existing Works . . . 25
3.3 Problem Statement . . . 26
3.4 JUSense: Urban Sensing Applications . . . 27
3.4.1 Subsystems Overview . . . 28
AirSense . . . 28
NoiseSense . . . 30
SenseDCity . . . 31
D&RSense . . . 31
3.4.2 Design and Architecture . . . 32
Participatory Sensing Layer . . . 33
Computation Layer . . . 34
3.4.3 Pre-processing . . . 36

Context-aware data cleaning . . . 36
Calibration . . . 36
Map-matching . . . 37
3.4.4 Data fusion . . . 38
3.4.5 Interaction among Components . . . 43
3.5 Evaluation . . . 43
3.5.1 Study Area and Data Collection . . . 44
3.5.2 Data preparation . . . 46
3.5.3 Performance metrics . . . 47
3.5.4 Results . . . 47
Performance Comparison with Baseline . . . 50
Case studies . . . 51
3.6 Summary . . . 55

4 Privacy Aware User Identification for Smartphone-based Sensing 57
4.1 Introduction . . . 57
4.1.1 Objective . . . 58
4.1.2 Contribution . . . 58
4.2 Existing Works . . . 60
4.3 Problem statement . . . 61
4.4 Proposed methodology . . . 61
Sensor data . . . 61
Data preparation . . . 62
Feature extraction . . . 64
Architecture of Deep CNN . . . 65
4.5 Evaluation . . . 66
4.5.1 Experimental setup . . . 66
4.5.2 Baselines and Evaluation metrics . . . 67
4.5.3 Results and discussion . . . 68
4.6 Summary . . . 70

5 Double Auction based Incentive Mechanism 71
5.1 Introduction . . . 71
5.1.1 Objective . . . 72
5.1.2 Contribution . . . 73
5.2 Existing Works . . . 74
5.3 Problem Statement . . . 75
5.3.1 System Model . . . 75
5.3.2 Problem Formulation . . . 79
Task Providers . . . 79
Task Executors . . . 79
Auctioneer . . . 80
5.4 Truthful Double Auction Based Incentive Mechanism for Task Allocation . . . 82

5.4.1 Details of TATA . . . 82
5.4.2 A Walk-Through Example . . . 84
5.4.3 Proofs of Desirable Properties . . . 87
5.5 Evaluation Results . . . 94
5.5.1 Experimental Setup . . . 94
5.5.2 Analysis of Results . . . 94
5.5.3 Performance Evaluation . . . 97
Performance metrics . . . 97
Comparison with other incentive mechanisms . . . 98
Impact of ϕ and N_{max} on auction outcome . . . 100
5.6 Summary . . . 102

6 Predictive Modeling and Spatio- temporal Analysis of Environment Data 103
6.1 Introduction . . . 103
6.1.1 Objective . . . 104
6.1.2 Contribution . . . 105
6.2 Existing Works . . . 105
6.2.1 Inverse distance weighting method . . . 106
6.2.2 Kriging . . . 107
6.3 Problem statement . . . 108
6.4 Proposed Approach . . . 109
6.4.1 Participatory sensing based framework . . . 109
Sensing and data collection . . . 109
Pre-processing . . . 110
Noise map as a Service . . . 111
6.4.2 Urban noise level measurement . . . 111
6.4.3 Context aware data cleaning . . . 111
6.4.4 GPS to MGRS Convertion . . . 112
6.4.5 GLIDW . . . 113
6.4.6 I-GLIDW . . . 115
6.4.7 GLIDW-OK . . . 117
6.4.8 Performance metrics . . . 118
6.5 Experiment and results . . . 119
6.5.1 Study Area and Data Collection . . . 119
6.5.2 Calibration . . . 122
6.5.3 Parameters and model selection . . . 123
6.5.4 Results . . . 124
Performance Comparison with baselines . . . 126
Discussions and observations . . . 128
6.6 Summary . . . 131

7 Mining Spatio-temporal Variation in the Performance of the Models 133
7.1 Introduction . . . 133
7.1.1 Objective . . . 134
7.1.2 Contribution . . . 134
7.2 Existing Works . . . 135
7.3 Problem statement . . . 136
7.4 Proposed approach . . . 136
Dataset description . . . 136
Data pre-processing and batching . . . 137
Deep learning models . . . 138
Implementation of deep learning models . . . 141
7.5 Evaluation . . . 143
7.5.1 Experimental setup . . . 143
7.5.2 Evaluation metrics . . . 143
7.5.3 Spatial variability in the models' performance . . . 143
7.5.4 Spatial distribution of optimal models . . . 145
7.5.5 Discussion . . . 147
7.6 Summary . . . 148

8 Geographically varying Spatio-temoral relationships of COVID-19 health outcomes 149
8.1 Introduction . . . 149
8.1.1 Objective . . . 150
8.1.2 Contribution . . . 150
8.2 Existing Works . . . 150
8.3 Problem Statement . . . 151
8.4 Proposed Approach . . . 151
8.4.1 Data description . . . 151
Datasets . . . 151
8.4.2 Modeling spatial relationship . . . 154
8.5 Results . . . 156
8.5.1 Performance of OLS and GWR Model . . . 157
8.5.2 Geographically varying relationships between COVID-19 deaths and the driving factors . . . 158
8.5.3 Discussion . . . 160
8.6 Summary . . . 163

9 Conclusion 165
9.1 Major Outcomes . . . 165
9.2 Limitations and Future scope . . . 166

List of Figures

1.1 A taxonomy of contributions. The shaded portions indicate the area of research where the thesis work best fits in terms of its contributions. 8

3.1 Workflow of JUSense from the end-users' viewpoint. 27
3.2 User interface of data logging activities for four applications (a) NoiseSense (b) AirSense (c) SenseDCity and (d) D&RSense. User interface of (e) air pollution map service and (f) noise pollution map service. 29
3.3 Architecture of the JUSense system 33
3.4 Multiview data fusion: The pollution level of the POI P_j (shown in green) at time T_k is predicted using spatial view and temporal view. 40
3.5 Formation of Overall Pollution Index (OPI). 40
3.6 Interaction among different components of the subsystems in JUSense 41
3.7 Study Area: Jadavpur area, Kolkata 45
3.8 (a) Air quality monitoring device (AQMD) (b) A participant is using AirSense application along with AQMD (c) A participant is using NoiseSense application 45
3.9 Calibration for NoiseSense for different Smartphones with respect to standard sound level meter (SLM). Here, the ideal line shows a line where the smartphone responses and corresponding SLM measurements should agree. . 46
3.10 Air pollution map of Jadavpur area, Kolkata. 48
3.11 Noise pollution map of Jadavpur area, Kolkata. 48
3.12 Diurnal variation of AQI and PM10. 49
3.13 Results of D&RSense (a) Rash driving prone zone (b) Normal driving (speed $\leq$ 40 kmph) trajectory of a volunteer. (c) Rash driving prone road segment. . 50
3.14 Results of SenseDCity (a) Common health issues among the volunteers based on location of food intake (b) Most popular locations in Jadavpur region. . . 51
3.15 The road conditions reported by SenseDCity based on citizens' feedback. . . 51
3.16 Overall pollution map of Jadavpur area, Kolkata. 52
3.17 Less polluted route between a source (JU Gate No. 1) and a destination (JU Gate No. 4). (a) Less air pollution route (the right most route) (b) Less noise pollution route (two right most routes). 53
3.18 Performance comparison of the models. (a) RMSE values of the models for air pollution level prediction. (b) MAE values of the models for air pollution level prediction. (c) Classification accuracy of the models for driving events detection. 53

3.19 (a) Bad road conditions detected by SenseDCity (b) Bad road conditions detected by D&RSense (c) Combining the results of both SenseDCity and D&RSense in map (d) Performance comparison with varying dataset size. . . 54
3.20 (a) Frequent travel path (blue line) of an user (b) Noise levels along the frequent travel path given by NoiseSense (c) User health database maintained by SenseDCity (d) Notification by SenseDCity 54

4.1 A taxonomy of the user identification / authentication methods. The shaded portion indicates the category in which the proposed work best fits. 60
4.2 A framework describing overall process of the proposed approach of user identification. The framework consists of major steps such as accelerometer data collection, pre-processing, windowing and feature extraction, train-test splits, model building, and evaluation 62
4.3 The magnitude spectrum from the Fourier transform of two typical accelerometer signals from two users. (a) acceleration vs time plot of the first user (b) magnitude spectrum of accelerometer signal of the first user (c) acceleration vs time plot of the second user (d) magnitude spectrum of accelerometer signal of the second user. 63
4.4 Architecture of the proposed CNN model. It consists of four convolutional layers, two max-pooling layers, two dropout layers, two fully connected layers, and one output layer. 64
4.5 The plots of training and validation accuracy versus number of epochs for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users. 66
4.6 The plots of training and validation loss versus number of epochs for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users. 67
4.7 Performance comparison of the proposed model (CNN) with the baselines (LR, SVM, DT, RF, KNN) in terms of four metrics (accuracy, precision, recall, and F1 measure) for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users. 68

5.1 A market of sensing task allocation between TPs and TEs. TP_i, TE_k, and R_j represent i^{th} task provider, k^{th} task executor and j^{th} *ROI* respectively. (a) TPs have sensing tasks at different *ROIs*. The TEs are able to execute the tasks at different *ROIs*. (b) The TPs place bids to the auctioneer for those *ROIs* where the sensing tasks need to be performed. The TEs place asks (offer price for executing the tasks) to the auctioneer for those *ROIs* where they can perform sensing tasks. Auctioneer decides the price charged to the winning TPs and the incentive/ reward for the winning TEs. 75
5.2 Three major stages of TATA: TATA-CTD, TATA-STD and TATA-PC. 81

5.3 Bipartite graph between the set of task executors and the set of task providers. TPs are arranged in non-increasing order of the submitted bids and TEs are arranged in non-decreasing order of the submitted asks. The asks less than the threshold asks (i.e. $A_k^j < A_\phi$) and the bids greater or equal to the threshold asks (i.e. $D_i^j \geq A_\phi$) are shown within rounded rectangles. The blue and the red edges represent the candidate trades ($\mathscr{T}$) and non-candidate trades respectively. 88
5.4 Bipartite graph between the candidates (executors and providers) selected by Algorithm 2. The blue and the red edges represent the successful trades ($\mathscr{S}^{\mathscr{T}}$) and unsuccessful trades respectively. 88
5.5 A pictorial summary of the overall working process of TATA for the walk-through example of Section 5.4.2. TP_i^j and TE_k^j indicate non-zero bids and asks respectively. From the possible trades, TATA-CTD first identifies a set of candidate trades. Next, TATA-STD finds a set of successful trades from the set of candidate trades. Finally, TATA-PC calculates the price charged to the winning TPs and payment/ incentive/ reward given to the winning TEs. . 88
5.6 Verification of truthfulness property when (a) task executors win task providers by placing truthful asks (b) task executors lose in the auction by placing truthful asks (c) task providers win task executors by placing truthful bids (d) task providers lose in the auction by placing truthful bids 95
5.7 Compare the (a,c,e) bids and prices of winning TPs for different threshold settings and (b,d,f) payment and ask of winning TEs for different threshold settings. 96
5.8 The normalized value of price collected from TPs and payments given to TEs. 97
5.9 Execution time of TATA when (a) number of TEs increases from 100 to 1000 and (b) number of TPs increases from 20 to 200. 97
5.10 Task Providers' satisfaction when (a) number of task executors increases and (b) number of providers increases. Task Executors' satisfaction when (c) number of task executors increases and (d) number of providers increases. . . 99
5.11 Total utility of task executors when (a) number of task executors increases and (b) number of providers increases. Total utility of task providers when (c) number of task executors increases and (d) number of task providers increases. 100
5.12 Comparison of total utility of task executor for TATA and TATA w/o EE (TATA with out executor elimination) when (a) number of task executors increases and (b) number of providers increases. 101
5.13 The impact of ϕ and N_{max} on the outcome of TATA. (a) Number of successful trades vs ϕ (b) Total utility of task executors vs ϕ (c) Total utility of task providers vs ϕ. The dotted vertical lines indicate the optimal value of ϕ at which the best performance of TATA is achieved. 102

6.1 Participatory sensing based framework 110
6.2 Normalized mean total acceleration and proximity sensor behavior for hand and pocket contexts for five different subjects 113

6.3 Process of I-GLIDW (a) Prediction of grid cells' noise having a sufficient number of noise samples. Small dots represent sample points. The samples within each circle are responsible to find the noise value of the grid cell within that circle (b) A particular grid cell is zoomed in. dB_i are the known noise values at sample points (square dots) having distances d_i for $i = \{1,2,3,4,5\}$ from the midpoint (circular dot) of the grid cell. (c) First iteration of I-GLIDW (d) Prediction of an unpredicted grid cell's noise using already predicted neighbors (shaded grid cells) in an iteration 117
6.4 Map of our study area. This map is created using QGIS. The study area is part of city Kolkata, India. Here, Easting and Northing are East-West and North-South positions respectively of MGRS coordinate value. 120
6.5 Demographic data of participants. 120
6.6 Spatial distribution of the sampling locations in the study area for the morning session (MS), afternoon session (AS), evening session (ES), data collected over the two-year period, weekdays, and weekends. Different colors in the map indicate sampling locations corresponding to different participants. 121
6.7 (a) Screenshot of data logging activity using our Android application (NoiseSense). (b) Noise pollution map as a service. 121
6.8 The details of the soatio-temporal data and the approach for evaluation of the spatial noise mapping techniques. 122
6.9 Scatter plots of SLM readings and cottesponding smartphone readings for three different smartphomes (a) Google Pixel (b) Moto Z2 Play (c) Panasonic Eluga. A regression line (red line) shows linear relationship between smartphone noise level measurements and true noise levels. The difference between regression line (red line) and ideal line (cyan) gives calibration offset ξ. 124
6.10 Prediction accuracy (RMSE values) for different combinations of neighboring threshold N_{Th} and cell dimension n. 125
6.11 Fitting of four commonly used semivariogram models (Spherical, Exponential, Gaussian and Linear) to the experimental semivariogram 125
6.12 Noise map produced by method GLIDW. Grid cells having an insufficient number of sample points remain unpredicted (white-colored regions). 125
6.13 Noise maps produced in different iterations ($I_1 - I_{15}$) by method I-GLIDW . . 126
6.14 Noise map produced by method GLIDW-OK 126
6.15 Observation vs Prediction plots of different methods (GLIDW, I-GLIDW, GLIDW-OK, IDW, OK) in different sessions (MS-Morning Session, AF-Afternoon Session, ES-Evening Session) of a day 127
6.16 Execution time (T_E) and communication time (T_c) for different methods by varying the number of data samples n_s. (a) T_E and T_c with $n_s \approx 4000$ (b) T_E and T_c with $n_s \approx 8000$ (c) T_E and T_c with $n_s \approx 12000$ 129
6.17 Relative Improvements between all possible pairs of methods in different sessions of a day 130

7.1 Overall framework for exploring spatio-temporal variability in the performance of deep learning based approaches for estimating state-level COVID-19 mortality in India. 136
7.2 Spatial distribution of state-level COVID-19 death counts. 137
7.3 Model architecture for state-level COVID-19 death count forecasting models (a) CNN (b) LSTM Autoencoder (c) RNN (d) Vanilla LSTM (e) Bi-Directional LSTM. 141
7.4 Spatial distribution of MAE values for different models. (a) LSTM Auto-Encoder (b) Bi-LSTM (c) CNN (d) vanilla LSTM (e) RNN 144
7.5 Spatial distribution of RMSE values for different models. (a) LSTM Auto-Encoder (b) Bi-LSTM (c) CNN (d) vanilla LSTM (e) RNN 145
7.6 Box and swarm plots of MAE values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks 145
7.7 Box and swarm plots of RMSE values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks 146
7.8 Box and swarm plots of R^2 values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks 146
7.9 Observation versus predicted plots of COVID-19 death count for some of highly covid-affected states (a) West Bengal (b) Maharashtra (c) Karnataka (d) Tamil Nadu (e) Delhi (f) Uttar Pradesh (g) Punjab (h) Andhra Pradesh . . 146
7.10 Geographical distribution of the best performing models for different forecast horizons (a) one week (b) two weeks (c) three weeks 147

8.1 Geographical distribution of COVID-19 deaths across India. The spatially continuous distribution map is generated in QGIS (https://qgis.org/en/site/) by using Inverse Distance Weighting (IDW) interpolation. 152
8.2 Stepwise variable selection for geographically weighted regression (GWR). . 156
8.3 Scatter plot of the observed and the predicted COVID-19 death count using (a) global OLS model and (b) GWR model. 158
8.4 Geographical distribution of R^2 values for geographically weighted regression (GWR) model. 159
8.5 Local parameter estimates of geographically weighted regression (a) Intercept (b) HH_With_TCMC (c) $Higher_Edu$ (d) $Tot_population$ (e) $PM2.5$ and (f) Age_Abv_50. 160

List of Tables

1.1 A summary of the thesis structure in terms of focus areas, objectives, contributions, and publication 10

2.1 Existing Literature 21

3.1 Individual and combined services provided by JUSense 44

4.1 A summary of existing literature on user identification based on the sensor data. 59
4.2 Features for user identity detection 63
4.3 Model accuracy 69
4.4 Model training time (in seconds) 69

5.1 Smartphone sensors used for different Participatory sensing based applications/ systems 73
5.2 A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics. (✓: The requirement/characteristic has been met, $\times$: The requirement/characteristic has not been met, IC : Incentive Compatibility, IR: Individual Rationality, BB: Budget Balance, CE: Computational Efficiency, LA: Location awareness, HMTA: Heterogeneous Multi-Task Assignment, MAAIM: Mental accounting auction incentive mechanism) 76
5.3 Notation 78
5.4 Ask matrix of 5 task Executors 87
5.5 Bid matrix of 5 Task Providers 87
5.6 Need matrix of 5 Task Providers 87
5.7 Comparison among IMRAL, LBS and TATA (for same dataset) 101

6.1 A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics. 107
6.2 Notation 109
6.3 Linear Models and Quality of Linear Regression 123
6.4 Performance comparison with baselines 127
6.5 Features and Theoretical Comparison among IDW, OK, GLIDW, I-GLIDW and GLIDW-OK 129
6.6 Performance of methods under different data sparsity levels 129

8.1 A summary of datasets 152

8.2 VIFs with all the 11 explanatory variables 153
8.3 VIFs after removing the variable $HH_Abv_8_P$ 153
8.4 Summary of the global model (OLS) for various socioeconomic, demographic, and environmental pollution related factors. 157
8.5 Summary of the local model (GWR) for various socioeconomic, demographic, and environmental pollution related factors. 157
8.6 Performance comparison OLS and GWR models in terms of three performance metrics: AICc, R^2, and Adjusted R^2. 158
8.7 Local R^2 and district-level parameter estimates by geographically weighted regression for some of the districts of India that are severely affected by COVID-19 disease. 161

Abstract

The rapid expansion of urban areas globally necessitates innovative approaches for understanding and managing complex urban dynamics. However, traditional urban monitoring methods relying on sparsely distributed static stations are inadequate to capture the intricate dynamics and fine-grained spatio-temporal variability inherent in complex urban environments. Participatory sensing, leveraging the ubiquitous presence of smartphones equipped with various sensors, presents a compelling alternative for acquiring large-scale, high-resolution data on diverse urban phenomena.

Despite its potential, existing participatory sensing solutions are hampered by several limitations. Notably, the absence of a unified framework hinders seamless integration and data exchange between diverse applications. Additionally, inadequate attention is paid to privacy-aware solutions, potentially compromising user anonymity and trust. Also, lack of fair task allocation techniques exist for participatory / crowd sensing systems which is leading to efficient incentive mechanism. Furthermore, limited analysis of the inherent spatio-temporal variability in urban dynamics restricts the accuracy and generalizability of insights derived from the collected data. Finally, insufficient exploration of the geographically varying relationships between urban dynamics and various risk factors hinders the development of targeted interventions and resource allocation strategies.

This thesis bridges these critical research gaps by introducing novel techniques for studying urban dynamics using participatory sensing data. We propose JUSense, a unified framework that integrates diverse urban sensing applications, fostering seamless data exchange and collaboration. We propose a privacy-preserving user identification module based on deep learning for robust user authentication while safeguarding anonymity. To address data quality concerns, we explore context-aware data cleaning and device-specific calibration mechanisms to ensure data accuracy. Additionally, we propose TATA, a truthful double auction-based incentive mechanism that promotes fair task allocation and motivates user participation in data collection.

To leverage the rich spatio-temporal data collected through participatory sensing, we explore advanced analytical methods. We present a system that enables dynamic noise pollution mapping as a service, employing novel spatial interpolation techniques to address sparse data scenarios and ensure high accuracy. Furthermore, we investigate the spatio-temporal variability of deep learning models for predicting health outcomes, such as COVID-19 cases and mortalities. This analysis allows for the identification of regions where specific models perform exceptionally well, enabling targeted and efficient resource allocation for healthcare interventions. Finally, we delve into the issue of geographical heterogeneity by employing geographically weighted regression models to explore the spatially varying relationships between COVID-19 health outcomes and diverse socio-economic, demographic, and environmental risk factors. This analysis provides valuable insights into the localized factors influencing the pandemic's course and informs targeted interventions tailored to specific geographical regions.

By addressing the critical research gaps, the contributions in this thesis pave the way for the development of robust and efficient urban dynamics analysis systems. These advancements hold significant potential to empower urban planners and policymakers with the necessary insights to make informed decisions, ultimately leading to improved resource management, enhanced urban planning strategies, and a better quality of life for citizens.

Chapter 1

Introduction

1.1 Overview

The last few decades have seen a fast expansion of urban areas worldwide. In light of ongoing trends in population density, coupled with the development of regions with high urban expansion probability, a substantial prediction of the change in worldwide urban land-cover is anticipated [1]. A research states that by 2030, urban land cover would have increased by 1.2 million km^2, approximately three times the worldwide urban land area as of 2000 [2]. This fast expansion of urban areas is raising the need for effective planning and informed decision making in order to enhance the quality of citizen's lives. However, the urban settings exhibit a complex system influenced by various factors/aspects such as air and noise pollution dynamics, health dynamics, road and traffic condition, population dynamics, and beyond. Hence, efficient sensing, analyzing, monitoring, and mapping of these diverse aspects is becoming a challenging task.

Traditionally, urban dynamics are assessed through government setup static monitoring stations sparsely distributed across locations (e.g., streets, railway stations, hospitals, airports, residential, and commercial areas) or through manual monitoring by human experts. However, Internet of things (IoT)-based participatory sensing / crowd sensing has recently grown as a captivating approach for sensing and acquiring data for urban dynamics [3]. Participatory sensing is a process in which a large number of people share data and extract crucial information from it in order to manage, estimate, or infer diverse phenomena of interests [4] [5]. Participatory sensing has become an effective way of sensing urban dynamics due to the widespread availability of smartphones among citizens. Sensors and communication technologies are regularly utilized in smart handheld (e.g., smartphones, tablets, etc.) and wearable devices. These large number of sensors, allowing to acquire a variety of data such as image, audio, video, geo-location, and environmental information. These smart IoT devices could be utilized for effective monitoring of different dynamics namely environmental pollution, health dynamics, traffic and road condition, etc. Participatory sensing data along with artificial intelligence (AI) techniques, offers a promising opportunity for analyzing large-scale, real-world data from citizens, which can help to better understand various urban dynamics.

In the aforementioned context, this PhD thesis aims to investigate predictive techniques for studying urban dynamics using both traditional and participatory sensing data. The focus

is on predictive modeling for spatio-temporal analysis of various urban dynamics, including air pollution, noise pollution, and citizen health. In the subsequent sections of this chapter highlight the gaps or research issues in the existing solutions for exploring participatory-based modeling and spatio-temporal analysis of various urban dynamics. This chapter also describes how those research gaps motivate us to conduct this PhD thesis work. Moreover, our major contributions that bridge the identified gaps in the existing research are also highlighted.

1.2 Research Issues

This section provides a comprehensive overview of research issues and gaps in the literature. There are existing methods that focus on exploring various participatory-based urban sensing applications [3, 6–16]. Chapter 2 provides a detailed description of such previous works. However, previous research exhibits several major shortcomings, including inefficient framework design for participatory sensing, insufficient emphasis on privacy-aware solutions, limited analysis of high-granular spatio-temporal variability in dynamics, and inadequate exploration of the geographical association between urban dynamics and various risk factors. Additionally, the issues like context aware-sensing, calibration, etc are also highlighted. A detailed examination of research issues / gaps in the literature is presented below.

G_1: *Need of a unified framework for participatory sensing-*
Urban sensing systems aim to tackle various urban challenges (e.g., deteriorating air and noise pollution, declining health conditions, transportation system issues, poor road conditions) by leveraging data from diverse city dynamics. Traditionally, separate urban sensing applications are designed to monitor different urban dynamics like environment, transportation, mobility, etc. However, combining these applications to aggregate information can lead to various new inferences. Existing solutions fall short in delivering a unified framework for participatory sensing. It's essential to note that the primary purpose of such a unified framework is to enhance urban sensing applications by mitigating their individual limitations.

G_2: *Less attention on context aware-sensing and calibration mechanisms-*
Existing techniques do not pay sufficient focus on the pre-processing for improving the quality of acquired raw participatory sensing data. We mainly identified two research issues associated with the pre-processing of participatory data.

(a) In participatory sensing data may be generated from different sensing contexts. The acquired data of different dynamics may not provide accurate values due to inappropriate sensing context. This approach should have a proper context-aware data cleaning technique to remove the measurements collected from inappropriate sensing contexts.

(b) Low-cost IoT devices, such as smartphones used in participatory sensing, may produce suboptimal data quality for specific urban dynamics, like ambient pollution levels. In essence, smartphone measurements might not reflect accurate values. To ensure accurate data, it is essential to implement proper calibration mechanisms specific to IoT devices.

(c) In participatory sensing based data collection, sometimes we need to use static stations' data as ground truth to validate the data quality. However, missing values in static stations' data, which are frequently brought on by hardware breakdowns or data corruption, are a serious problem in environmental pollution related research. These missing data affect real-time evaluation participatory data quality and future data processing.

G_3: *Insufficient emphasis on privacy-aware sensing solutions-*
As sharing smartphones with guest users (e.g., members of the family, colleagues, friends, etc.) is a common scenario, it is necessary for the crowd sourcing platforms to identify the actual data provider in smartphone-based crowd / participatory sensing systems. It is because the raw participatory / crowd sensing data may suffer from inaccuracies due to the variety of expertise, reliability, and trust value of users. Under the circumstances mentioned above, it is required to have a privacy protected authentication system that can efficiently, silently and inconspicuously identify the current user (i.e. participatory data provider) from their smartphone sensor data. However, the existing solutions do not pay attention on such a privacy protected authentication system for participatory sensing.

G_4: *Lack of fair task allocation techniques for participatory / crowd sensing systems which is leading to efficient incentive mechanism-*
In a participatory sensing based system, a task provider (also known as a crowdsourcer) may have a set of sensing tasks regarding different dynamics of a city. Here, adequate users' participation is necessary to acquire a sufficient amount of data which is a key factor for the participatory sensing based systems to provide good service quality. But, existing works on sensing task allocation suffer from lack of good incentive mechanisms that are attractive and fair enough to produce high satisfaction for both the task providers and executors.

G_5: *Limited predictive analysis of spatio-temporal variability in urban dynamics-*
There are two major research issues we have found for predictive analysis of spatio-temporal variability in urban dynamics.

(a) The IoT-based participatory sensing approach generates extensive data on diverse urban dynamics. However, data collection depends on participant mobility, resulting in spatially sparse datasets in certain regions. Sparse datasets hinder accurate estimation of urban dynamics at high spatial granularity, impacting overall accuracy. Spatial missing values in participatory sensing data further impede achieving high accuracy. Efficient techniques are required to address sparse

datasets, ensure high accuracy, and improve response time for the analysis and visualization of urban dynamics.

(b) For spatio-temporal predictive analysis, deep learning techniques have emerged as the state of the art in many domains, such as pollution monitoring, urban planning, public health, environmental management, and so on. Even though there are numerous effective deep learning-based models that can handle different problem-specific difficulties, it has been identified that no single ideal model can perform better across all the regions. It is therefore necessary to conduct research to look into the spatiotemporal variation in how well predictive methods for evaluating urban dynamics perform.

G_6: *Inadequate exploration of the geographically varying association between urban dynamics and various risk factors-*
The variation in the pattern of urban dynamics may be related with a number of underlying variables, such as demographic, socio-economic, atmospheric or environmental pollution related risk factors. Past studies mainly utilized the global models to explore such relationships where the relationships are assumed to be constant and independent of location. These models lack in identifying the spatially heterogeneous effects of the risk factors on the outcome of interest.

1.3 Motivation

As highlighted earlier, the rapid expansion of urban areas and the resulting challenges in effective urban planning and decision-making necessitate innovative approaches to understand and manage complex urban dynamics. The motivation behind this research stems from the limitations or gaps observed in current approaches and predictive methodologies. Traditional methods of monitoring urban dynamics, relying on static government setups monitoring stations and manual interventions, are proving insufficient to address the evolving needs of our expanding urban regions. While government monitoring stations and manual interventions have been foundational, they often lack the granularity and also unable to provide a scalable solution for monitoring urban dynamics.

The advent of Internet of Things (IoT)-based participatory sensing provides an effective way of sensing urban dynamics due to the widespread availability of smartphones among citizens. More specifically, in participatory sensing, people and communities use mobile devices (e.g. smartphones, tablets, etc.) to collect and cloud services to store and analyze systematic data in order to explore any phenomena of interest. It provides a dynamic and scalable method for gathering spatial as well as temporal data of multiple modalities, such as textual data (e.g., user reviews about any point of interest), numerical data (e.g., Sensor readings of environmental parameters), images (e.g., urban road condition images), audio (e.g., ambient sounds), video (e.g., surveillance footage), etc. It can produce a huge volume of data that are ideal for spatio-temporal analysis using advanced artificial intelligence (AI) techniques, having a potential for enhanced urban dynamics modelling. The idea to perform

spatiotemporal predictive modelling of participatory sensing data comes from the need to harness this potential for urban planning, resource management, and improvement of QoL (quality of life) of the residents.

Although the recent use of participatory sensing rather than relying solely on traditional approaches has created numerous new opportunities for in-depth analysis, it has also raised critical research issues and gaps that must be addressed in order to effectively analyse participatory sensing data. In this context, the 'Research Issues' section (i.e., Section 1.2) reveals critical issues and gaps in the existing participatory-based solutions. The lack of a unified framework for participatory sensing, insufficient emphasis on privacy-aware solutions, limited analysis of spatio-temporal variability, and inadequate exploration of the geographical associations between urban dynamics and risk factors present compelling challenges. Addressing these research gaps is essential to unleash the full potential of participatory-based urban sensing, monitoring, and methods.

1.4 Objective

The primary objective of this PhD thesis is to come up with efficient techniques to study urban dynamics from participatory sensing data. More specifically, the aim of this thesis is to perform predictive modelling for spatio-temporal analysis of various urban phenomena of interest. In light of the research issues / gaps mentioned in Section 1.2, we have a specific research objective corresponding to each and every research issue / gap.

O_1: To bridge the gap/issue G_1, the objective is to design and implement a unified framework for to integrate IoT-cloud based various applications for participatory sensing.

O_2: To explore pre-processing techniques for participatory sensing data. This objective is divided into two sub-objectives corresponding to the research issues G_2(a), G_2(b) and G_2(c) as follows.

(a) To address the research issue G_2(a), the objective is to create context aware data cleaning technique for IoT-based participatory sensing data collection.

(b) To address the research issue G_2(b), the objective is to develop IoT device specific proper calibration mechanisms.

(c) To address the research issue G_2(c), the objective is to predict missing values in government setup static stations' data.

O_3: To address the research issue G_3, the objective is to develop predictive models for privacy protected user identification / authentication for smartphone based participatory sensing applications.

O_4: To address the research issue G_4, the objective is to design an incentive mechanism for participatory sensing to perform fair allocation of sensing tasks.

O_5: To perform predictive analysis of spatio-temporal variability in urban dynamics. This objective is divided into two sub-objectives corresponding to the research issues G_5(a) and G_5(b) as follows.

(a) To address the research issue G_5(a), the objective is to come up with techniques that can address diverse scenarios in spatial data analysis like sparse datasets, high accuracy, better response time, etc. for participatory sensing data.

(b) To address the research issue G_5(b), the objective is to explore spatio-temporal variability analysis in the performance of deep learning based predictive models for modelling any phenomena of interest.

O_6: To address the research issue G_6, the objective is to explore geographically varying relationships / associations between any phenomena of interest and various risk factors.

1.5 Contribution

This section describes the detailed contribution made in this thesis. Before discussing the specific contributions we made, we first discuss the research areas where we have contributed. The contributions are mainly spread over three major focus areas of research: urban dynamics and societal applications, sensing methodologies, and predictive modelling for spatio-temporal data of urban dynamics. The shaded parts in Figure 1.1 better highlight the broad research areas where we contribute. For instance, although there are various urban dynamics, we consider health dynamics and pollution dynamics such as noise and air pollution to be phenomena of interest. In the area of participatory-based sensing methodology, we mainly focus on four areas of issue and challenge, namely privacy preservation, data quality, incentive mechanisms, and task allocation. Finally, in the area of predictive modelling for spatio-temporal data on urban dynamics, we focus on statistical, machine learning, and deep learning techniques.

The contributions in this thesis work lie in achieving the objectives (O_1 - O_6) that addressed the presented research gaps/issues (G_1 - G_6). To achieve each of the objectives, we have made a set of contributions as given below.

C_1: To achieve the objective O_1, the following contributions are made.

(a) A unified framework called JUSense (Judicious Urban Sensing) is proposed to integrate IoT-cloud based various applications for urban sensing. Four urban problems are considered and their corresponding subsystems are integrated with JUSense.

(b) The framework JUSense is also designed to act as a recommendation system. For example, depending on the users' health profile, a suitable route (less air/noise polluted) which is good for his/her health is recommended.

(c) The proposed system JUSense is evaluated using largescale experiments on real world participatory sensing based dataset collected by participants.

C_2: To achieve the objective O_2, the following contributions are made.

(a) IoT-Device specific calibration mechanism is developed for noise pollution dynamics in smartphone-based crowd / participatory sensing

(b) A classification algorithm is developed that detects the sensing contexts for smartphone based ambient noise pollution data collection. The classification algorithm mainly detects whether the phone is in the pocket or in hand so that noise data collected during inappropriate sensing contexts can be removed.

(c) Efficient predictive models coupled with the site selection methods are developed that can be used to perform the task of missing value imputation from multi-site air pollution time series data. Specifically, a total of six machine learning methods (e.g., Elastic Net, Decision Tree, Random Forest Regressor, LightGBM, ANN, CNN) and three different site selection methodologies (e.g., Radius based, KL divergence, and Cluster based) are explored for predicting missing Values for atmospheric Ozone data.

C_3: To achieve the objective O_3, the following contributions are made.

(a) Deep learning model based on Convolutional Neural Network (CNN) is proposed for privacy protected user identification with smartphone equipped accelerometer sensor data in crowd/participatory sensing systems.

(b) Extensive experiments are performed to evaluate the performance of the proposed CNN model on a real-world benchmark dataset. To validate the effectiveness of the proposed model, it is compared with five baseline methods namely Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbors (KNN).

C_4: To achieve the objective O_4, the following contributions are made.

(a) An effective incentive mechanism is proposed for participatory sensing based systems related to different city dynamics. The incentive mechanism is modeled as a double auction game to address multiple task provider and multiple task executor scenario.

(b) The proposed auction based incentive mechanism TATA (Truthful Double Auction for Task Allocation) is designed in such a way that it satisfies desirable properties such as truthful, budget balanced, individually rational, and computationally efficient.

C_5: To achieve the objective O_5, the following contributions are made.

(a) We have introduced a system that enables the users to get spatio-temporal variability of an ambient noise pollution dynamics as a service.

(b) We have proposed three methods (GLIDW, I-GLIDW, GLIDW-OK) for dynamic noise pollution mapping using spatial interpolation techniques based on participatory sensing data. The results show that our proposed methods (I-GLIDW,

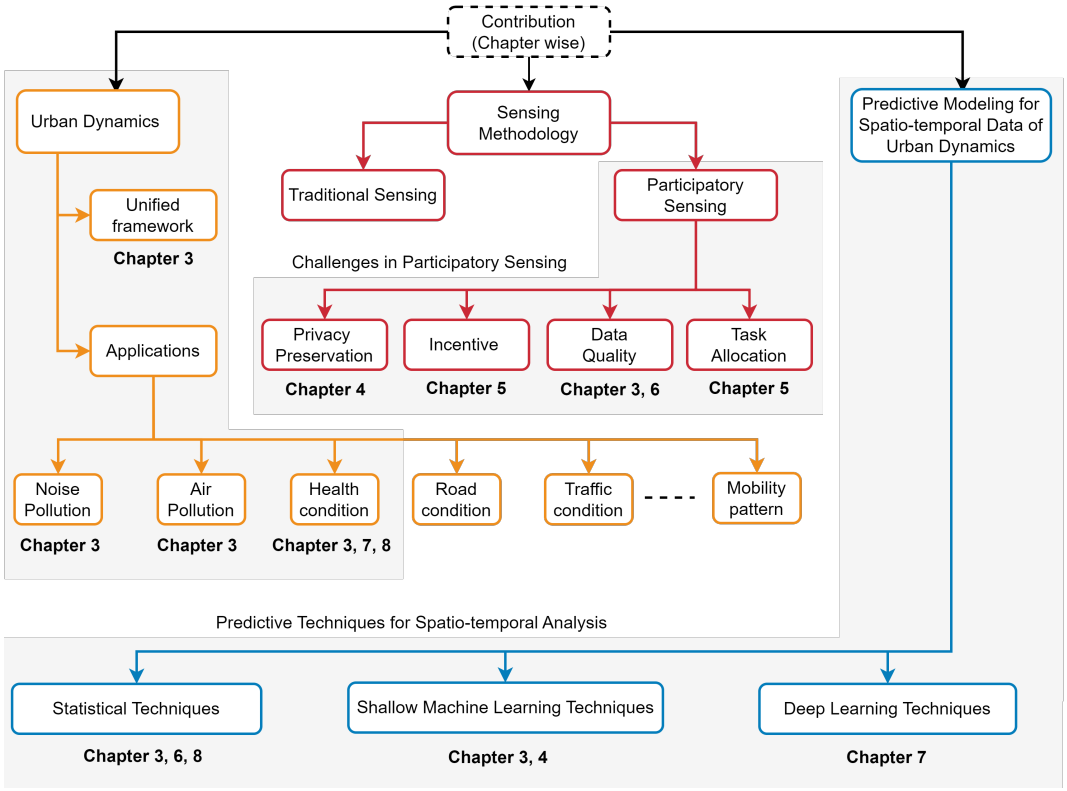


FIGURE 1.1: A taxonomy of contributions. The shaded portions indicate the area of research where the thesis work best fits in terms of its contributions.

GLIDW-OK) perform more accurately compared to the baseline methods (IDW, OK) for participatory sensing noise data.

(c) We have built deep learning based forecasting models for exploring spatio-temporal variation of health dynamics. Specifically, Explore spatio-temporal variability pattern in the performance of deep learning based forecasting models (namely CNN, RNN, Vanilla LSTM, LSTM Autoencoder, and Bidirectional LSTM) for predicting COVID-19 health outcomes in India.

(d) Finding spatial distribution of best-performing models for predicting Covid-19 mortality data for different temporal resolutions (e.g., one week, two weeks, etc).

C_6: To achieve the objective O_6, the following contributions are made.

(a) The potential socio-economic, demographic, and environmental driving factors for COVID-19 health dynamics has been explored.

(b) The geographically varying relationships of COVID-19 health outcome with the driving factors has been investigated by employing local (GWR) model.

1.6 Organization of the Thesis

As previously mentioned, we have pinpointed research issues and gaps (refer to Section 1.2) and established corresponding objectives (outlined in Section 1.4) to tackle these challenges.

The subsequent chapters systematically justify the contributions (refer to Section 1.5) made to realize these objectives. Table 1.1 provides a comprehensive summary of chapter organization, encompassing focus areas, objectives, contributions, and publications. Moreover, Figure 1.1 can also be referred that highlights the chapter wise contributions. The details of the organization of the thesis is as follows:

Chapter 1 presents the problem of predictive modeling for spatio-temporal analysis of participatory sensing data. The chapter commences with an overview of the problem, and proceeds to examine the research issues and gaps within the existing literature that are pertinent to our thematic domain and contribute to the resolution of the identified problem. Following this, the chapter provides a discussion on motivation, objectives, and the specific contributions.

Chapter 2 extensively examines the detailed literature pertaining to diverse urban dynamics and the different sensing techniques, including both traditional and participatory sensing. This chapter comprehensively addresses the challenges and issues insufficiently tackled in prior research related to participatory sensing. Furthermore, it presents a detailed discussion of existing research focusing on predictive modeling for spatiotemporal data associated with urban dynamics.

Chapter 3 presents a unified framework called JUSense (Judicious Urban Sensing) that can derive benefits from these applications by combining their functionalities. It explains how JUSense provides the opportunity for applications to tackle the challenges associated with data collection, aggregation of data in cloud, calibration, data cleaning, and prediction. Further, a multi-view fusion model is proposed for spatiotemporal urban air and noise pollution map generation. This chapter also provides the large scale experiments to evaluate the efficacy of JUSense on real-world dataset.

Chapter 4 introduces technique that attempts to performs privacy-aware user authentication based on their gait patterns (i.e. unique walking patterns). More specifically, a deep Convolution Neural Networks (CNN) model is designed and implemented for the user identification with accelerometer data generated from users smartphone sensors.

Chapter 5 presents a double auction based incentive mechanism called TATA (Truthful Double Auction for Task Allocation) for participatory sensing. The chapter highlights that TATA performs fair allocation of tasks which is leading to efficient incentive mechanism. In the case of TATA, the fair allocation of sensing tasks of the task providers to the task executers indicates that the proposed double auction mechanism is able to satisfy the truthfulness property in order to resist market manipulation (i.e., untruthful bidding and asking). Specifically, the chapter shows that TATA achieves all the desirable properties like individual rationality, truthfulness (i.e. incentive compatibility), budget balance, etc.

Chapter 6 provides a way of using IoT-cloud based framework to generate dynamic (i.e. changes in terms of time and space) noise pollution mapping as a service with the help of spatial interpolation techniques. In this chapter, we have proposed three spatial interpolation techniques (GLIDW, I-GLIDW, GLIDW-OK) that work on participatory sensing based noise pollution data collected using smartphones as IoT devices to generate dynamic noise maps. Proposed techniques can address diverse scenarios like sparse datasets, high accuracy, better

TABLE 1.1: A summary of the thesis structure in terms of focus areas, objectives, contributions, and publication

Chapter	Focus area of the chapter	Objectives (as per Section 1.4) of the chapter	Contributions (as per Section 1.5) of the chapter	Publication
Chapter 1	Problem overview, research issues, motivation, objective, and contribution	-	-	-
Chapter 2	Literature review on urban dynamics, sensing methodologies, and predictive techniques for spatio-temporal analysis	-	-	[17]
Chapter 3	Participatory sensing framework, pollution dynamics, recommendation, context aware data cleaning, calibration mechanism, etc	O_1, O_2(a)(b)	C_1(a)(b)(c), C_2(a)(b)(c)	[18, 19]
Chapter 4	Privacy protected authentication in participatory sensing, deep CNN based modelling	O_3	C_3(a)(b)	[20]
Chapter 5	Incentive mechanism design for participatory task allocation, Game theoretic modelling	O_4	C_4(a)(b)	[21]
Chapter 6	Spatial interpolation techniques for participatory sensing data, Spatio-temporal variability mapping of noise pollution dynamics	O_5(a)	C_5(a)(b), C_2(a)(b)	[22]
Chapter 7	Spatio-temporal variation of Covid-19 health dynamics, deep learning based forecasting models	O_5(b)	C_5(c)(d)	[23]
Chapter 8	Geographically varying relationships of COVID-19 health dynamics with the driving factors, GWR model building for participatory sensing data	O_6	C_6(a)(b)	[24]
Chapter 9	Conclusion and the future directions	-	-	-

response time, etc.

Chapter 7 conducts a case study to examine the spatio-temporal variation in the performance of deep-learning-based techniques for predicting COVID-19 health outcomes in India. Several widely used deep learning approaches, including CNN (convolutional neural network), RNN (recurrent neural network), Vanilla LSTM (long short-term memory), Bidirectional LSTM, and LSTM Autoencoder are investigated for spatio-temporal performance variation.

Chapter 8 conducts a study where ordinary least square (global) and geographically weighted regression (local) methods are employed to explore the geographical relationships between COVID-19 deaths and different driving factors. It also investigates whether geographical heterogeneity exists in the relationships. More specifically, in this chapter, the geographical pattern of COVID-19 deaths and its relationships with different potential driving factors in India are investigated and analysed.

Chapter 9 concludes the thesis work by providing summaries of the presented work, its limitations, and some future directions.

Chapter 2

Literature Review

A key component of creating a more advanced society relies on the process of sensing and monitoring diverse facets of urban life, including pollution, traffic, and health [3, 6]. More specifically, efficient acquisition of data from various urban dynamics is a critical criterion for creating societal application for better and smarter monitoring. In this context, this chapter mainly discusses the existing works associated with various urban dynamics and different sensing techniques such as traditional and participatory sensing. The various challenges and problems that are not adequately addressed in existing work for participatory sensing are also covered in this chapter. Moreover, the existing works on the predictive modelling for spatio-temporal data of urban dynamics are also presented. Specifically, Section 2.1 discusses about urban dynamics and societal applications. Section 2.2 and Section 2.3 presents traditional sensing and participatory sensing respectively. Section 2.4 discusses major issues and challenges associated with the participatory sensing. Some of the representative existing literature are presented in Table 2.1.

2.1 Urban Dynamics and Societal Applications

The rapid growth of the worldwide urban population would bring to the cities to face some increased variety of risks, concerns, and problems. For example, degrading environmental conditions of air, water and noise pollution, deteriorating transportation system, poor road condition etc. In this context, an urban sensing system seeks to address these issues by using the data generated from different city dynamics [6, 25]. Here, city/urban dynamics can be defined as those aspects of a city that directly influence citizens' quality of urban life like environment, transportation, mobility, driving and road condition, etc. The study of urban dynamics at high granularity in a large area like a city has a significant role to correctly tackle the issues related to different dynamics.

Traditionally, separate urban sensing applications are designed to monitor different urban dynamics like environment, transportation, mobility, etc. Sensing and monitoring of different aspects of a city (pollution, traffic, heath, etc.) for improving the quality of citizens' life is an essential component of building various societal application. A lot of societal applications were already designed and implemented in past and are in use today. For instance, applications like AirSense [7], HazeWatch [8], Common Sense [26], and GasMobile [27] develop light weight and low-cost devices for air quality assessment. These applications depend on

citizens' direct involvement in air quality monitoring, which necessitates the use of IoT-based modules to acquire data. Similarly, several applications are also available for health monitoring and smart home. HealthAware [9] and SPA [28] are examples of smartphone assisted systems for health care and wellbeing.

2.2 Traditional Sensing

This type of sensing makes use of conventional techniques of the human senses or standard devices to collect data from the environment. There are existing methods that use traditional sensors for sensing different dynamics of a city. Even though these traditional sensors have the advantages of high reliability and accuracy, there are several problems associated with it.

1. The traditional approaches of using static sensors for sensing urban dynamics are not only expensive but also unable to provide a scalable solution. For example, in the case of studying city pollution dynamics using the traditional methods, the cost of installing and maintaining the static air and noise pollution monitoring stations in a city becomes very high which limits the scalability. Similarly, trying to curb rash-driving or monitoring the transportation system by installation of ultrasonic and loop detectors or constant vigilance through CCTVs throughout a city is not very economical for developing countries. Also, continuous monitoring of traffic conditions, road conditions, etc. at a citywide scale is difficult because it is not financially feasible to deploy sensors for every segment of the roads.

2. Moreover, the traditional approaches to study urban dynamics work at a macro level. That's why it fails to produce high granular inferences. For example, the sparsely distributed pollution monitoring stations in a city unable to track individuals' exposure to pollution levels at high granularity.

3. Traditionally, separate urban sensing systems are designed for different city dynamics. This approach looses the benefits of having new inferences that can be drawn by aggregating information by combining several urban sensing systems.

2.3 Participatory Sensing

There is a relatively new user-centric sensing paradigm called participatory sensing that can be used to address the above-mentioned shortcomings. Here, participatory sensing is a method where a large group of individuals collectively share data and extract vital information from that data in order to manage, plot, estimate or infer various phenomenon of their common interest [4] [5]. In other words, this method of sensing allows individuals to voluntarily participate in contributing high granular data for different dynamics of a city. The number of mobile phone users was forecasted to pass the seven billion mark by 2021 [29]. Now, the rich set of smartphone equipped sensors enable the users to sense the location and other sensor data and allow them to capture images and videos for the sensing contexts. Thus, they can act as sensor nodes and collect a huge amount of location-aware data that can be

anonymized and processed making participatory sensing an extremely effective approach in the way of building a smarter and better society. Participatory sensing prioritizes the involvement of citizens for sensing and noting their daily activities, like where they live, work, eat or monitor their exposure to the ambient air or noise pollution, etc. It can vary from individual observations to the composition of data collected from several users to interpret and map different patterns across the entire city. Participatory sensing is not only highly economical but also greatly effective in spreading general awareness of the existing urban problems among the citizens and directly paves the path for a better life in a healthier society. This has led to the emergence of the idea of noise and air pollution monitoring, driving and road condition monitoring, etc.

Participatory sensing using IoT devices (e.g. smartphones) has recently been used as a sensing method to explore a city's various dynamics (e.g. pollution, human mobility, road condition, etc.) at high spatiotemporal resolution [3, 10, 15, 30]. The IoT based participatory sensing generates sensor data stream [31, 32] that could be analyzed in many applications that are related to those dynamics. There are numerous attempts to make citizen-centered improvements through the utilization of such sensing data [6]. This type of sensing generates a huge amount of data that covers a wide variety of information, including human mobility, city pollution, road conditions, etc. Many works investigate data acquired from participatory sensors such as from smartphone equipped sensors [11–13, 18, 27, 33], vehicles GPS [34], car-mounted cameras [35], on-body sensors [36], etc. For instance, the existing works [12, 33] explore different phenomena related to human mobility dynamics using citizens' location data from smartphones. In [13], the authors focus on exploring road condition dynamics using smartphone sensor data. They proposed an efficient way of estimating the transverse slope profile of a road segment using GPS, gyroscope, and accelerometer sensor data of smartphones. In [37], Cao et al. utilize high granular trajectory data (mobile cellular accessing data) to explore daily commuting patterns on a citywide scale. In [35], Tokuda et al. proposed an approach for generating a spatio-temporal pedestrian density map using street-level images of New York City. Moreover, smartphone based human mobility and activity data are also used for transit map generation [38], quantifying people's behavior in public areas [39], health-optimal routing [40], etc. Similarly, there are existing works that focus on exploring pollution dynamics of the city using smartphone based participatory sensing data [41].

It is worth noting that the above mentioned applications fall under independent urban sensing systems for different urban dynamics. However, in a vast region such as a city, the high granular study of urban dynamics plays a crucial role in accurately addressing the problems associated with various phenomena of interest. Therefore, combining data from these separate urban sensing systems is necessary to make a number of novel inferences. In this particular case, conclusions that are more accurate can also be drawn, which will benefit society greatly.

2.4 Challenges of Participatory Sensing

There are a number of open challenges and issues in IOT-based participatory sensing, which required the researchers' attention. In this section, we primarily concentrate on some significant issues associated with participatory sensing, namely *privacy preservation* [14, 42], *incentive mechanism design* [15], *data quality* [16], and *task allocation*. For instance, participatory sensing systems rely heavily on the engagement of a substantial user base to gather the necessary data. However, these systems face challenges related to user participation. On one hand, users are burdened by the expenses associated with sensing and transmitting data, while on the other hand, they are concerned about privacy issues stemming from the disclosure of sensitive information such as their GPS locations. These dual concerns, pertaining to both cost and privacy, often discourage users from actively participating in sensing tasks. Consequently, offering a compelling reward or incentive becomes crucial to motivate users and encourage their involvement in these tasks. It is worth noting that efficient incentive mechanism design is a complex issue, as it requires a trade off between a number of different goals (e.g., increased participation, effective allocation of sensing tasks, resource optimization, privacy preservation, etc.) to build a sustainable and effective participatory sensing ecosystem. Moreover, the participatory-based data collection technique depend on contributions from a diverse group of people, hence, there may be serious issues with the quality of the data [16]. The participants, for instance, may contribute incorrect and low-quality data, which needs to be handled carefully.

2.4.1 Privacy Preservation

The scientific community has spent considerable time and effort in the last few years looking into privacy and security concerns in participatory sensing-based systems [43–48]. Ding. et al. [49] presents an application for s-Health (i.e., smart health) and the associated privacy issues. The privacy-aware solution helps people who have respiratory disorders by suggesting low-pollution paths for individuals in order to alleviate their respiratory-related issues. In [50], Xie et al. proposed a system called PAMS for privacy-aware traffic monitoring. They demonstrated that the collected information can preserve location privacy of drivers while ensuring effective traffic monitoring. In [51], Zavalyshyn et al. developed a privacy-aware framework called HomePad for home environments. The framework intends to offer the users with control over how applications access and process confidential data acquired by smart devices (such as web cams), as well as to stop applications from running unless they adhere to the privacy constraints set by the users. Also, several other existing works [52–54] focus on the security aspect and privacy of IoT-based health and home environment. Some IoT-based pollution monitoring applications also exist that focus on privacy issues of the end users [55, 56].

2.4.2 Task Allocation and Incentive Mechanism Design

In a participatory sensing based system, a task provider (also known as a crowdsourcer) may have a set of sensing tasks regarding different dynamics of a city. Note that adequate users'

participation is necessary to acquire a sufficient amount of data which is a key factor for the participatory sensing based systems to provide good service quality. The task providers appoint a set of task executors (smartphone users i.e. participants of crowdsensing tasks) to execute those sensing tasks. But, existing works on sensing task allocation suffer from lack of good incentive mechanisms that are attractive for the task executors.

In theory, there are several incentive mechanisms available for participatory sensing based systems that encourage participants to perform the sensing tasks [57–61]. In [62], Chow *et al.* discuss the issue of the incentives for the exchange of sensing data. There are several literature on designing incentive mechanisms for mobile crowdsensing, that is similar to participatory sensing [63–65]. In [66], the authors propose a participant recruitment system for participatory sensing based data collection. However, the system only enables the organizers to identify appropriate participants for sensing services, it doesn't provide any incentive mechanism to motivate the users for data collection. Jin et al. [67] design a quality-aware incentive mechanism for MCS systems using reverse auction. However, they make the assumption that the total QoI (quality of information) that all winners have on a particular task is the sum of the QoIs of the winners that execute the task. In [68], Lee *et al.* introduce a reverse auction based incentive method to encourage user participation. They consider dynamic pricing, where the participants provide the sensing data to the provider at the bid values claimed by the participants. However, the introduced reverse auction based incentive model unable to hold certain desirable properties like truthfulness. Esmaeilyfard et al. [69] provide a game-theoretic incentive mechanism for a multi-publisher and multitask mobile crowdsensing system. They use Stackelberg model where the task publishers act as leader because they determine the rewards. Besides, users act as followers because they choose the level of task participation based on the publishers' rewards. However, their proposed model may not satisfy some important characteristics such as budget balance, individual rationality, etc. In [63], Yang *et al.* propose two incentive mechanisms: platform-centric model and user-centric model. The platform-centric model is based on a Stackelberg game, where the platform (the leader) enjoys more benefits than the participating users (the followers). On the other hand, the authors also provide a user-centric model called LBS that is based on an auction game where the participating users have more control over the incentive/reward determination. Another interesting work based on the Stackelberg game is presented by Li et al. [70]. For the strategic interactions in their privacy-preserving MCS scenario, they suggest a three-stage Stackelberg incentive system. They take into account privacy, fairness, and quality issues in their incentive system. However, the method does not satisfy the truthfulness property. In [71], Liu *et al.* propose an incentive mechanism called IMRAL for location aware-sensing. The incentive mechanism IMRAL is designed based on a reverse auction game. IMRAL aims to increase the participation of the users by maximizing their utility. Later in Section 5.5.3, the performance of TATA is compared with two above mentioned existing auction based incentive mechanisms IMRAL [71] and LBS [63]. In [72], along with the incentives, the authors also consider the privacy protection issue in the design of the incentive mechanism. The above mentioned incentive mechanisms are designed for the offline

scenario, although the online incentive mechanisms for participatory sensing are also available in literature [73, 74]. In case of the offline scenario, all the participating users announce their strategic types beforehand. In contrast, in case of the online scenario, the users may appear in a random manner and the decisions will be taken immediately upon arrival of the users.

The design of double auction-based incentive models for crowdsensing has gained popularity in recent years. Zhang et al. [75] provide auction-based incentive mechanisms using three auction-based models (IMC-SS, IMC-SM, and IMC-MM) and show that the proposed models satisfy various desirable properties. The IMC-SS and IMC-SM models only consider a single requester, with competition limited to providers. The IMC-MM modelm on the other hand, considers several requesters and providers, with a double auction used to engage competition among both providers and requesters. However, the work suffers from several limitations. For instance, the assumption in their problem formulation is that, each job should comprises of a number of smaller tasks, and the winner can work on a certain subgroup of these tasks. Moreover, the essential condition for ensuring truthfulness in the incentive mechanism proposed by Zhang et al. [75] is that there is no monopoly provider over the selected jobs. Also, they do not consider location awareness in their problem formulation. Another important incentive model is presented by Ni et al. [76]. Specifically, on the basis of the double auction and taking into account worker dependabilities, Ni et al. [76] suggest a differentially private mobile crowdsensing incentive model. They create the incentive system by applying the exponential process in a two-sided auction to pick the market price. They employ the widely-used uniform price system, in which all successful workers are paid the same clearing paying price and all successful requesters are charged the same clearing charging price. For double-sided heterogeneous MCS marketplaces, Chen et al. [77] present a truthful double auction. However, they make the assumption that there is no user collusion during the auction and that each user's stated expenses are unique. Chen et al. [78] develop a two-sided auction called SPARC that is strategy-proof for MCS. However, their problem formulation assume that the demand for sensing time cannot be manipulated by the sensing task. Moreover, the presented model does not hold some important characteristics, such as budget balance and computational efficiency. Deng et al. [79] investigate a double auction market design in which the auctioneer seeks to maximise its overall revenue. They make use of discriminative pricing models. Jin et al. [80] introduce a framework for crowdsensing that comes up with an incentive model. They presume that the platform already understands the reliability level matrix and has a history of it. However, the incentive mechanism is mainly designed in a manner that will enable the most user engagement, and every winning worker requester receives the supremum and infimum of her bid from the pricing process. Also, budget balance may not be ensured in their work.

2.4.3 Data Quality

The raw data in a participatory sensing-based approach is typically collected using low-cost sensors of IOT devices and smartphones [81]. But, the use of low-cost IOT devices and smartphones for sensing urban dynamics (e.g., ambient pollution level, road condition, etc) do not

provide accurate high quality measurements. For instance, Maisonneuve et. al. [82] shows that sensor devices that are not properly calibrated might provide data that is potentially inaccurate or even deceptive. Specifically, the authors conduct an experiment for an ambient noise pollution monitoring application to assess the accuracy of the loudness calculated with a low-cost smartphone in comparison to a sound level meter. Zhu et. al. [83] also emphasise the significant measurement inaccuracies that result from the direct use of inexpensive equipment, such as smartphones, for participatory-based sound level sensing. Moreover, the quality of participatory sensing data also depends on the sensing context [82]. For instance, the accuracy of the ambient noise data that are collected using the smartphones also depends on the sensing contexts, like whether the smartphone is carried in hand or pocket. Existing research also emphasise other dynamics, such as air pollution, where the sensing context (landuse, indoor, outdoor, traffic situation, etc.) will be useful in excluding the data samples originating from improper sensing contexts [41].

2.5 Predictive Techniques for Spatio-temporal Analysis

In the previous works, several predictive methods are employed to perform spatio-temporal analysis on the participatory sensing data. Specifically, various state-of-the-arts statistical **[ref]**, machine learning**[ref]** and deep learning**[ref]** techniques exists that could be used as the predictive methods. These methods are used to infer knowledge, spot patterns, and discover trends in urban dynamics. For instance, environmental pollution monitoring, traffic and road condition monitoring, smart health, smart home, etc. are some of the city dynamics where the participatory sensing based applications use these techniques. Capponi et al. [3] perform a detailed survey on participatory sensing/crowd sensing sensing systems that highlights the use of statistical, machine learning and deep learning-based modelling for managing large and heterogeneous sensing data.

Buelvas et al. [84] present the role of spatial method like interpolation to enhance the spatial or temporal resolution of IoT-based air quality data. In another study, Nunez et al. [85] employ spatial interpolation techniques to determine the geographic distribution of the air pollutants. In literature, there are several methods available for spatial interpolation [86]. Among them, the two popular methods are Inverse Distance Weighting (IDW) [87] and Kriging [88], which are frequently used in the field of environmental science. Data from various urban dynamics such as ambient air and noise pollution that are collected using IoT-based approach can be used by spatial interpolation methods to generate spatially continuous pollution maps. Note that, spatial interpolation is the process of using the values at observed locations to predict the values at other unknown locations in geographic space. The rationale behind spatial interpolation is the assumption that everything is related to everything else, but near things are more related than distant things in space (Tobler's Law of Geography [89]).

Apart from the predictive techniques such as spatial interpolation, various other statistical techniques are widely used for spatio-temporal analysis of various urban dynamics. For instance, Geographically Weighted Regression (GWR) [90,91] is a local statistical modelling

technique that examines the location-specific relationships between a set of independent variables and a dependent variable. It is a extension on the conventional Ordinary Least Squares (OLS) [92] regression that takes geographical heterogeneity into consideration and makes it possible to look at how the relationship between variables varies across different locations. Many variants of GWR such as semi-parametric geographically weighted regression (s-GWR) [93] and multi-scale geographically weighted regression (MGWR) [94] exist in the literature. Note that GWR and its variants can be used to explore the geographical relationships of urban dynamics with various driving factors including socioeconomic, demographic, or environment related factors.

A wide range of applications of machine learning and deep learning techniques are also found in spatio-temporal analysis of urban dynamics. For example, CNN (convolutional neural network), RNN (recurrent neural network) and various LSTM (long shortterm memory) variants such as Bi-LSTM, Stacked LSTM, etc are popularly used in spatiotemporal forecasting of different urban dynamics [95–98]. In [96], Ong et al. presented a RNN based approach that makes use of an auto-encoder for $PM_{2.5}$ time series forecasting. Mohimont et al. [98] employed temporal CNN to forecast various health outcomes, e.g., number of confirmed cases, deaths, etc. Moreover, various learning techniques are also frequently used in other spatio-temporal analysis of urban dynamics such as optimal site selection, missing value imputation, spatial and temporal mapping of any phenomena of interest, etc [99–101].

2.6 Summary

This chapter provides a comprehensive overview of the present state of research in urban sensing, emphasising the strengths, limitations, and challenges of traditional and participatory sensing methodologies. Furthermore, it highlights the significance of predictive modelling tools for analysing spatio-temporal data in the context of diverse urban dynamics. Traditional sensing methods are discussed, highlighting their reliability and accuracy advantages. However, challenges such as high cost, limited scalability, and lack of granularity are found, especially in applications like pollution and health dynamics monitoring. Participatory sensing, a user-centric paradigm leveraging smartphones as IoT devices, is discussed as a cost-effective and citizen-engaging alternative. Applications such as pollution monitoring and health awareness are cited, emphasizing the economic benefits and effectiveness of involving citizens in data collection. The chapter discusses issues and challenges in participatory sensing, including privacy preservation, incentive mechanism design, and data quality. Privacy concerns related to disclosure of personal information (e.g., location data) and the need for compelling incentives to motivate user participation are highlighted. The impact of data quality on the reliability of participatory sensing is also acknowledged. The chapter also discusses the use of predictive techniques for spatio-temporal analysis in the context of urban dynamics. Spatial interpolation, statistical methods, and AI techniques, including CNNs, RNNs, and LSTMs, are explored for forecasting and analyzing data from participatory sensing, enhancing our understanding of urban phenomena at high resolution. The literature review provides a comprehensive overview of the current state of research in urban sensing,

TABLE 2.1: Existing Literature

Research area	Year	Author	Application	Method	Description
Urban dynamics	2010	Rana et al. [102]	Ear-Phone	-	Urban sensing approach to monitor roadside ambient noise
	2008	Mohan et al. [103]	Neri cell	-	Monitor traffic and road condition of a city using the in-build sensors of smartphone.
	2017	Dutta et al. [7]	AirSense	-	Low-power, low-cost, light weight IoT device is developed to monitor air pollution.
	2017	Dutta et al. [104]	NoiseSense	-	Noise pollution monitoring system, based on context aware smartphone sensing.
	2013	Sivaraman et al. [8]	HazeWatch	-	Low-cost participatory sensing framework for monitoring urban air quality levels.
Privacy preservation	2020	Benegui et al. [105]	-	CNN	User Identification based on Motion Sensor data of Tap on screen motion represented
	2020	Angrisano et al. [106]	-	RF	Walker identification using ensemble learning
	2016	Neverova et al. [107]	-	CNN	Walker identification by dense convolutional Clockwork RNNs
	2016	Buriro et al. [108]	-	NN	User identification based on sensory data of finger movement on smartphone screen
	2011	Shi et al. [109]	-	NB	Senguard: Passive user identification on smartphones using multiple sensors
Incentive and task allocation	2015	Zhang et al. [75]	-	Double auction	Provide auction-based incentives mechanisms using the three auction-based models
	2015	Jin et al. [67]	-	Reverse auction	Proposed a quality-aware incentive mechanism for MCS systems
	2022	Zhang et al. [110]	-	Reverse auction	Incentive mechanism based on non-deterministic mobility forcrowdsensing data collection.
	2022	Esmaeily et al. [69]	-	Stackelberg game	Incentive mechanism for a multi-publisher and multitask mobile crowdsensing
	2022	Chen et al. [111]	-	Reverse auction	Incentive scheme for vehicular crowdsourcing
Data quality	2016	Guillaume et al. [81]	-	-	Noise mapping depending on participatory measurements
	2015	Zhu et al. [83]	-	-	Explore inaccuracies that result from the use of inexpensive equipment, like smartphones, for participatory-based sound level sensing
	2009	Maisonneuve et al. [82]	Noise Tube	-	Shows calibration for smartphones-based ambient noise data to determine noise exposure of individuals.
Predictive modelling for spatio-temporal data	2016	Yabe et. al. [33]	-	-	Predicts irregular movement of humans after the mid-level disasters.
	2017	Gu et. al. [12]	-	-	Predict venue(s) visited by citizens from the GPS trajectory.
	2021	Mohimont et al. [98]	-	CNN	Forecasting methods for COVID-19 health outcome
	2020	Yang et al. [112]	-	GWR	Explores spatially varying relationships of PM2.5 with meteorology
	2018	Alimissis et al. [113]	-	ANN	Spatial estimation of urban air pollution

emphasizing the strengths and limitations of traditional and participatory approaches and the importance of predictive modeling techniques.

Based upon earlier discussions (refer to Section 2.3 and Section 1.2), the initial observation is the necessity for a unified framework in participatory sensing. Consequently, in the upcoming chapter (Chapter 3), we develop and implement a unified framework designed to integrate various applications for participatory sensing, leveraging IoT-cloud technologies.

Publication(s) related to this Chapter:

- *Middya, A.I., Dey, P. and Roy, S., 2022. IoT-based crowdsensing for smart environments. In Internet of Things for Smart Environments (pp. 33-58). Cham: Springer International Publishing. (DOI: https://doi.org/10.1007/978-3-031-09729-4_3)*

Chapter 3

Unified Framework Design for Participatory Sensing Applications

3.1 Introduction

The focus of this chapter is to design and implement a unified framework for integrating various participatory-based urban sensing applications that are used to improve citizens' QoL (Quality of Life) by collaborating data from the applications. A participatory-based urban sensing system can be considered to be a system that utilizes information and communication technologies (ICT) like sensors and actuators, wireless sensor networks (WSN), Internet of Things (IoT), etc. to enhance the quality and performance of urban services and utilities. The rapid growth of the worldwide urban population would bring to the cities to face some increased variety of risks, concerns, and problems. For example, degrading environmental conditions of air, water and noise pollution, deteriorating transportation system, poor road condition etc. In this context, a participatory-based urban sensing system seeks to address these issues by using the data generated from different city dynamics. Here, city dynamics can be defined as those aspects of a city that directly influence citizens' quality of urban life like environment, transportation, mobility, driving and road condition, etc. Traditionally, it is common to have independent urban sensing systems for different city dynamics. However, combining these independent urban sensing systems to aggregate information can help us to draw various new inferences. In this case, inferences can also be drawn with a higher accuracy that will give numerous benefits to society. Thus, an urban sensing system needs to be visualized as a collection of applications related to different city dynamics. For example, data from the environment through applications like air pollution monitoring and noise pollution monitoring can impact the transportation or mobility of people. More specifically, it integrates urban sensing applications, data storage, data analysis, and service offering into a continuous process to improve the lives of people. Therefore, the future of urban sensing systems will revolve around the ability to combine data from different dynamics of a city to make better decisions in our day-to-day life.

3.1.1 Objective

There are existing methods that use traditional sensors for sensing different dynamics of a city. Even though these traditional sensors have the advantages of high reliability and accuracy, there are several problems associated with it (see Chapter 2.2 for more details). In this context, participatory sensing is a new user-centric sensing paradigm that can be used to address the above-mentioned shortcomings (see Chapter 2.3 for more details). Now, in the previous works, some participatory sensing applications (namely AirSense, NoiseSense, SenseDCity, and D&RSense) were introduced for building a smarter and better society [114–118]. These applications can directly address specific urban problems such as air pollution (AirSense [114]), noise pollution (NoiseSense [115]), human mobility (SenseDCity [116]), aggressive driving and poor road conditions (D&RSense [117]) using IoT devices and smartphones. In this context, the objective of the chapter is to come up with a unified framework called JUSense (Judicious Urban Sensing) which is used to derive benefits from these applications by combining their functionalities. Here, the primary motivation of designing JUSense comes from the lack of existence of such a unified framework. Moreover, in order to enhance the accuracy and reliability of the participatory sensing data, it is essential for such participatory sensing based unified framework to address some critical pre-processing steps namely context-aware data cleaning, calibration, etc. The context-aware data cleaning can be used to remove erroneous sensor values. Specifically, it ensures that erroneous sensor values, not recorded in appropriate sensing contexts, are removed, thereby enhancing the reliability of the data. On the other hand, since low-cost IoT devices and smartphones may not provide high-quality measurements, calibration is required which helps predict actual measurements from sensor responses. Specifically, sensor calibration can improve data accuracy before analysis. Hence, the additional objectives are to explore context aware data cleaning technique and calibration mechanisms for IoT-based participatory sensing data collection. These ensure that the data used for decision-making is reliable and of high quality.

3.1.2 Contribution

The major contributions of this chapter can be summarized as follows:

- A unified framework called JUSense (Judicious Urban Sensing) is proposed to integrate IoT-cloud based various applications for urban sensing. Four urban problems are considered and their corresponding subsystems are integrated with the current implementation of JUSense. These subsystems can accumulate information independently but at the same time also improve their functionalities and results, and validate their conclusions through mutual interaction. JUSense can also act as a recommendation system. For example, depending on the users' health profile, a suitable route (less air/noise polluted) which is good for his/her health is recommended.

- A context-aware data cleaning algorithm is explored that distinguishes between "in-pocket" and "in-hand" phone positions during participatory based data collection. Moreover, IoT-Device specific calibration mechanism is developed for noise pollution dynamics in smartphone-based crowd / participatory sensing.

- A new multi-view data fusion technique is proposed for spatiotemporal urban air and noise pollution map generation. The inference of air/noise pollution level of a POI is done through its spatial view as well as temporal view. A spatial interpolation technique is employed to create urban air and noise pollution maps.

- The proposed system JUSense is evaluated using large-scale experiments on real world participatory sensing based dataset collected by participants over a period of 1 year (June, 2018-June, 2019).

3.2 Existing Works

In this section, first we present a few existing systems/ platforms that can support multiple urban sensing applications. Next, we provide different participatory sensing based methods that are used in existing literature for exploring different dynamics of a city.

There are lack of existing systems/platforms that can works on a diverse set of applications. Although, some loosely related platforms or projects are available in the existing literature. For example, Zhao et. al [119] proposed an integrated processing platform called DeCloud4SD for traffic sensor data. It is designed to provide services for retrieval, storage, acquisition, and computation of traffic sensor data in a scalable framework. Li et. al [120] proposed a cloud-based holistic data management system for both real-time and historical trajectory data. This system accommodates various trajectory queries that can be utilized by various urban sensing applications so that they can be realized in a much simpler way. Hochreiner et. al [121] proposed an approach called VISP for elastic data stream processing in IoT. VISP helps the user through the reuse of existing components for case-specific topologies and a GUI to create a new topology. Also, there are other existing literature [122] [123] that can work on various application based services in IoT environment.

A considerable amount of research has been performed on monitoring urban noise pollution dynamics. Developing countries lag in noise assessment and noise control due to financial constraints faced in deploying expensive noise monitoring stations. In this context, noise monitoring through participatory sensing can have a remarkable impact. Rana et al. [102] have presented an end-to-end noise pollution mapping system named Earphone. It utilizes the microphone along with the GPS and system clock available in smartphones to show differences in noise levels when the phone was carried in hand, bag, pocket, and waist. Can et al. have proposed a cross-calibration method that depends upon outlier detection, sensor-based correction and L_{den} (Day-Evening-Night Sound Level) calculation. It is used for managing and refining the raw data gathered through low-cost sensor networks using a participatory sensing based approach [81]. Also, there are approaches where citizens themselves act as sensors to sense the ambient noise. They voluntarily sense the environment and provide the sensed facts for processing. For instance, New York City (NYC) had opened a complaint line named 311 where the citizens could lodge complaints about different problems across the city. In this context, Zheng et.al [124] focused on noise pollution since noise was the third greatest class of complaints. The authors have modeled the noise condition of NYC

by aggregating the data collected from citizens across the city along with the road networks data, social media data, and Points of Interests (PoI).

Now, in the case of urban air quality monitoring, the concept of citizens as sensors along with smartphone or IoT as sensors can also be applied. U-Air is an approach where real-time information about air quality has been informed across a city [125]. It is based on both real-time and historical air quality data that was reported by established air quality monitoring stations in the city. HazeWatch project [8] uses lightweight air quality monitoring devices. These devices are mounted on private vehicles and they interface with smartphones to send accumulated data to a server. The collected data can be aggregated with data collected by other users so that one can have a city wide pollution map. In [126], the authors have proved the effectiveness of social media data for monitoring air quality trends and associated human perception and response. They mine a total of 93 million messages from Sina Weibo (it is China's largest microblogging website) and perform a qualitative study of the content of air pollution related social media messages. The authors come up with the fact that the volume of air pollution related social media messages are highly correlated with a particular pollution level.

Some important research works are also available on monitoring other urban dynamics like road condition, driving, human mobility, etc. Eriksson et al. have proposed an approach called Pothole Patrol (P2) to gather data from vibration and GPS sensors of smartphones, and processes the data to assess road surface conditions [127]. Nericell [103] is an approach that monitors road and traffic conditions using the data gathered from smartphone sensors like accelerometer, microphone, GPS, etc. Ly et al. [128] have put forward research on driving style recognition and classification. They use the gyroscope and accelerometer sensors found on smartphones, and they analyze data using machine learning algorithms such as SVM and k-means clustering. Kuang et. al [129] propose a deep-learning based model to accurately predict taxi demand using historical taxi trips records. In [130], the authors propose an approach to provide next POI recommendation based on users' check-in data.

Each of the above mentioned works is crucial and independent research work mainly based on IoT and participatory sensing and can inspire various research works in the field of urban sensing. All these works have been carried out independently. However, the benefits arising from combining these kinds of research works are not explored yet. So, in this chapter, we have developed a participatory sensing based system named JUSense that essentially looks into the collective advantages that can arise by combining results of some individual systems.

3.3 Problem Statement

While existing urban sensing systems utilize sensors and networks to gather data for improving city services and citizen quality of life (QoL), they are often isolated and address specific dynamics like air pollution, noise pollution or traffic independently. This limits the ability to draw broader insights and make better decisions by combining data from different aspects of the city. Additionally, traditional sensors can be expensive and have limitations in terms of

coverage and deployment. This research addresses the need for a unified framework that integrates various participatory sensing applications focused on different urban dynamics. By leveraging user-generated data from smartphones and low-cost IoT devices, the framework aims to:

1. *Combine data from diverse applications*: Overcome the limitations of isolated systems by integrating data from applications addressing air pollution, noise pollution, traffic, and road conditions.

2. *Enhance data quality and reliability*: Develop context-aware data cleaning techniques and calibration mechanisms to ensure the accuracy and reliability of participatory sensing data before analysis and decision-making.

3. *Derive broader insights and improve QoL*: Enable the generation of comprehensive spatiotemporal maps of air and noise pollution, and leverage the combined data for personalized recommendations (e.g., suggesting healthier routes based on user health profiles).

3.4 JUSense: Urban Sensing Applications

JUSense is a participatory sensing based system that involves four major applications (namely AirSense, NoiseSense, SenseDCity and D&RSense) to address the primary issues like air and noise pollution, citizens' mobility pattern, road conditions, etc. JUSense has been developed keeping in mind the additional and enhanced services that could be provided by an urban sensing system when the subsystems would interact with each other. With this motive, the framework of JUSense has been proposed that can also be used by other societal applications. In this chapter, a unified framework called JUSense (Judicious Urban Sensing) is proposed to integrate IoT-cloud based various applications for urban sensing. The unified framework JUSense not only integrate IoT-cloud based above-mentioned urban sensing applications but also enhance the functionality of the applications and report on improved personalized recommendations through mutual interactions.

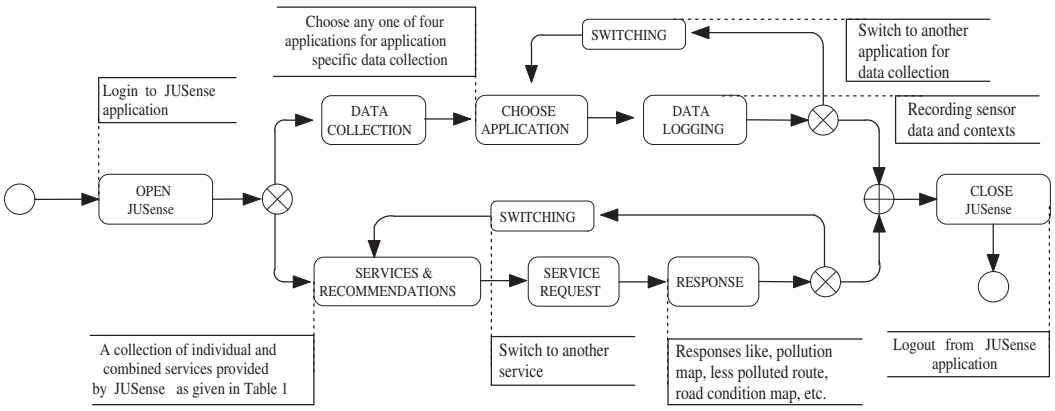


FIGURE 3.1: Workflow of JUSense from the end-users' viewpoint.

3.4.1 Subsystems Overview

In this section, from the end-users' viewpoint, we first demonstrate an abstract workflow of the application JUSense. The workflow is presented in Fig. 3.1. As shown by this figure, a particular user can login to JUSense using his/her login details. The system JUSense allows the user to share various data (smartphone sensor data, ambient air and noise levels, feedback, etc.). Here, a user can choose any application (e.g. NoiseSense, SenseDCity, etc.) to share that application specific data. During data collection, the user can logout from JUSense at any time or can switch to other applications. The user interfaces (UI) of JUSense for the data logging activities of different applications are shown from Fig. 3.2(a) to Fig. 3.2(d). Fig. 3.1 also shows that JUSense allows the user to get various services. If the user shares his/her data with JUSense, it not only provides services like air pollution map, noise pollution map, rash driving prone zones, road conditions, popular locations but also provides him/her with some personalized recommendations. The user interface of the services like air pollution map and noise pollution map that are provided to the users are shown in Fig. 3.2(e) and Fig. 3.2(f) respectively.

In this chapter, we have concentrated on the symbiotic advantages that can arise from the information exchange of the following applications.

AirSense

AirSense is a low-cost IoT based application that along with the help of a smartphone can capture an individual's air pollution exposure. It collects ambient air pollution data using an Arduino platform based air quality monitoring device called AQMD (Air Quality Monitoring Device) with MQ series gas sensors and transfers the sensed data to the smartphone via bluetooth connectivity [114] [7]. Additionally, meteorological data (e.g. temperature, humidity, pressure, wind speed, etc.) and landuse information (e.g.residential, retail, street, etc.) are also collected by making calls to OpenWeathertMap API [131] and Overpass API [132] respectively. It gives more fine-grained information of the contexts that help us to extract more valuable statistics. Collected data are aggregated in the cloud for analysis purpose. The users receive real-time notification of the surroundings' air quality as well as air pollution map via his/her mobile application. In previous works [114] [7], a significant limitation was that the collected AQI data are not sufficient to generate a high resolution air pollution map due to the problem of data sparsity. To overcome this problem, in this work a spatial interpolation method called Inverse Distance Weighting (IDW) [133] is used to show overall spatial variability of air pollution. Here, IDW is used to predict air pollution levels at unknown locations using the pollution levels at known locations. The general form to calculate the predicted value $\hat{Z}(P_j)$ at an unknown point P_j based on samples $V(P_i)$ for $i = 1,2,\ldots,N$ using IDW is

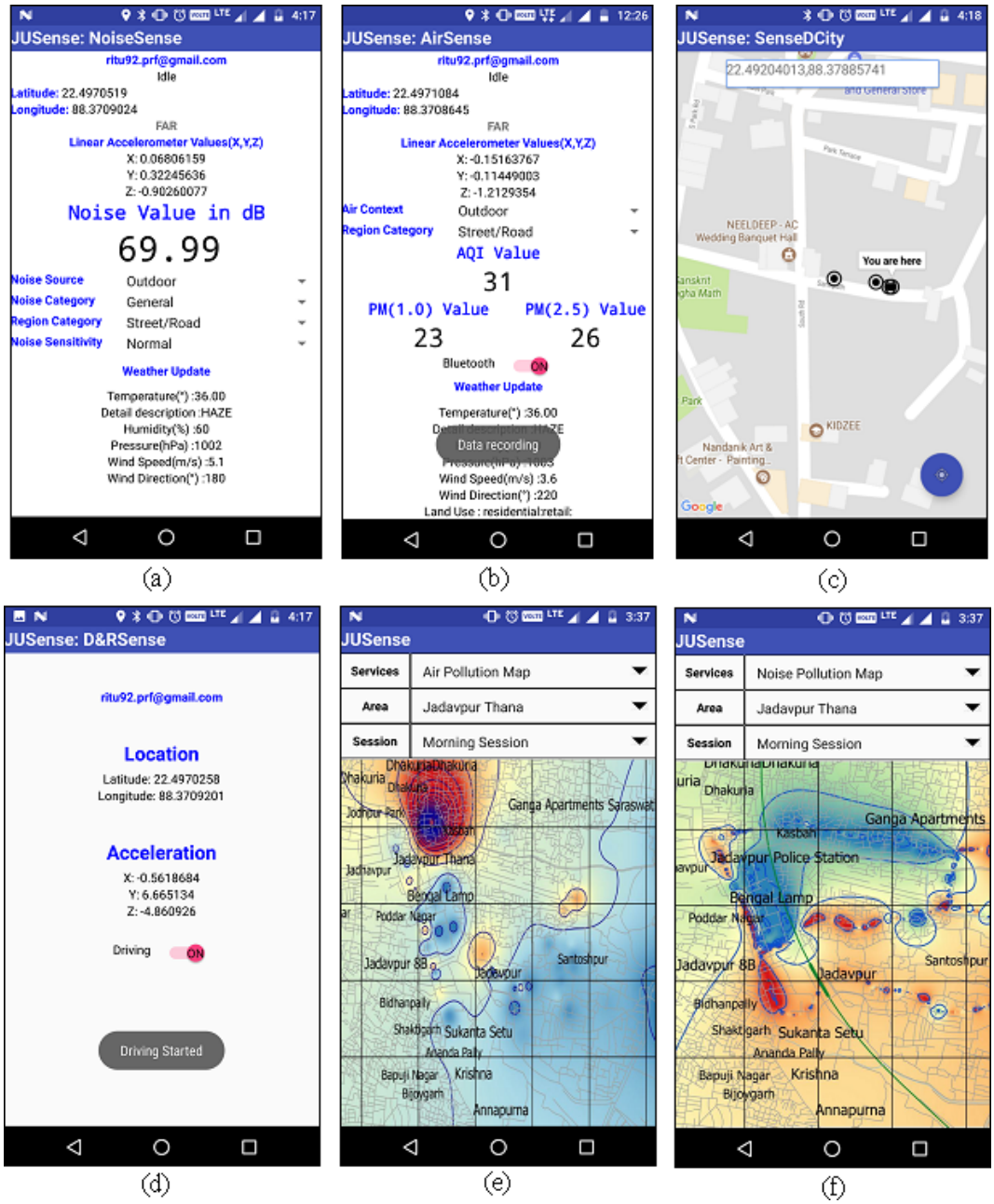


FIGURE 3.2: User interface of data logging activities for four applications (a) NoiseSense (b) AirSense (c) SenseDCity and (d) D&RSense. User interface of (e) air pollution map service and (f) noise pollution map service.

given as follows [133]:

$$\hat{Z}(P_j) = \begin{cases} \dfrac{\sum_{i=1}^{N} W_{ji} V(P_i)}{\sum_{i=1}^{N} W_{ji}}, & \text{if } dist(P_j, P_i) \neq 0 \text{ for all } i \\ V(P_i), & \text{otherwise} \end{cases} \tag{3.1}$$

Where,

$$W_{ji} = \frac{1}{dist(P_j, P_i)^{\beta}} \tag{3.2}$$

Here, P_j denotes prediction point (interpolated point), P_i is a known point (interpolating point), $dist(P_j, P_i)$ is the 2-D spatial distance between P_i and P_j, N represents the total number of known points used in interpolation and β is the power parameter. Along with the micro and macro level of air quality monitoring, an individual's pollution footprint is also monitored in this application.

NoiseSense

NoiseSense is a context aware participatory sensing and smartphone based application which gives its users real-time updates regarding individual's noise pollution exposure as well as city's dynamic noise map [115]. It uses inbuilt sensors of the smartphones to get ambient noise data from the devices using participatory sensing. For collecting the noise data, we utilize the microphone of the smartphone. It is a context aware solution that utilizes the sensing context of the user for removing erroneous noise values collected through the smartphones. For example, when a user is talking over the phone or the phone is in the bag or in the pocket then the collected noise values do not represent the true ambient noise pollution levels. Eventually, it is required to eliminate erroneous noise samples that are collected from inappropriate sensing contexts. The previous work [115] focused on POI based noise map generation. But this current work produces dynamic noise map to show overall spatial variability of noise pollution using spatial interpolation techniques. Here, Modified Shepard's Method (MSM) [134] is used as an interpolation technique which is a variant of the above IDW method (i.e. Eq.3.1 and 3.2). MSM calculates the expected value at an unknown point (i.e. location) using only the nearest sample points within a search radius R rather than all the sample points. Now, the predicted noise value $\hat{z}(p_0)$ at an unknown location p_0 based on the observed noise values $z(p_i)$ at the sampling locations p_i is given as follows [134]:

$$\hat{z}(p_0) = \begin{cases} \dfrac{\sum_{i=1}^{N} w_i(p_0) z(p_i)}{\sum_{i=1}^{N} w_i(p_0)}, & \text{if } d(p_0, p_i) \neq 0 \text{ for all } i \\ z(p_i), & \text{if } d(p_0, p_i) = 0 \text{ for some } i \end{cases} \tag{3.3}$$

where,

$$w_i(p_0) = \left(\frac{max(0, R - d(p_0, p_i))}{R.d(p_0, p_i)} \right)^2 \tag{3.4}$$

Here, $d(p_0, p_i)$ is the distance between p_i and p_0, N is the total number of known points used in interpolation.

SenseDCity

SenseDCity is a participatory sensing and smartphone based application that logs location information (say, l_i) from an individual's smartphone and keeps track of his or her daily spatio-temporal GPS trajectory [116]. Here, l_i is a triplet $l_i = (l_i.lat, l_i.lngt, l_i.t)$, where $l_i.lat$ and $l_i.lngt$ are latitude and longitude at time $l_i.t$. This application focuses to study individuals' mobility to explore travel pattern, popular locations, etc. Using 'citizen as a sensor' it also aims to identify road conditions, health issues among the citizens, etc. The system parses the GPS logs to get trajectory. Here, a trajectory is nothing but a sequence of locations $T = l_1 \rightarrow l_2 \rightarrow l_3 \ldots \rightarrow l_n$, where, $l_i.t < l_{i+1}.t$. Form the huge GPS trajectory data, stay points (e.g. bus stop, park, university, etc.) are identified that show the locations where a user stayed for a while [116]. The stay points are detected based on two parameters, a time threshold λ_t and a distance threshold λ_d. A stay point s can be considered as a virtual location that represents a group of consecutive GPS locations $L = \{l_k, l_{k+1}, \ldots, l_m\}$, where, for all i, $distance(l_k, l_i) \leq \lambda_d$ and $|l_m.t - l_k.t| \geq \lambda_t$. Now, the stay point s can be represented by $s = (s.lat, s.lngt, s.t_{cin}, s.t_{cout})$, where, $s.lat$ and $s.lngt$ denotes the latitude and longitude of the stay point respectively. The terms $s.t_{cin}$ and $s.t_{cout}$ represent the check-in time and check-out time for the stay point s. Here, $s.lat, s.lngt, s.t_{cin}$ and $s.t_{cout}$ are computed as follows [135]:

$$s.lat = \frac{\sum_{i=k}^{m} l_i.lat}{|L|} \tag{3.5}$$

$$s.lngt = \frac{\sum_{i=k}^{m} l_i.lngt}{|L|} \tag{3.6}$$

$$s.t_{cin} = l_k.t \tag{3.7}$$

$$s.t_{cout} = l_m.t \tag{3.8}$$

Hence, we find a travel sequence of stay points $S = s_1 \rightarrow s_2 \rightarrow s_3 \ldots \rightarrow s_q$ for a user's trajectory $T = l_1 \rightarrow l_2 \rightarrow l_3 \ldots \rightarrow l_n$, where $q \leq n$. Now, SenseDCity employ *frequent sequence mining* to identify frequent travel path from travel sequences. Users' stay point information also helps us to discover popular locations among the city. Additionally, SenseDCity analyses the users' feedback to provide road condition map of the city and using frequent itemset mining it identifies common health issues among the citizens.

D&RSense

A safe transportation system is one of the most essential requirements for urban sensing. D&RSense is also a participatory sensing based system which essentially utilizes the accelerometer and GPS sensors of our smartphones to report transportation anomalies [117]. It detects aggressive driving events (e.g. sudden acceleration, braking, etc.) and road anomalies (e.g. bumps, potholes, etc.). The driving events are classified by an ensemble learning

method, Random Forest (RF). The raw sensor data like latitude, longitude, 3-axis accelerometer values (acc_x, acc_y and acc_z) are used as features. The other derived features that are found to be significant are as follows: variance of acc_x, acc_y and acc_z, skewness of acc_x and acc_z, kurtosis of acc_x and acc_y, median of acc_y and acc_z, mean of acc_x and acc_y. The random forest classifier uses gini impurity [136] as a function to measure the quality of a split as follows.

$$I_g = \sum_{j=1}^{N} p(j) \times (1 - p(j)) \tag{3.9}$$

where, N is the total number of classes and $p(j)$ is the probability that a tuple in the dataset belongs to class $C_j, 1 \leq j \leq N$. By studying driving pattern of drivers we can get valuable insights into their braking style and accelerating patterns which can help categorize their driving nature while detection of bumpers and potholes can help recommend which areas of the city need maintenance. Whenever the car is driven aggressively, the phone generates an alert warning the user to check his/her aggressive driving style. Now, in order to indicate the rash driving prone areas of the city, D&RSense utilizes density-based spatial clustering of applications with noise (DBSCAN) [137] clustering to cluster several locations corresponding to those GPS points where aggressive driving incidents have occurred. Suppose, $L^{(a)} = \{l_0, l_1, \ldots, l_n\}$ is a set of such locations where aggressive driving incidents have occurred. Now, DBSCAN is based on two parameters ε and *minPts*. Here, $\varepsilon > 0$ is the search radius of a neighborhood considered for every location $l_i \in L^{(a)}$. The $\varepsilon - neighborhood$ of a location $l_i \in L^{(a)}$ is defined as a set of locations $N_\varepsilon(l_i) = \{l_j \in L^{(a)} | \; dist(l_i, l_j) \leq \varepsilon\}$, where, $dist(l_i, l_j)$ indicates the distance between l_i and l_j. The other parameter *minPts* specifies how dense the locations where aggressive driving incidents have occurred. Next, some core locations $l_i^{(c)} \in L^{(a)}$ are found that satisfy the following condition:

$$|N_\varepsilon(l_i^{(c)})| \leq minPts \tag{3.10}$$

Now, all the overlapping $\varepsilon - neighborhood$ i.e. all overlapping $N_\varepsilon(l_i^{(c)})$ sets are group together in a cluster to form a rash driving prone area/zone.

3.4.2 Design and Architecture

Inspired by the utility of the applications mentioned in the previous section, an urban sensing system called JUSense has been designed. It meets the requirements of *mutual interactions* and *exchange of information* between the individual applications for more functionalities and better performance. The symbiosis between the applications should be managed to maximize the benefit to the society. In JUSense, the selected context-aware participatory sensing based applications (namely AirSense, NoiseSense, SenseDCity, and D&RSense) can function on their own way to address specific issues of a city. In this context, a layered architecture is specified based on the required set of capabilities at each layer. The architecture is broadly divided into two functional layers: the *participatory sensing layer* and the *computation layer*.

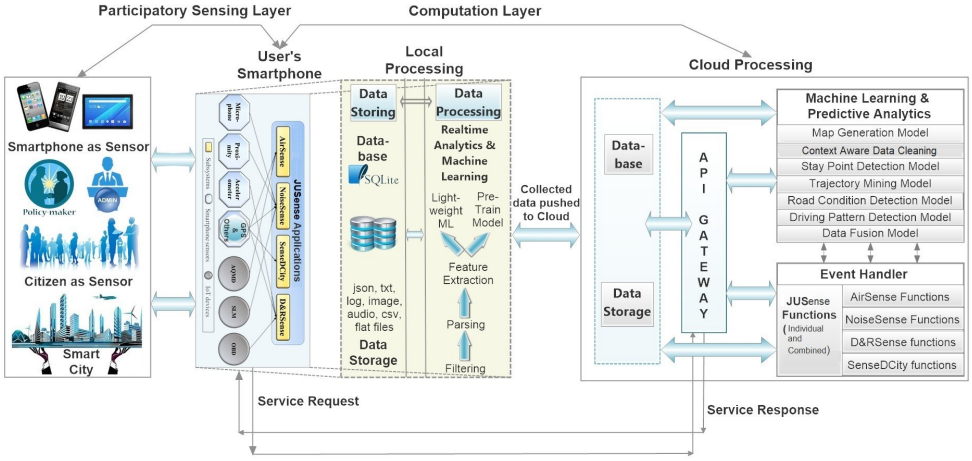


FIGURE 3.3: Architecture of the JUSense system

The proposed architecture is depicted in Fig. 3.3 that involves computation in the smartphones and in cloud. In participatory sensing layer, application specific smartphone sensor data (e.g. GPS, accelerometer, microphone, etc.) and IoT device data (e.g. AQMD for air pollution data, SLM for sound level data, etc.) are collected. Here, mobile applications (namely AirSense, NoiseSense, SenseDCity, and D&RSense) are responsible for logging the data in users' smartphones, performing basic preprocessing of the data and pushing the preprocessed data to the cloud. Some basic processing e.g. filtering, parsing, feature extraction along with lightweight machine learning based prediction is performed in the smartphone to provide individual user specific services (e.g. individual user's exposure to ambient pollution level). Now, for better accuracy and to get the aggregated view of the entire city, all the data from individual subsystems were ingested into the cloud. Now, several machine learning and predictive models (e.g. Road condition detection model, stay point detection model, data fusion model, etc.) are deployed in cloud that can make inferences based on the aggregated data. These inferences are served as a *service response* to the users based on their application specific *service request*. In the current version of JUSense, it incorporates the four specified subsystems and also handles the interactions between them.

Participatory Sensing Layer

The participatory sensing layer includes all the citizens of the city those who participate in sensing, the city administrators and the sensing medium i.e. smartphones, sensors, IoT devices etc. The participatory sensing layer essentially uses citizen as sensor i.e. citizens participate in sensing their surroundings and share the information along with their opinions and feedback via their smartphones. Here, the participatory sensing data may be application specific *sensor data* or *contextual data*. The sensor data comprises air pollution data (e.g. AQI_value, $PM1.0$, $PM2.5$, etc.), noise pollution data (e.g. ambient noise levels in dB), driving related data (e.g. accelerometer readings) and mobility information (e.g. location information). Here, Air Quality Monitoring Device (AQMD) is used to collect ambient air pollution data. The AQMD uses MQ-135 as an ambient air quality sensor. Whereas, DSM501A is used to sense particle density in ambient air. In AQMD, Bluetooth capability

is provided by HC-05 that enables citizens to transfer sensor data from AQMD to the smart handheld devices. On the other hand, the NoiseSense application measures ambient noise pollution levels within a time interval as an equivalent continuous sound level (L_{eq}). The (L_{eq}) in decibel (dB) is measured with the help of following equation [138]:

$$L_{eq} = 10\log_{10}\left[\frac{1}{t_2 - t_1}\int_{t_1}^{t_2}\left(\frac{P_t}{P_o}\right)^2 dt\right] \tag{3.11}$$

Here, in Eq. 3.11, t_1 and t_2 denote the start and end of the time interval respectively, P_t represents the sound pressure of the sound signal, whereas P_o represents the reference sound pressure level (typically 20 micropascals). In the case of D&RSense application, 3-axis accelerometer values (Acc_x, Acc_y, Acc_z) data are mainly recorded using smartphone equipped accelerometer sensor. Finally, for SenseDCity applications, users' mobility data in the form of GPS locations (latitudes and longitudes) are sensed using GPS sensor of smartphones. However, location information is useful for all the applications. The system also provides services and notifications to the users and that is why the participatory sensing layer resides in both ends of the system.

Computation Layer

The computation layer comprises of computation in the smartphone and cloud end. Uploading all the IoT and smartphone generated data into the cloud is not recommended since the entire dataset is not always useful. Moreover, it is quite challenging to handle the data on this scale. Due to this, we divide the computation process into two parts: *Smartphone* and *Cloud*.

Smartphones are primarily used for storing, computing and communicating with the cloud. They are also being used to get real-time notifications or feedback. SQLite and flat files are used to store the dataset. One of the problems while performing the computation in the smartphone is that it is not possible to come up with the cumulative picture. It is because, all the processing is done on the basis of the data collected in the same smartphone. But for better accuracy and to get the aggregated view of the entire city, we need cloud computing here.

The processing in the cloud is done through the interaction of the following components.

(1) *Database:* Filtered data collected from all the subsystems in the smartphone are ingested into the cloud. Database is used for storing the data in more organized way to ease the data fetching. This component is also used to store the results.

(2) *Data Storage:* This component is used to store the files (like csv, json, text, image, audio, etc.) that are collected and processed in smartphone during participatory sensing.

(3) *API Gateway:* In JUSense architecture (Fig. 3.3), this component is responsible to provide an API management solution. It appears as a single entry point into the system and creates APIs, which perform dedicated tasks to communicate with the end user. Using this component we create the REST APIs. Through these APIs, http requests are sent from smartphone applications to the cloud. The event handler in cloud then

initiates necessary processing to access the cloud database and data storage for the required data. After the processing is over, http responses are given back from the event handler to the smart phone applications through these APIs.

(4) *Event Handler:* This is one of the major components of the JUSense architecture and is responsible for processing purpose. It contains individual functions for processing the data of the individual subsystems of JUSense. Additionally, it also contains fusion functions that use the output data of more than one subsystems (or from more than one sources) and analyse it to extract more statistical information. Event handler runs these functions when events are triggered from the aforementioned components and takes the data from database and data storage as required. Based on the triggered event it may store the results in the database for future use or it can provide real time response to the front end user using the API.

(5) *Machine Learning and Predictive Analysis:* Among the back-end services that are being provided, this component is responsible for the purpose of Machine Learning and Predictive Analysis. When some predictive analysis tasks are required to be performed, this component gets triggered through the event handler. Here, context aware data cleaning model of this component is used for pre-processing task to remove those sensor readings that are not coming from appropriate sensing contexts 3.4.3. The data fusion model is used to make inference of pollution levels from spatial as well as the temporal views of the pollution datasets (see Section 3.4.4). For different applications, the map generation model is used to make visualization of the predicted results on the map. Stay point detection and trajectory mining are mainly related to SenseDCity application that is used to explore human mobility pattern. The road condition detection and driving pattern detection models are related to D&RSense application.

The mutual interaction of the subsystems will not result in any additional lag in the individual applications since the exchange of information will take place in the cloud. Hence, the overall system of JUSense has its processing done in two major phases, the smartphone and the cloud. Real-time response is provided to the end user through smartphone processing as well as cloud processing. The purpose of processing within smartphone is to provide quick and real-time responses even in the absence of internet connection. It alerts and keeps track of an individual's pollution and travel footprints without focusing much on accuracy. Rather, it concentrates on giving the fastest possible response with an acceptable accuracy by implementing algorithms and methods of low computational complexity. Users can also directly send queries to the cloud and get real-time responses against that query. Events get triggered from the smartphone through REST API and the event handler in the cloud does the required lightweight processing. Having only cloud to do all the processing would result in high accuracy. Because we don't have to worry about the resources in cloud. But, the response time would suffer, especially with bad Internet connectivity. Processing done in the smartphone will successfully handle the processing to a certain extent even without an active Internet connection. So, the required trade off in the two phases could be made by proper

distribution of processing between the smartphone and the cloud to get quick responses and high accuracy whenever desired.

3.4.3 Pre-processing

All of the four applications use participatory sensing data. Following pre-processing steps need to be performed on the participatory sensing data before any kind of analysis.

Context-aware data cleaning

One of the problems with participatory sensing is that the dataset may contain noisy data. It is because, this type of sensing uses smart hand held devices as sensors, and the sensor reading from those devices vary based on the sensing context (e.g. whether the smartphone is in bag or in hand, phone orientation, etc.). JUSense performs context-aware data cleaning as one of the pre-processing steps for removing those erroneous sensor values that are not recorded in appropriate sensing context.

A classifier is built to correctly identify the sensing context in which the sensor data had been recorded. In order to do that, we examine the feasibility of smartphone equipped sensor data (like accelerometer data, gyroscope data, light sensor value, proximity sensor value, etc.) as the features for sensing context classification. For this purpose, a total of six volunteers are appointed to record the sensor readings using smartphones. Two identical smartphones are provided to each of the volunteers. They are asked to carry one smartphone in hand and the other in pocket/bag. Here, each of the volunteers is asked to acquire 60 minutes data. After studying these sensor data we choose to exclude light sensor, it is because the light sensor values in two sensing contexts are not significantly different. The data from three axis accelerometer, the proximity sensor and three axis gyroscope are used as features for context classification. Naive Bayes algorithm is used for context classification. We choose Naive Bayes as it can not only predict the appropriate context in a fair accuracy (81%) but also easy to interpret and lightweight compared to other effective machine learning algorithms like support vector classification. Here, leave-one-out cross validation (LOOCV) approach is used for calculating the classification accuracy.

Calibration

IOT devices and smartphones are used in JUSense for participatory sensing based air and noise pollution monitoring. The accuracy of the pollution map depends on the quality of participatory sensing based air and noise pollution sensor data. But, the use of low-cost IOT devices and smartphones for air and noise pollution data collection do not provide accurate high quality measurements. Hence, it is essential to calibrate sensor devices for predicting actual measurements from the sensor responses. In order to do that, it is required to compare sensor readings and actual ground truth measurements. This helps us to find an offset ζ. Later, this offset ζ needs to be added with measured pollution level to calibrate low-cost sensors with standard pollution measuring device. For example, in case of NoiseSense application, the modified equivalent continuous sound level (say, L'_{eq}) is given by modifying Eq.

3.11 as follows:

$$L'_{eq} = 10\log_{10}\left[\frac{1}{t_2 - t_1}\int_{t_1}^{t_2}\left(\frac{P_t}{P_o}\right)^2 dt\right] + \zeta \tag{3.12}$$

In this work, in order to estimate the calibration offset ζ we measure and compare the the readings from standard pollution monitoring device and our low-cost sensors under different pollution levels. Suppose, $S_m = \{v_s^1, v_s^2, \ldots, v_s^n\}$ is a set of low-cost sensor responses corresponding to ground truth measurements $G_m = \{v_g^1, v_g^2, \ldots, v_g^n\}$. Where, $v_g^i \in G_m$ is a grund truth measurement corresponding to its low-cost sensor response $v_s^i \in S_m$. Now, the calibration offset is calculated as follows:

$$\zeta = \frac{1}{n}\sum_{i=1}^{n}(v_g^i - v_s^i) \tag{3.13}$$

Note that in the case of air pollution dynamics, sometimes we need to use static stations' data as ground truth to validate the data quality. To evaluate the quality of the participatory sensing based air pollution data, we have used the data of static air pollution monitoring stations. However, missing values, which are frequently brought on by hardware breakdowns or data corruption, are a serious problem in environmental pollution related research. In the case of such an issue we use efficient predictive model and the site selection method that can be used to perform the task of missing value imputation from multi-site time series [19].

Map-matching

In this pre-processing step, the raw sampling location data (i.e. locations from where the sensor data samples had been collected) need to be placed onto the existing road networks or POIs (e.g. universities, restaurants, playgrounds, shopping malls, parks, etc.). In JUSense, an interactive-voting based method [139] is used to perform the task of map matching. Map-matching first map GPS sampling locations to corresponding road segments and POIs and after that combine all of them.

Suppose $T = l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow \ldots \rightarrow l_n$ is a trajectory denoting a sequence of sampling locations (l_i). Also, let us consider $R = \{r_1, r_2, \ldots, r_m\}$ is a set of road segments and $P = \{p_1, p_2, \ldots, p_k\}$ is a set of POIs. Now, the map matching consists of following steps:

(1) *Candidate preparation:* For each location l_i we create a set of candidate points $CP_i = \{c_1^i, c_2^i, \ldots, c_t^i\}$ where, a candidate point c_j^i is either a projection of l_i on any road segment of R that is within distance d from l_i or it can be a POI of P within that distance. Now, for each candidate point $c_j^i \in CP_i$, its observation probability with respect to location l_i is computed as follows [140]:

$$X(c_j^i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_j^i - \mu)^2}{2\sigma^2}} \tag{3.14}$$

where, d_j^i denotes the distance of $c_j^i \in CP_i$ to the sampling location l_i. Now a spatial and temporal analysis function (also known as *ST-function*) [141] is used to create a candidate graph.

(2) *Score matrix building:* Next, for each location l_i of the trajectory T a score matrix is built [139]. Where, each item of the score matrix indicates the probability of candidate point $c_j^i \in CP_i$ to be a correct projection. Now, for all the candidate points c_j^i of l_i, first a trajectory path $P_j^i : c_s^1 \rightarrow \ldots c_j^i \ldots \rightarrow c_q^n$ of candidate points is found that has the largest score summation.

(3) *Interactive voting:* Next, the candidate point c_j^i votes for all the candidate points on the path P_j^i [139]. Suppose the votes obtained by the candidate points of location l_i is denoted by the set $V^i = \{v_1^i, v_2^i, \ldots, v_t^i\}$, where, v_j^i is the number of votes obtained by candidate point $c_j^i \in CP_i$. After the voting, we find the candidate point $c_k^i \in CP_i$ who has the most votes for each sampling location l_i i.e. $v_k^i = max(V^i)$. Such a candidate point is known as winner candidate point $c^{i.(win)}$.

(4) *Path finding:* Finally, the winner candidate points are just linked one by one to find the map matched path. Hence, the optimal map matched path for the trajectory $T = l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow \ldots \rightarrow l_n$ will be $P^T = c_1^{(win)} \rightarrow c_2^{(win)} \rightarrow c_3^{(win)} \ldots \rightarrow \ldots c_n^{(win)}$.

3.4.4 Data fusion

In JUSense, there are different datasets (like air quality data, ambient noise data, POIs data, human mobility data, road networks data, meteorological data etc.) from different sources. Each of these datasets has a different representation, characteristic, distribution and scale. In our work, we use the concept of data fusion to discover new knowledge (or improve our existing knowledge) from more than one different datasets (but potentially connected) or from different views of the same dataset.

In order to create more accurate air and noise pollution map at high spatio-temporal resolution, the idea of multi-view data fusion is used. Each of the air and noise pollution datasets can have two distinct views *spatial view* and *temporal view*. *Spatial view* of a dataset shows how pollution levels are distributed over the space. On the other hand, *temporal view* of a dataset describes temporal variability of pollution levels at a particular POI (Point Of Interest). Fig. 3.4 shows how the pollution levels of a particular POI can be better predicted using spatial and temporal views of the dataset. Pollution levels of a POI P_j at time T_k is predicted using an interpolation technique (in our case, IDW) based on the spatial view of the dataset. Similarly, the prediction of pollution levels of the POI P_j at time T_k can also be done based on the temporal view using IDW on a single dimensional temporal space. The two predictors make their predictions $PolLvl_{T_k}^{(sv)}(P_j)$ and $PolLvl_{T_k}^{(tv)}(P_j)$ independently based on spatial and temporal view respectively for the POI P_j at time T_k as follows.

$$PolLvl_{T_k}^{(sv)}(P_j) = \begin{cases} \frac{\sum_{i=1}^{N} W_{ji}^{(s)} V_{i.T_k}^{(s)}}{\sum_{i=1}^{N} W_{ji}^{(s)}}, & \text{if } dist(P_j, S_i) \neq 0 \text{ for all } i \\ V_{i.T_k}^{(s)}, & \text{otherwise} \end{cases} \tag{3.15}$$

$$W_{ji}^{(s)} = \frac{1}{dist(P_j, S_i)^{\beta}} \tag{3.16}$$

Here, the impact of pollution on P_j due to the point S_i is expressed by the weight $W_{ji}^{(s)}$. The distance between S_i and P_j is $dist(P_j, S_i)$ which is simple 2-D spatial euclidean distance and β is the power parameter. $V_{i.T_k}^{(s)}$ indicates the observed value at point S_i at time T_k. Similarly, from temporal view, the pollution level can be predicted as follows

$$PolLvl_{T_k}^{(tv)}(P_j) = \begin{cases} \frac{\sum_{i=1}^{M} W_{ki}^{(t)} V_{j.T_i}^{(t)}}{\sum_{i=1}^{M} W_{ki}^{(t)}}, & \text{if } dist(T_k, T_i) \neq 0 \text{ for all } i \\ V_{j.T_i}^{(t)}, & \text{otherwise} \end{cases} \tag{3.17}$$

Where,

$$W_{ki}^{(t)} = \frac{1}{dist(T_k, T_i)^{\beta}} \tag{3.18}$$

Here, $dist(T_k, T_i) = \sqrt{(T_k - T_i)^2}$ is 1-D euclidean distance in temporal space. $V_{j.T_i}^{(t)}$ indicates the observed value at time T_i at P_j. After that, these two predictions are combined to make an overall prediction at POI P_j. A weighted average of these two predictions are taken in order to compute the overall prediction at the POI as shown below.

$$PolLvl_{T_k}^{(Overall)}(P_j) = \xi^{(sv)} PolLvl_{T_k}^{(sv)}(P_j) + \xi^{(tv)} PolLvl_{T_k}^{(tv)}(P_j) \tag{3.19}$$

In Eq. 3.19, $PolLvl_{T_k}^{(sv)}(P_j)$ and $PolLvl_{T_k}^{(tv)}(P_j)$ are the predicted pollution levels at P_j from spatial and temporal views respectively. Here, $\xi^{(sv)}$ and $\xi^{(tv)}$ are the weights applied to spatial and temporal predictions that satisfy the condition $\xi^{(sv)} + \xi^{(tv)} = 1$. The best setting of $\xi^{(sv)}$ and $\xi^{(tv)}$ is one for which MVDF yields best accuracy (in terms RMSE value). In this work, MVDF yields best performance for the setting $\xi^{(sv)} = 0.6$ and $\xi^{(tv)} = 0.4$.

Here, the details of the fusion model for predicting pollution levels at any POI is presented in Algorithm 1. The input to the algorithm are sample data $\mathcal{D}$, a set $\mathcal{P}$ of POIs for prediction, the time of prediction T_k. For spatial prediction we create a dataset $\mathcal{D}_{T_k}^{(s)} \subseteq \mathcal{D}$ having samples with sampling time T_k. Now, for each POI $P_j \in \mathcal{P}$, we first predict the pollution level $PolLvl_{T_k}^{(sv)}(P_j)$ from the spatial view (line 5-line 10). Now, for temporal prediction at POI P_j we create a dataset $\mathcal{D}_{p_j}^{(t)}$ that contains samples at P_j having different sampling times. Next, for POI $P_j \in \mathcal{P}$ we predict the pollution level $PolLvl_{T_k}^{(tv)}(P_j)$ from the temporal view (line 11-line 17). After calculating the pollution level for all POIs $P_j \in \mathcal{P}$ we combine them to find the overall pollution level (line 19-line 22).

In this chapter, the concept of data fusion is also used to define an index called Overall Pollution Index (OPI) to measure an overall pollution level by combining air and noise pollution levels at any POI. Different countries have their own national air and noise pollution standards. Depending on the standards, the number of pollution categories (e.g. good, satisfactory, moderately polluted, etc.) and their ranges can be different. We propose a simple two step method to calculate OPI at a particular time of a POI from Air Quality Index (AQI) value and noise value of that POI at that time. This method works for all national standards.

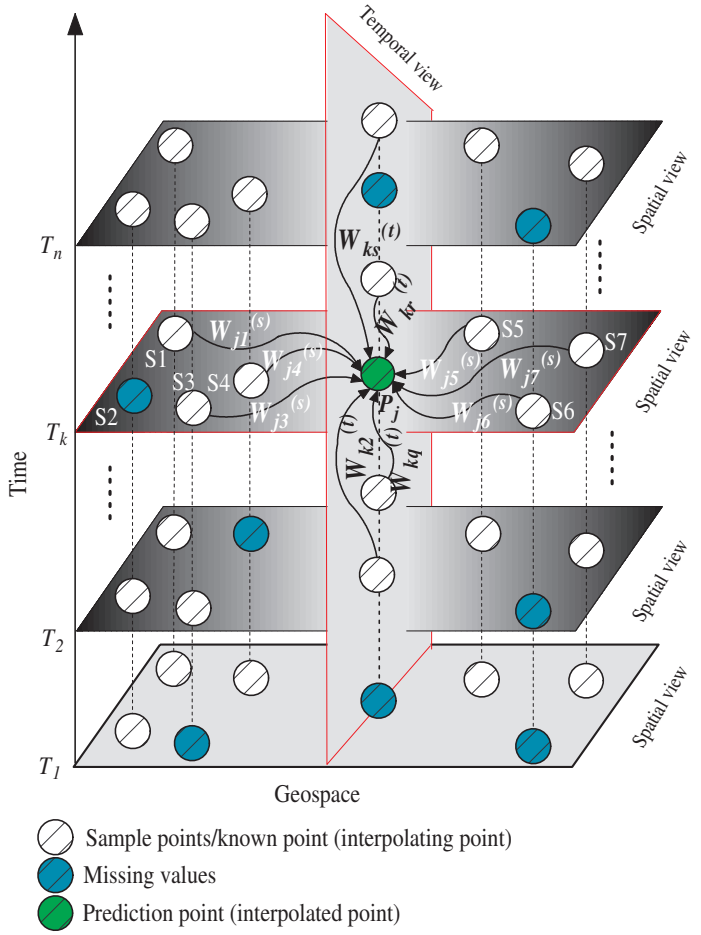


FIGURE 3.4: Multiview data fusion: The pollution level of the POI P_j (shown in green) at time T_k is predicted using spatial view and temporal view.

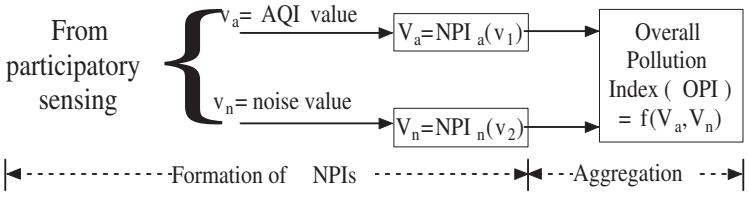


FIGURE 3.5: Formation of Overall Pollution Index (OPI).

As shown in Fig. 3.5, the two steps that are involved to find overall pollution index (OPI) of a POI are as follows:

- Formation of Normalized pollution Index (NPI) for air/noise pollution.
- Aggregation of NPIs to get an Overall Pollution Index (OPI) of a POI.

In order to explain this two step method for any arbitrary pollution standard, let us consider, $\mathscr{C}_i$, $I_i^{(low)}$ and $I_i^{(high)}$ denote i^{th} pollution category, the lowest permissible value in $\mathscr{C}_i$ and the highest permissible value in $\mathscr{C}_i$ respectively.

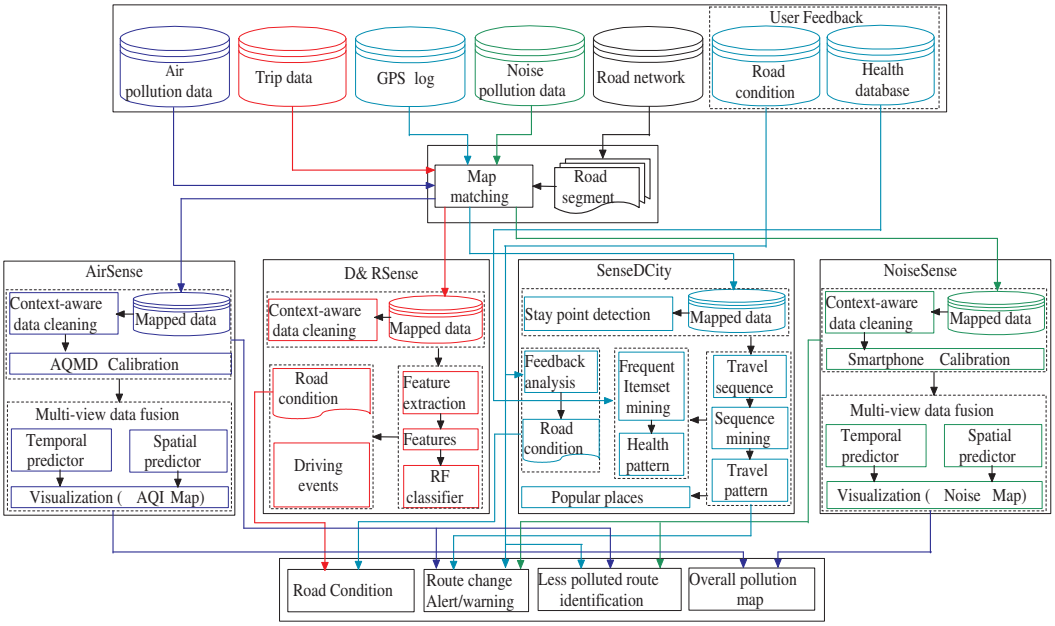


FIGURE 3.6: Interaction among different components of the subsystems in JUSense

Formation of NPI: NPI is formulated as a piecewise linear function of AQI/Noise values. To convert a AQI/Noise value (say, v) of any standard to NPI, the following function is defined.

$$NPI(v) = \frac{\left(v - I_i^{(low)}\right)}{N\left(I_i^{(high)} - I_i^{(low)}\right)} + \frac{(i-1)}{N} \tag{3.20}$$

where, $v \in \left[I_i^{(low)}, I_i^{(high)}\right]$ and N is the total number of pollution categories.
Aggregation of NPIs to get an OPI: Once, the NPIs are found, they are combined in a simple weighted additive form as follows:

$$OPI = w_a NPI_a(v_a) + w_n NPI_n(v_n) \tag{3.21}$$

$$\text{where, } w_a + w_n = 1$$

Here, $NPI_a(v_a)$ and $NPI_n(v_n)$ indicates NPIs of air and noise pollution levels v_a and v_n respectively. On the other hand, w_a and w_n are the weights associated with air and noise pollution levels respectively. Later, in Section 7.5, the spatial variability of OPIs of Jadavpur area (a part of city Kolkata) is shown in the form of a pollution map.

On the other hand, in the tasks like road condition monitoring or detection of rash driving prone zones, fusion of data generated from multiple applications (e.g. SenseDCity and D&RSense) help us in identifying road conditions more accurately. In Section 3.5.4, a case study is provided to show that the mutual interaction among different applications can achieve better accuracy than individual applications.

Algorithm 1 Data Fusion ($\mathcal{P}$, $\mathcal{D}$, T_k)

Input: Set of POIs for prediction $\mathcal{P}$, Sample data $\mathcal{D}$, Time of prediction T_k.
Output: POIs' predicted pollution levels $\mathcal{P}^{(pred)}$.

$\mathcal{P}^{(pred)} \leftarrow \emptyset$, $\mathcal{P}_{sv}^{(pred)} \leftarrow \emptyset$, $\mathcal{P}_{tv}^{(pred)} \leftarrow \emptyset$
$S_i \leftarrow$ location of pollution sample data $d_i \in \mathcal{D}$
Create a set $\mathcal{D}_{T_k}^{(s)} = \{d_i | d_i \in \mathcal{D}$ having sampling time $T_k\}$
for each POI $P_j \in \mathcal{P}$ **do**
  if $dist(P_j, S_i) \neq 0$ for all $d_i \in \mathcal{D}_{T_k}^{(s)}$ **then**
    $PolLvl_{T_k}^{(sv)}(P_j) = \frac{\Sigma_{i=1}^{N} W_{ji}^{(s)} V_{i.T_k}^{(s)}}{\Sigma_{i=1}^{N} W_{ji}^{(s)}}$;
  else
    $PolLvl_{T_k}^{(sv)}(P_j) = V_{i.T_k}^{(s)}$;
  end if
  $\mathcal{P}_{sv}^{(pred)} = \mathcal{P}_{sv}^{(pred)} \cup PolLvl_{T_k}^{(sv)}(P_j)$
  Create a set $\mathcal{D}_{p_j}^{(t)} = \{d_i | d_i \in D$ and $P_j = S_i\}$
  if $dist(T_k, T_i) \neq 0$ for all $d_i \in \mathcal{D}_{p_j}^{(t)}$ **then**
    $PolLvl_{T_k}^{(tv)}(P_j) = \frac{\Sigma_{i=1}^{M} W_{ki}^{(t)} V_{j.T_i}^{(t)}}{\Sigma_{i=1}^{M} W_{ki}^{(t)}}$
  else
    $PolLvl_{T_k}^{(tv)}(P_j) = V_{j.T_i}^{(t)}$;
  end if
  $\mathcal{P}_{tv}^{(pred)} = \mathcal{P}_{tv}^{(pred)} \cup PolLvl_{T_k}^{(tv)}(P_j)$
end for
for each POI $P_j \in \mathcal{P}$ **do**
  $\mathcal{P}_{j}^{(pred)} = \xi^{(sv)} \mathcal{P}_{sv.j}^{(pred)} + \xi^{(tv)} \mathcal{P}_{tv.j}^{(pred)}$
  $\mathcal{P}^{(pred)} = \mathcal{P}^{(pred)} \cup \mathcal{P}_{j}^{(pred)}$
end for
return $\mathcal{P}^{(pred)}$

3.4.5 Interaction among Components

JUSense provides a unified framework to combine the subsystems (e.g. AirSense, NoiseSense, D&RSense and SenseDCity). It is effective in terms of accuracy (see section 3.5.4) which is achieved with the help of mutual interactions among the subsystems. The detailed view of the components of each of the subsystems and the interactions among those components within the subsystems and between subsystems are shown in Fig. 3.6. The subsystems use the real world datasets (e.g. air and noise pollution data, trip data, GPS logs, road network data, users' feedback, etc.). The Fig. 3.6 shows that the individual subsystems can provide some services on their own. For example, AirSense and NoiseSense applications generate air pollution map (AQI map) and noise pollution map respectively by using a multi-view data fusion technique as given in Algorithm 1. D&RSense uses Random Forest (RF) classifier to recognizes driving events (e.g. acceleration, braking, etc.) and road conditions (e.g. bumps, potholes, etc.) by using smartphone equipped sensors' values (e.g. 3-axis accelerometer data, 3-axis gyroscope data, etc.) as features. The SenseDCity produces users' travel pattern from their trajectories (sequence of GPS locations. SenseDCity also helps to collect users feedback about personal health conditions and city's road conditions. Frequent itemset mining is performed on users' travel pattern and health data to identify their health pattern. On the other hand, feedback analysis is performed on road condition related users' feedback to identify current condition of roads at different parts of the city. Travel patterns from all the users are analyzed to identify the popular places of the city. Each of the subsystems first uses map-matching module to map GPS sampling locations to corresponding road segments and POIs and after that combine all of them to create application specific mapped data. For NoiseSense, AirSense and D&RSense, context aware data cleaning is required to filter out data samples that are not from appropriate sensing contexts. AQMD calibration and smartphone calibration are used for AirSense and NoiseSense applications respectively.

Additionally, aggregated information from two or more subsystems are used to make new inferences. The information aggregation from more than one subsystems can help us improving existing services by providing higher accuracy. For example, air and noise pollution maps can be combined to provide an overall pollution map of the city. Similarly, D&RSense and SenseDCity can detect road conditions of the city individually. The results from these two are combined to get a bigger and more complete picture of the city's road conditions. Other combined services like *route change alert* and *less polluted route identification* is performed by analyzing users' travel patterns (from SenseDCity) and pollution maps (from AirSense and NoiseSense) of the city. Table 3.1 lists all the services (individual and combined) provided by JUSense. For each of the services, it shows the subsystem(s) responsible for providing that service.

3.5 Evaluation

In this section, we experimentally evaluate the performance and effectiveness of the proposed system JUSense. We perform large scale experiments on real world participatory sensing based datasets. At the beginning, we explain the participatory sensing based real world data

TABLE 3.1: Individual and combined services provided by JUSense

Service type	Service name	Societal applications				Figure(s)
		AirSense	NoiseSense	D&RSense	SenseDCity	
Individual service	Air Quality Index (AQI) map	✓				Fig. 3.10
	Noise pollution map		✓			Fig. 3.11
	Driving events detection			✓		Fig. 3.13
	Road Condition detection			✓		Fig. 3.19 (b)
	Travel pattern detection				✓	Fig. 3.20 (a)
	Road Condition detection				✓	Fig. 3.15 (a)-(c)
	Health pattern detection				✓	Fig. 3.14 (a)
	Popular places identification				✓	Fig. 3.14 (b)
Combined service	Road Condition detection			✓	✓	Fig. 3.19 (c)
	Route change alert/warning	✓	✓		✓	Fig. 3.20 (d)
	Less polluted route identification	✓	✓		✓	Fig. 3.17
	Overall pollution map	✓	✓			Fig. 3.16

✓: The presence of check mark indicates that a service that is shown in a particular row of the table is provided (or jointly provided) by the application in the associated column.

collection and data preparation in order to perform the experiments. Next, we provide the metrics used to evaluate the performance of the methods. Finally, we provide the experimental results that include individual and combined services provided by the applications, the comparison with baselines and case studies to show the effectiveness of JUSense in terms of accuracy which is achieved with the help of mutual interactions among the subsystems.

3.5.1 Study Area and Data Collection

The study area chosen for the study is Jadavpur area of city Kolkata (West Bengal, India). It is extended between $22°48'91''N - 22°50'70''N$ latitude and $88°36'76''E - 88°40'07''E$ longitude covering an area of 6 km^2. The region is densely populated and having varying landuse. The study area is shown in Fig. 3.7. Data are collected for all the four applications (AirSense, NoiseSense, SenseDCity and D&RSense) separately in the study area. Volunteers of different ages, genders and professions are participated in the data collection process. As mentioned earlier, smartphone sensors and IoT devices (e.g., AQMD device in AirSense) are used as a sensing device for these applications. Along with this, we have used the concept of *Citizen as Sensor*. The contexts of the users as well as their health profiles and the ambient atmospheric information are taken as an input from the users to better understand the effectiveness of these services.

For AirSense and NoiseSense applications data were collected outside of Jadavpur University area (Sulekha crossing → 8B Bus stand → Jadavpur Police Station → Abhishikta crossing → Ajaynagar crossing → Sulekha crossing) and also inside the main campus. Fig. 3.8 (a) shows a working prototype of Air Quality Monitoring Device (AQMD) used for air quality data collection using the AirSense application. Fig. 3.8 (b) and Fig. 3.8 (c) show that the participants are collecting air and noise pollution data using AQMD and smartphone respectively. In this experiment, three time intervals of a day (9am-11am in the morning, 2pm-4pm at afternoon and 6pm-8pm at evening sessions) are considered for collecting data. Here, our main motive behind selecting these three time intervals is to monitor air and noise

pollution levels during the rush hours in the city. It is important to study ambient pollution levels during the rush hours in the city because the previous studies [142] [143] have showed that the individuals' daily exposure to air and noise pollution is usually critical during the rush hours.

Along with these applications, data are also collected for SenseDCity and D&RSense applications. In D&RSense, data were collected for bike ride along the route from Jadavpur University Main Campus to Baruipur via Baruipur Bypass (to and fro journey) over an approximate distance of 50 km and around Jadavpur area for 4-5 days a week. Templates for all the driving events like normal and sudden acceleration, braking, turns and road conditions (potholes and bumps) were recorded and pushed to cloud and also processed locally (i.e., smartphone). The normal and sudden acceleration templates were created while the speed was increased from 0 to 20, 0 to 40, 0 to 60, 20 to 40, and 40 to 60 kmph. The braking templates were created while the speed was decreased in the opposite direction of all the previously mentioned values.

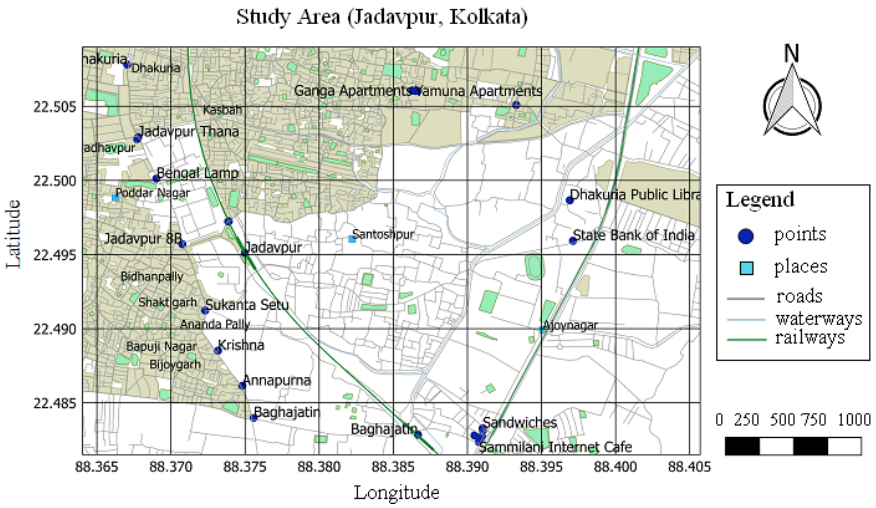


FIGURE 3.7: Study Area: Jadavpur area, Kolkata



FIGURE 3.8: (a) Air quality monitoring device (AQMD) (b) A participant is using AirSense application along with AQMD (c) A participant is using NoiseSense application

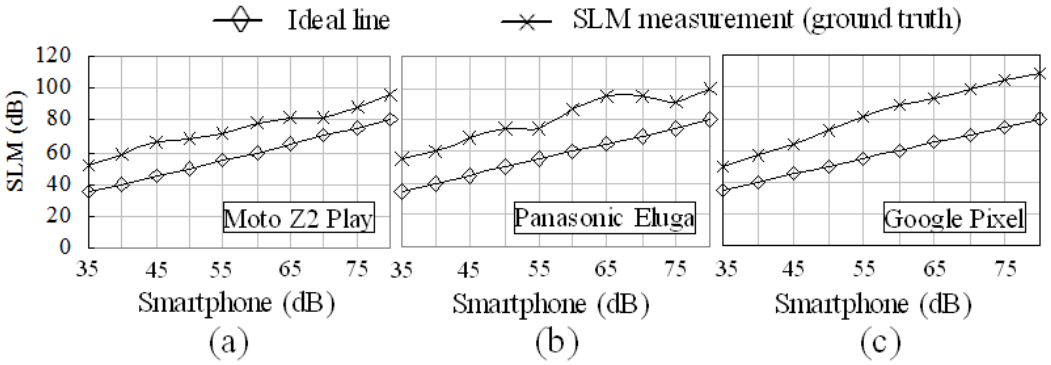


FIGURE 3.9: Calibration for NoiseSense for different Smartphones with respect to standard sound level meter (SLM). Here, the ideal line shows a line where the smartphone responses and corresponding SLM measurements should agree.

3.5.2 Data preparation

Now, as discussed in Section 3.4.3, data preparation (i.e. pre-processing) is required to improve the quality of raw sensor data for further analysis and processing. In this work, data cleaning, calibration and map-matching are performed on the raw data for this purpose.

For data preparation, the proper calibration of low-cost sensors is an essential part as the low-cost sensors that are used for sensing may not provide accurate measurements. As given in Section 3.4.3, it is required to calculate a calibration offset ζ for accurate measurement. Now, to estimate the calibration offset ζ, it is required to compare the low-cost sensor readings (e.g. sound levels, PM2.5 values, PM10 values, etc.) and actual ground truth measurements of standard devices. For AirSense application, AQMD values are compared with standard pollution measuring devices like IQAir AirVisual Pro Air Quality Monitor (Atlanta Healthcare) [144] and Forbix Semicon [145]. In NoiseSense, Amprobe SM10 and Mextech SL-4012 sound level meter are used for ground truth reference. For D&RSense application, the compass of a smartphone is calibrated with google maps for accurate sensing and data collection. One thing to be noted that, different manufacturer of smart handheld devices may use different sensors in their smart handhelds. Hence, the calibration of these different handhelds are also different. For example, in case of NoiseSense application, Fig. 3.9 shows different calibrations for three different smartphone models (MotoZ2 Play, Panasonic Eluga and Google Pixel). The upper curve on Fig. 3.9 shows the ground truth measurements of standard sound level meter (SLM) compared to the responses (i.e. L_{eq} value) of the smartphone. Now, the required calibration offset ζ is calculated using Eq. 3.13 that needs to be added with measured responses to calibrate low-cost sensors with standard measuring devices. For some applications, JUSense also employ context aware data cleaning technique for data preparation. As explained in Section 3.4.3, the data coming from inappropriate sensing contexts are removed by a light weight Naive Bayes classifier. For example, in case of NoiseSense application, if the phone is in bag or pocket during data collection then those sound data are removed by detecting the sensing contexts. Also, raw GPS location data may suffer from positional errors. Now, to infer the location more accurately an existing map matching technique [139] is used.

3.5.3 Performance metrics

In this work, cross validation approach is used to measure the accuracy of our proposed methods. For AirSense and NoiseSense applications, two common statistical measures like mean absolute error (MAE) and root mean squared errors (RMSE) are used to evaluate the prediction accuracy of our proposed Fusion based method for air and noise pollution monitoring. Here, MAE and RMSE are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Z(p_i) - \widehat{Z}(p_i)| \tag{3.22}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\{Z(p_i) - \widehat{Z}(p_i)\}^2} \tag{3.23}$$

Where, N represents the total number of samples in the validation set, $Z(p_i)$ and $\widehat{Z}(p_i)$ represent the measured and predicted pollution levels respectively. Now, in case of driving events detection using D&RSense application, the classification accuracy is used to measure the performance of the Heroku deployed Random Forest (RF) model. The classification accuracy is defined as follows:

$$Acc = \frac{E^{(correct)}}{E^{(total)}} \times 100\% \tag{3.24}$$

Where, $E^{(correct)}$ and $E^{(total)}$ denote the number of correctly classified events and the total number of events respectively.

3.5.4 Results

The results of various experiments are structured as follows: at the beginning, the individual services as well as combined services are presented. Next, the performances of the predictive models used to provide those services reported are compared with some baseline methods. Finally, we conduct case studies to show the effectiveness of mutual interaction among different subsystems of JUSense.

Both AirSense and NoiseSense applications perform spatial distribution analysis of urban air and noise pollution levels using the proposed fusion model. The best setting for model parameters β, $\xi^{(sv)}$ and $\xi^{(tv)}$ of the fusion model are: $\beta = 2$, $\xi^{(sv)} = 0.6$ and $\xi^{(tv)} = 0.4$. Now, the Fig. 3.10 shows the spatial variability of air pollution levels in the study area (i.e. Jadavpur area, Kolkata) in February, 2019. It is observed that the landuse types like local market, road side eateries, road crossing, industrial areas, etc mainly suffer from relatively high air pollution. Similarly, Fig. 3.11 shows the spatial variability of noise pollution levels in the study area. It is found that the noise pollution levels are much higher near the major road segments. One interesting thing to be observed from Fig. 3.10 and 3.11 is that we didn't find much correlation between air and noise pollution levels in the study area. Another large scale study has been performed to explore diurnal variability of air pollution (in terms of AQI

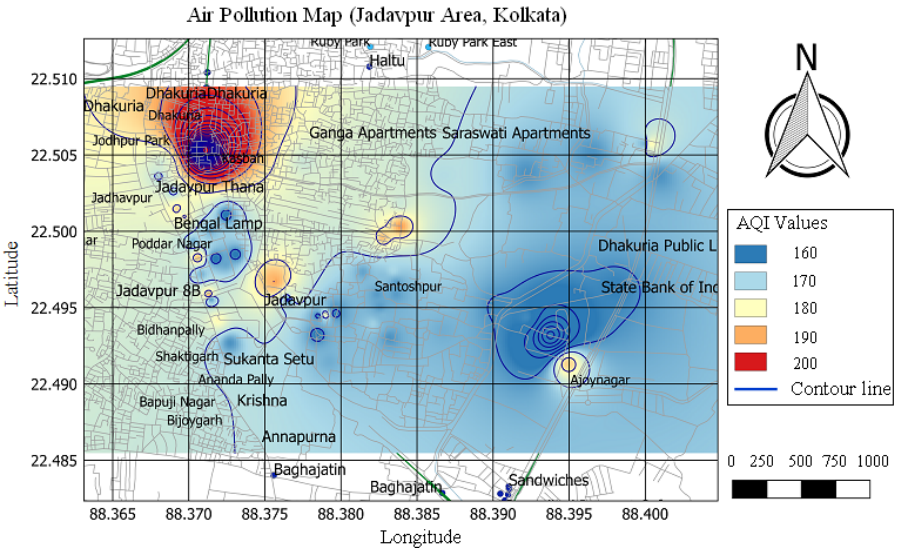


FIGURE 3.10: Air pollution map of Jadavpur area, Kolkata.

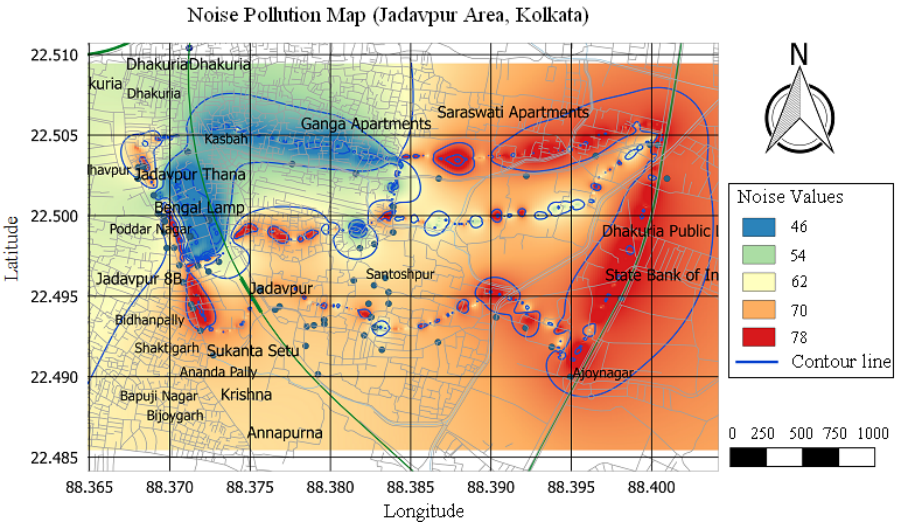


FIGURE 3.11: Noise pollution map of Jadavpur area, Kolkata.

and PM2.5) at different sessions of a day like MS: morning session (09:00-11:00), NS: noon session (12:00-14:00) and AS: afternoon session (15:00-17:00). In order to perform this study the air pollution data collected by AirSense at different landuse types like crossing, universities and hospitals are segregated. The diurnal variation of air pollution is shown in Fig. 3.12. For example, the top left graph of Fig. 3.12 shows box plots of AQI and PM2.5 for three different sessions (MS, NS, AS) for weekday and crossing. The levels of pollution show significant variation during weekdays but during weekends the levels of pollution do not have that much significant variation.

Now, Fig. 3.13 (a) shows a representative result of D&RSense that indicates few rash-driving prone zones in the study area. In order to detect rash-driving prone zones, the Random Forest (RF) classifier is first used to detect driving events like sudden braking, sudden acceleration etc. which works based on smartphones' sensor data. Next, the Density-based spatial

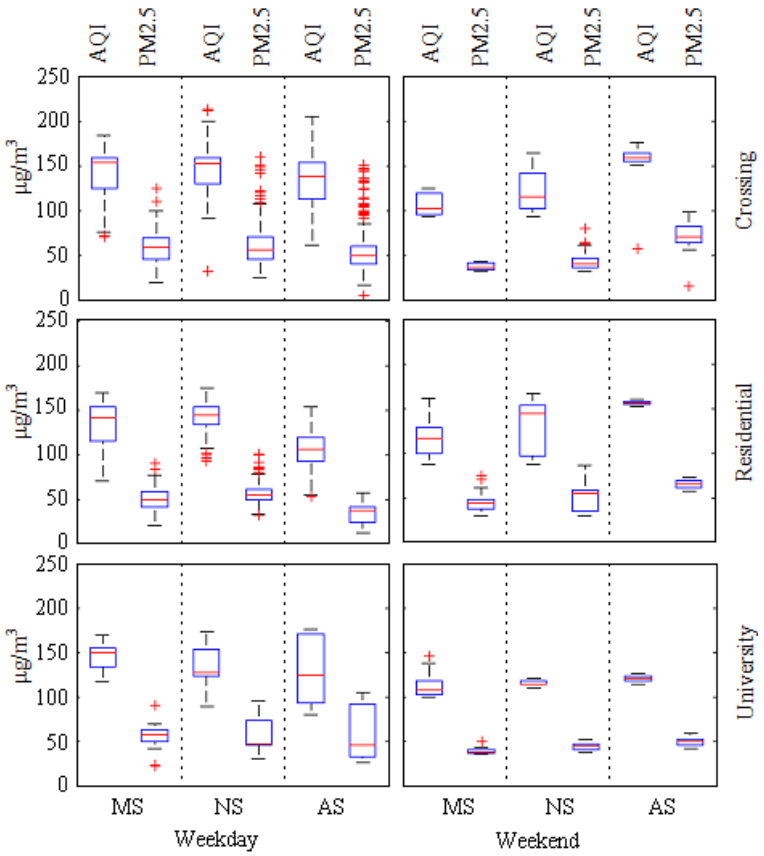


FIGURE 3.12: Diurnal variation of AQI and PM10.

clustering of applications with noise (DBSCAN) clustering is utilized to cluster several locations corresponding to those GPS points where at least 10 aggressive driving incidents continued to occur over a 5-day period of trip data collection at different times of the day. In order to obtain the optimal result, the Random Forest (RF) classifier makes use of following settings of model parameters: Number of trees in the forest=500, The function to measure the quality of a split = "entropy", The maximum depth of the tree = 10. Additionally, D&RSense can report whether over-speeding has occurred or not. It is detected by checking a predefined speed limit with the current driving speed. D&RSense also capable to maintain the locations in the cloud where over-speeding frequently occur. Fig. 3.13 (c) shows such a road segment (marked in red) where over-speeding occurs. Whereas, Fig.3.13 (b) shows a normal driving (within speed limit) trajectory of a user. Now, the common health problems depending on food intake locations can be studied using SenseDCity. Here, SenseDCity application allows the users to share their check-in details to restaurants, canteens or road side eateries for food intake activities. From these data, as shown in Fig. 3.14 (a), SenseDCity can provide a simple visualization for frequently visited food intake locations by the users and the related health issues that may be caused due to food intake from such locations. Similarly, depending on the total number of check-ins to a particular location SenseDCity identifies the popular locations in the study area (as shown in Fig. 3.14 (b)). Also, SenseDCity application allows the users to send the feedback related to poor road conditions with geolocation. Now, as shown in Fig. 3.15, SenseDCity shows a visual representation of road conditions by simply plotting those

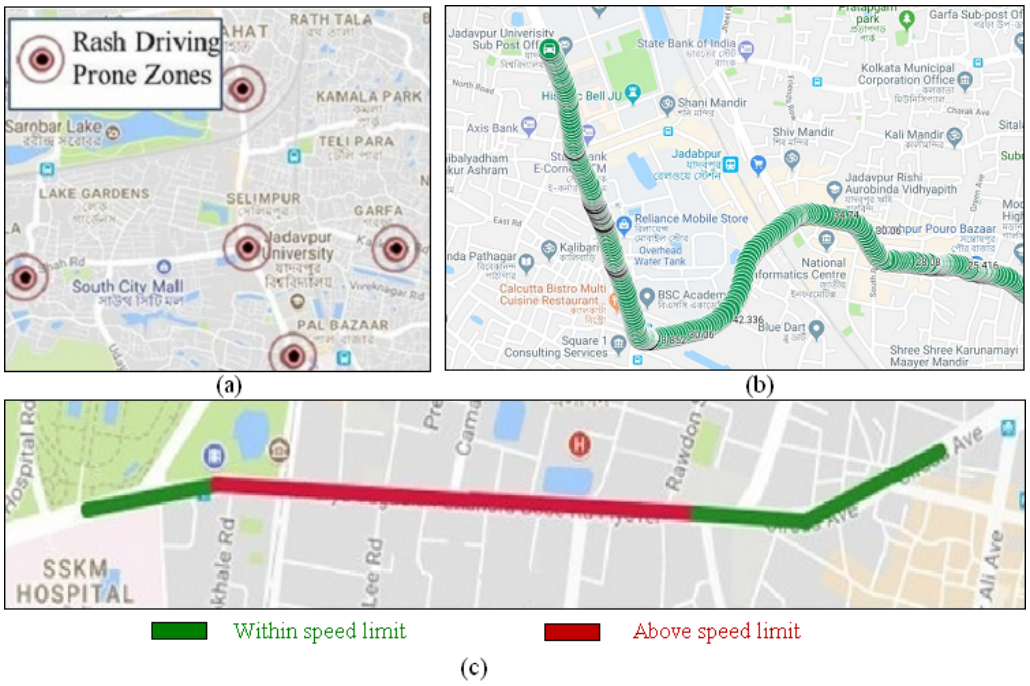


FIGURE 3.13: Results of D&RSense (a) Rash driving prone zone (b) Normal driving (speed $\leq$ 40 kmph) trajectory of a volunteer. (c) Rash driving prone road segment.

locations on the map. Here, black hexagons are the locations where poor road conditions have been identified.

On the other hand, JUSense also provides combined services by mutual interactions among the applications. For example, the pollution data from AirSense and NoiseSense are fused to create an overall pollution map of the city. This is performed by calculating OPI (Overall Pollution Index) at different POIs (see Section 3.4.4). Fig. 3.16 shows overall pollution map of the study area using IDW interpolation technique. The Fig. 3.17 shows another combined service called less polluted route identification. Here, first, the frequent travel paths of a user between a source and destination are found using SenseDCity application. After that, the pollution data that are collected along those paths using AirSense and NoiseSense are analyzed to recommend the less polluted route between that source and destination. The results of other combined services like road condition detection and route change alert/warning are discussed in Section 3.5.4 where the effectiveness of mutual interactions among the applications are presented.

Performance Comparison with Baseline

Here, we compare our proposed models with some baseline methods. In order to assess the performance of the proposed multiview fusion model, it is compared with three baseline methods: Modified Shepard's Method (MSM), Simple Kriging (SK) and Nearest Neighbors (NN) [146] [147]. On the other hand, the performance of the Random Forest (RF) classifier build for D&RSense application is evaluated and compared with three methods like, Naive Bayes (NB), Support Vector Machine (SVM) and Artificial Neural Network (ANN). Fig. 3.18 (a) and (b) show RMSE and MAE of different methods for predicting pollution levels

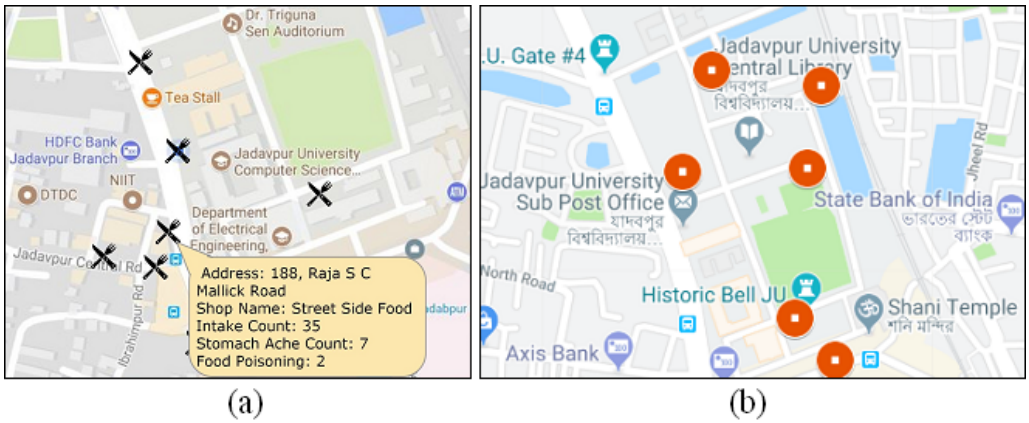


FIGURE 3.14: Results of SenseDCity (a) Common health issues among the volunteers based on location of food intake (b) Most popular locations in Jadavpur region.

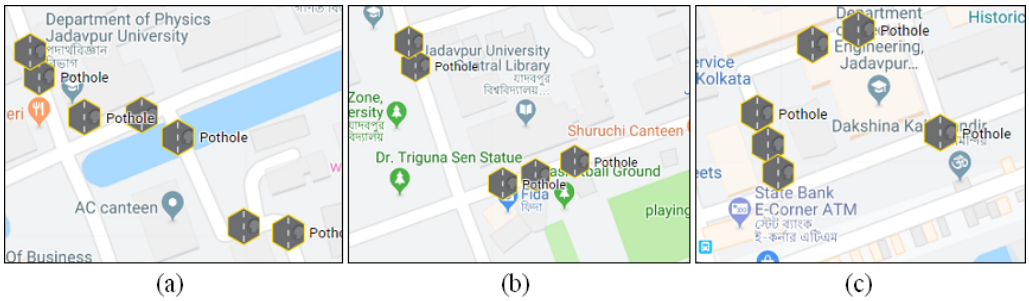


FIGURE 3.15: The road conditions reported by SenseDCity based on citizens' feedback.

respectively. It is clear from Fig. 3.18 (a) and (b) that our proposed fusion method achieves better accuracy in comparison to the baselines (i.e. MSM, SK and NN) in terms of RMSE and MAE. On the other hand the experimental results in Fig. 3.18 (c) shows that Random Forest (RF) dominated over NB, SVM and ANN for driving events detection.

Case studies

In this work, two scenarios are considered to show the effectiveness of the proposed system JUSense in terms of accuracy which is achieved with the help of mutual interactions among the subsystems. The interactions among the subsystems in JUSense for the following scenarios can be better understood from Fig.3.6.

Scenario 1: D&RSense is capable of generating a road condition map of the city (say, road map 1) on its own while SenseDCity also maintains its own road condition map (say, road map 2) with direct input from the citizens. In the case of D&RSense, when a citizen is riding a bike, it may not detect all the bumps or potholes. Similarly, in the case of SenseDCity, a citizen also may not roam the entire city. Merging the maps (road map 1 and road map 2) would result in a bigger and more complete picture of the city's road condition.

Scenario 2: SenseDCity generates individual users travel pattern and maintains their personal health database (User health database). On the other hand, AirSense and NoiseSense maintain dynamic pollution maps of the city using which SenseDCity can notify the user when the environmental conditions are no longer favorable to his health. For instance,

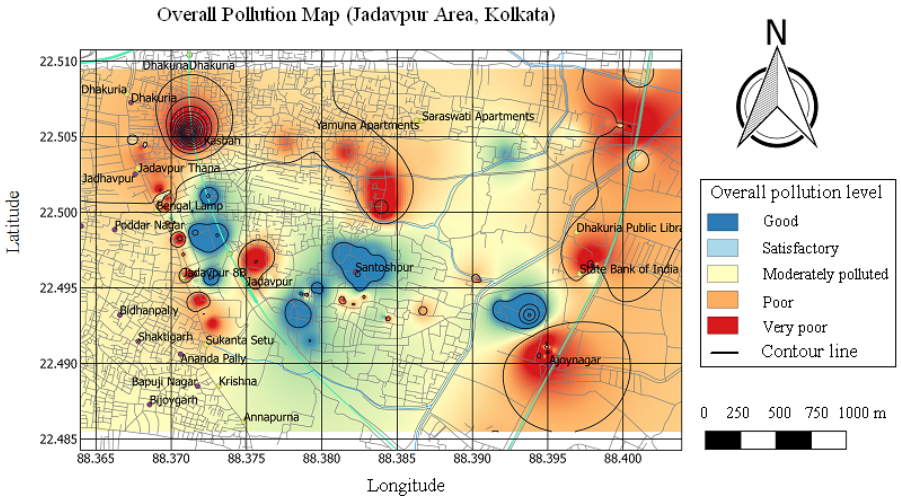


FIGURE 3.16: Overall pollution map of Jadavpur area, Kolkata.

SenseDCity would suggest a person suffering from asthma to change his route when it will receive information from AirSense that the air quality in his regular route has fallen below tolerable limits.

In this experiment, it is found that the collaborative approach works better than any single service and is verified here as illustrated in Fig. 3.19 and Fig. 3.20 respectively. Scenario 1 represents an important service i.e., road condition monitoring. Both D&RSense and SenseDCity are capable to record different road conditions. Both have their own advantages and disadvantages. D&RSense can detect bad road conditions such as bumps and potholes only when the vehicles go through it and the corresponding jerking is noted in the smartphone. So, this detection depends on two factors primarily. The first one is the vehicle and the second one is the smartphone. Practically, different vehicle has different suspension, hence same bump or pothole will be noted differently in different vehicles. Some of the issues of D&RSense experienced during experiments are highlighted here.

- It is often noticed that a pothole which is detected using a motorcycle might not be detected in the same road while traveling with a four wheeler. So, we can say that the performance of D&RSense is vehicle dependent.

- Another challenge is to capture the variation of smartphones in noting the same reading, i.e., different smartphone sensors may sense the environment differently due to dissimilarity in calibration. Thus, same bump/pothole data are recorded in different smartphones differently, i.e., same sensor will store different data readings for different devices.

- Sometimes road conditions might not be properly captured. It is more likely for a driver to avoid a bump or pothole on his way of driving. Hence, even if there are poor road conditions, most of the time it is avoided by the driver, hence it is not reflected and recorded in the D&RSense database. Another reason for not getting all the bumps/potholes throughout the road is that one single car may not cover the whole road segment which is sometimes more than 30 feet wide. So, large scale participatory

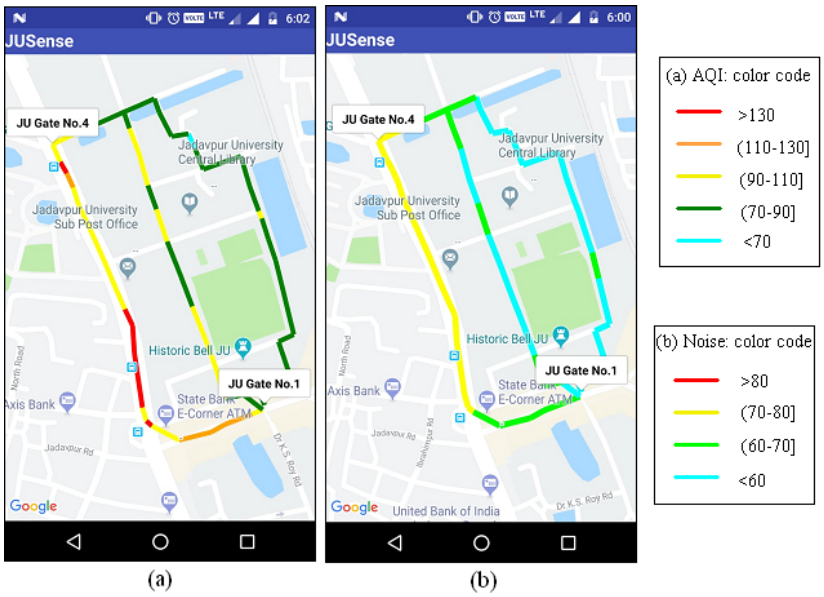


FIGURE 3.17: Less polluted route between a source (JU Gate No. 1) and a destination (JU Gate No. 4). (a) Less air pollution route (the right most route) (b) Less noise pollution route (two right most routes).

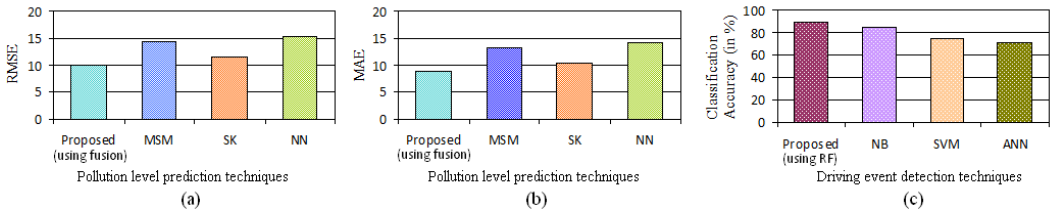


FIGURE 3.18: Performance comparison of the models. (a) RMSE values of the models for air pollution level prediction. (b) MAE values of the models for air pollution level prediction. (c) Classification accuracy of the models for driving events detection.

sensing via D&RSense is one of the most promising solutions here to cover the whole thing.

D&RSense has some limitations in road condition monitoring. However, the same road condition is also monitored by SenseDCity service, which basically relies on human input i.e., citizen as sensor. Based on human inputs, road segments are marked as road with bumps and potholes or damaged. In this work, we are not concentrating on the truthfulness of the collected data which is a different research aspect for any crowd/participatory sensing based application and out of scope of this work. Here these inputs are considered to be true as data are collected by our volunteers. Also, a point needs to be noted here that, the numbers of bad road conditions detected via SenseDCity are less as compared to D&RSense. It is because, manual inputs are necessary here to register any road condition related issue. However, with the above mentioned limitations of individual applications, we are getting some promising result which is shown in Fig. 3.19. Here, when the road conditions are tested with only participatory sensing based D&RSense, it gives an accuracy of 78.9%. However, on combining the results of the two systems (D&RSense and SenseDCity) as above, we get 87% accuracy, i.e., 8.1% increase in accuracy. This accuracy is calculated as follows: $Accuracy =$

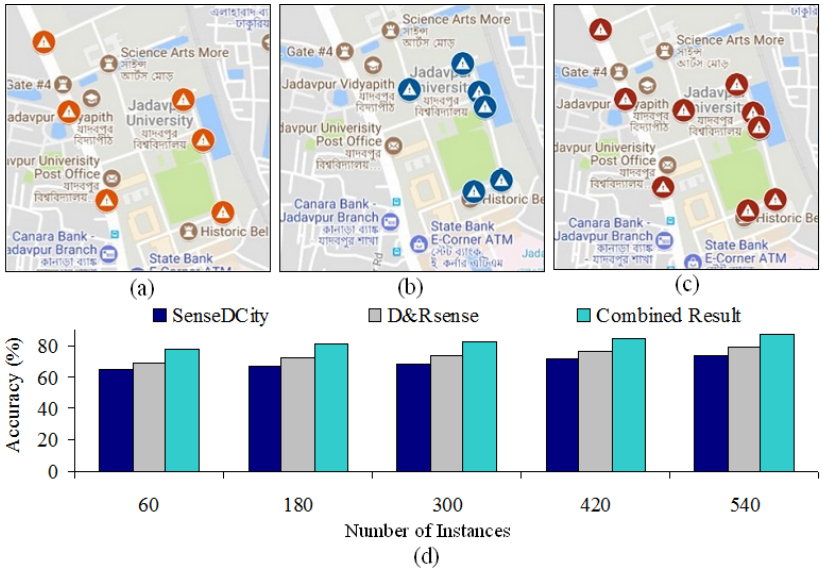
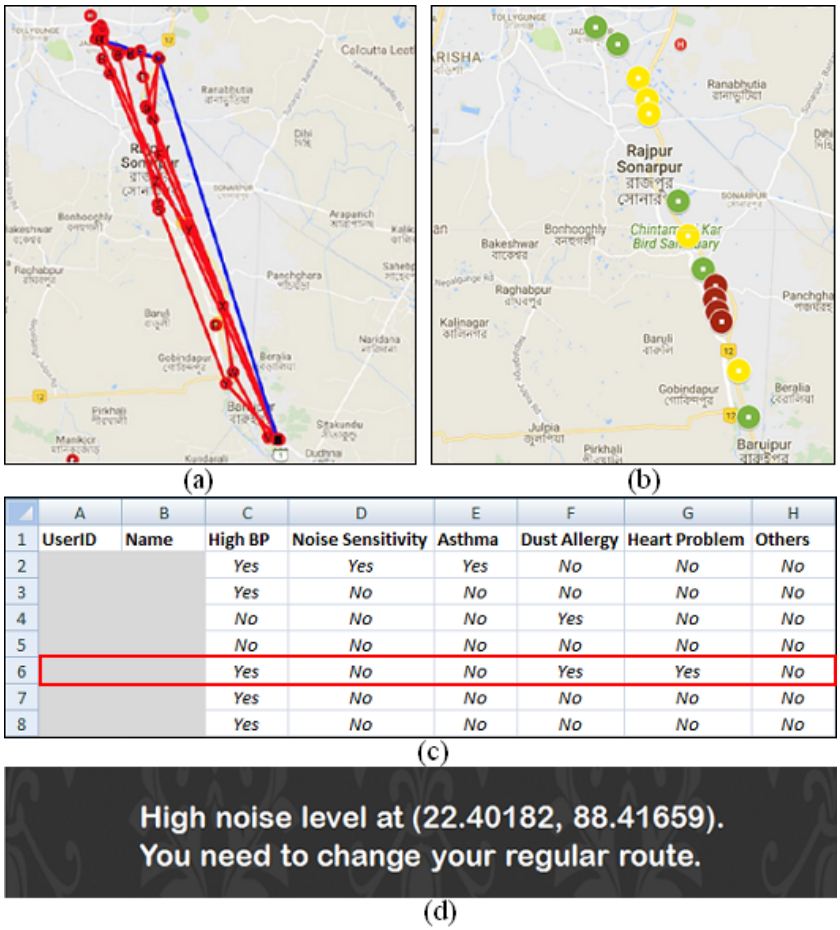


FIGURE 3.19: (a) Bad road conditions detected by SenseDCity (b) Bad road conditions detected by D&RSense (c) Combining the results of both SenseDCity and D&RSense in map (d) Performance comparison with varying dataset size.



	A	B	C	D	E	F	G	H
1	UserID	Name	High BP	Noise Sensitivity	Asthma	Dust Allergy	Heart Problem	Others
2			Yes	Yes	Yes	No	No	No
3			Yes	No	No	No	No	No
4			No	No	No	Yes	No	No
5			No	No	No	No	No	No
6			Yes	No	No	Yes	Yes	No
7			Yes	No	No	No	No	No
8			Yes	No	No	No	No	No

(c)

(d)

FIGURE 3.20: (a) Frequent travel path (blue line) of an user (b) Noise levels along the frequent travel path given by NoiseSense (c) User health database maintained by SenseDCity (d) Notification by SenseDCity

$\frac{RCCD}{TRC} \times 100\%$. Where, Road condition Correctly detected (RCCD) = (True Positive + True Negative) whereas, Total Road Conditions (TRC) = (True Positive + True Negative + False Positive + False Negative).

In scenario 2, the benefits of pollution monitoring applications (AirSense and NoiseSense) are combined with SenseDCity application. SenseDCity records data to keep track of the user's health information and its frequent travel pattern. The city's dynamic air and noise pollution map can be generated from AirSense and NoiseSense respectively. In JUSense, with the help of mutual interaction and information exchange, it can suggest a route to the user. Depending on the user's health, a suitable route which is not only the shortest but also good for his/her health is recommended.

In this experiment, participants with heart problems and asthma problems are identified from the SenseDCity application (as mentioned by the users). Besides, air and noise pollution maps are generated from AirSense and NoiseSense systems respectively. Here, we have recommended the route based on point of interest (POI). After merging these three results, JUSense can recommend an overall green route to the users based on their health (as mentioned by them).

In Fig. 3.20 (a), the blue line shows that one of the users frequent travel path in a week as detected by SenseDCity. Fig. 3.20 (b) shows noise levels along that path as maintained by NoiseSense. Fig. 3.20 (c) shows a snapshot of the health database maintained by SenseDCity application. The health profile of the above-mentioned user is shown within the red box which indicates that the user suffers from high blood pressure and heart problem. Due to its interaction with NoiseSense, SenseDCity successfully detects that the noise levels on that path has risen above tolerable limits and can be harmful to the user. Eventually, it sends an alert to the user as shown in Fig. 3.20 (d) so that the user will be able to change his travel route.

The proposed system JUSense is beneficial for users with health problems as well as for normal persons. Here, users will be notified about the pollution levels of that area and will also get a recommendation for an alternative route.

3.6 Summary

This chapter presents a resourceful and cost effective system JUSense based on a unified architecture that brings together different subsystems having their core in participatory sensing. It involves smartphones and IoT devices in order to improve the lives of citizens in the way of realizing a participatory based urban sensing system. JUSense considers four different applications (namely AirSense, NoiseSense, D&RSense and SenseDCity) to explore various dynamics of a city. Although, each of the subsystems (i.e. applications) can perform their task independently, this chapter mainly aims at bringing these independent applications under the same architecture concentrating on the benefits that would result from their mutual interactions. In JUSense we have proposed a multiview fusion model to explore the pollution dynamics of the city. On the other hand, a random forest (RF) classifier is built to detect various driving events. In order to evaluate the performance of JUSense several

large scale experiments are performed on participatory sensing data collected by participants over the course of 1 year (June 2018-June 2019) in Jadavpur area of city Kolkata. The fusion model yields better accuracy (RMSE $\approx$ 10 and MAE $\approx$ 8.8) compared to the baseline methods (namely, MSM, SK, NN). Whereas, random forest (RF) classifier achieves highest classification accuracy (89.2%). Additionally, case studies have been conducted in order to show interactions among applications through extensive experiment which further validate our proposed approach. JUSense is implemented in cloud using Amazon Web Service and Heroku cloud platforms. The participatory sensing approach may produce a variety of big data for different dynamics of a city. Here, the utilization of the cloud infrastructure makes it easier to store, track, analyze and handle big data.

In this work, an intriguing challenge arises in utilizing smartphone-based participatory sensing for urban dynamics. Sharing smartphones with guest users is common (e.g., family members, colleagues, friends), necessitating participatory sensing platforms to accurately identify the genuine data provider. This identification is crucial as raw sensing data may vary in accuracy based on user expertise, reliability, and trustworthiness. To address this, an efficient and inconspicuous privacy-protected authentication system is essential for identifying the current user (i.e., participatory data provider) from their smartphone sensor data. Hence, the next chapter of the thesis provides a solution for the privacy protected user authentication / identification for smartphone-based participatory sensing

Publication(s) related to this Chapter:

- *A.I. Middya, S. Roy, J. Dutta, and R. Das, 2020. Jusense: a unified framework for participatory-based urban sensing system. Mobile Networks and Applications, 25, pp.1249-1274. (DOI: https://doi.org/10.1007/s11036-020-01539-x)*

Chapter 4

Privacy Aware User Identification for Smartphone-based Sensing

4.1 Introduction

In smartphone-based crowd/participatory sensing systems, it is necessary to identify the actual sensor data provider. In this context, this chapter attempts to recognize the users' identity based on their gait patterns (i.e., unique walking patterns). In smartphone-based crowdsensing [3] and participatory sensing [18] systems, participants share surrounding dynamics or personal information with others. More specifically, in these paradigms of sensing participants use their smartphones to collect high granular data that can be used to explore different dynamics of a city (e.g., road condition monitoring [148], pollution monitoring [**?**, 22, 149], travel pattern monitoring [11], etc.). Now, as sharing smartphones with guest users (e.g., members of the family, colleagues, friends, etc.) is a common scenario, it is necessary for the crowdsourcing platforms to identify the actual data provider. It is because the raw participatory/crowd sensing data may suffer from inaccuracies due to the variety of expertise, reliability, and trust value of users. In this context, smartphone user identification is of utmost importance for personalized services and sensing data authentication in participatory sensing systems.

There exist explicit knowledge-based user identification mechanisms [150], such as Personal Identification Numbers (PINs), passwords, or answering specific security questions. Also, there is some biometric-based authentication [151] such as fingerprint, face recognition, voice recognition, signature, iris print, etc. However, the participatory sensing based applications sometimes running in the background for data collection from smartphone equipped sensors, in such context, it may be annoying for the users to authenticate themselves at frequent time intervals. Moreover, several issues are associated with each of the above-mentioned authentication mechanisms. For instance, password or PIN based approaches require a lot of memorizing and remembering, or they can be lost or stolen away. According to a survey [152], an average of 21 passwords (a few of those up to 70) must be remembered by a high-end user, 49 percent of users enter or store their passwords in files and 67 percent of users do not change passwords. The fingerprint recognition requires direct physical contact with the sensor and the finger, and the slightest damage or impurity on the surface can result in malfunction. On the other hand iris or face recognition requires a sufficient

amount of lighting, voice recognition requires an absence of any background disturbances, etc. In addition to these, over prolonged periods of usage time, the sensors tend to lose their sensitivity, and the data such obtained from them are not similar to those as were obtained when the sensors or the smartphone were new. As a result of this, after a certain point of time, the biometric systems tend to lose their efficiency, unless the sensors are changed. At the same time, creating a universal biometric system, which can work on as many devices as desirable is a very tedious job at a civilian level, the sole reason being that different devices and smartphones use different models and types of sensors and their data are incompatible with the algorithm of one another.

4.1.1 Objective

Under the circumstances mentioned above, it would be better for the crowdsourcing platform to efficiently identify the current user (i.e. sensor data provider) from their smartphone sensor data, as well as automatically authenticate them. In this context, the chapter mainly aims to explore the potential of Convolutional Neural Network (CNN)-based deep learning model for analyzing accelerometer data to uniquely identify users without requiring explicit interaction or compromising their privacy. More specifically, the objective of the work presented in this chapter is to develop a privacy-aware user identification system for smartphone-based participatory sensing using gait recognition. Gait refers to the characteristics and mannerisms in which a person walks, and is fairly unique for every individual [153]. Gait recognition is among the latest emerging biometrics and can alleviate several issues. It is a non-contact biometric that does not require direct interaction with the subject. This work deals only with inbuilt sensors within smartphones in the form of MEMS (Micro-Electro-Mechanical Systems) to capture the accelerometer values of a person when they walk. The accelerometer data are processed to establish the persons' unique gait so that the data obtained from various data sources could be distinguished.

4.1.2 Contribution

The major contributions of this chapter are summarized as follows:

- A CNN-based deep learning model is proposed for privacy aware user identification with smartphone equipped accelerometer sensor data in crowd/participatory sensing systems.
- Extensive experiments are performed to evaluate the performance of the proposed CNN model on a real-world benchmark dataset.
- To validate the effectiveness of the proposed model, it is compared with five baseline methods namely Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbors (KNN).

TABLE 4.1: A summary of existing literature on user identification based on the sensor data.

Classifier	Ref & Year	Contribution	Performance			Number of data samples used	Number of sensors used	Number of users classified
			Accuracy	Precision	Recall			
CNN	Benegui et al. [105], 2020	User Identification based on Motion Sensor data of Tap on screen motion represented as Images	90.75	–	–	150	2	50
RF	Angrisano et al. [106], 2020	Walker Identification using Ensemble Learning	93.8	92.0	93.0	32	3	18
CNN	Neverova et al. [107], 2016	Walker Identification by Dense Convolutional Clockwork RNNs	69.41	–	–	500	3	587
NN	Buriro et al. [108], 2016	User Identification based on sensory data of hand and finger movement on smartphone screen	–	–	79.0	175	3	30
RF	Buriro et al. [108], 2016	User Identification based on sensory data of hand and finger movement on smartphone screen	–	–	71.0	175	3	30
NB	Shi et al. [109], 2011	Senguard: Passive user identification on smartphones using multiple sensors	96.4	–	–	512	1	4
NN	Kwapisz et al. [154], 2010	Walker Identification using Neural Networks and J48 classifier	74.7	–	–	200	1	36
J48	Kwapisz et al. [154], 2010	Walker Identification using Neural Networks and J48 classifier	79.3	–	–	200	1	36
KNN	Yam et al. [155], 2004	Walker Identification by gait signature extraction from Image Analysis	85.00	–	–	–	1	20

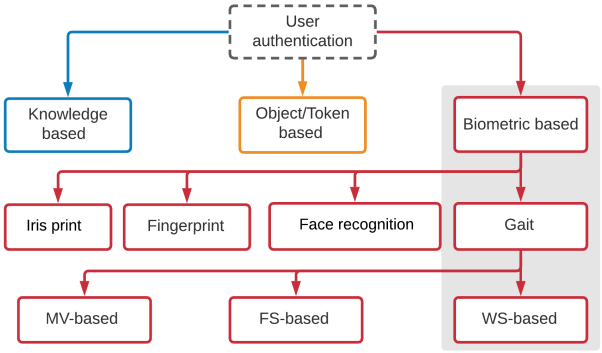


FIGURE 4.1: A taxonomy of the user identification / authentication methods. The shaded portion indicates the category in which the proposed work best fits.

4.2 Existing Works

This section provides a background study explaining existing literature of user identity recognition. First, a taxonomy of the existing approaches is provided. After that, the details of various studies on sensor based user identification are discussed.

As shown in Fig. 4.1, the existing user authentication methods could be categorized into three groups namely knowledge based [150], token/object based [156], and biometric based [151, 157–159]. Nowadays, gait-based biometric studies [153, 160] are new trends that are increasingly used for the recognition of user identity due to their outstanding performance. Biometric gait-based research works can be divided into three groups depending upon the placement and the type of sensors used to determine the individual's gait, namely machine vision based [161], wearable sensor based [162, 163], and floor sensor based [164]. This study of user identification using smartphone equipped accelerometer sensor data best fit into the second category (i.e., wearable sensor-based research).

Recent gait research shows promising results by the use of a person's distinctive movement (walking or running) to authenticate or to identify one's identity. Angrisano et al. [106] proposed a random forest based approach that utilizes a combination of sensors (accelerometer, gyroscope and orientation) for user identification. Their model architecture was composed of two different types of classifies - C2 to classify the device pose (the place where the device was kept while walking) and 4 C1 classifiers for classifying users for each pose. The two types of classifiers formed the hierarchical classifier C3 which was used for real binary classification to identify a user. In [154], Kwapisz et al. used the accelerometer data and employed state-of-the-art models such as WEKA's J48 and artificial neural network for classification. Benegui et al. [105] have collected accelerometer and gyroscope data while the user tapped on the screen. They transformed their data into 2D grey scale images and then used that to train a CNN model. In [155], Yam et al. tracked the movement of the thigh and lower leg while running and walking and extracted gait signatures. They performed cluster separation of the features and used K-Nearest Neighbour for classification purposes. There is another interesting work [107] in which Neverova et al. proposed an architecture called Conv-DCWRNN. They used Conv-DCWRNN to perform a large scale study based on the accelerometer and gyroscope sensor data. Buriro et. al [108] proposed a new multi modal

behavioural biometric that considers the features collected during the swiping movement of the user while trying to receive a call. They particularly used the screen swipe, the arm movement while bringing the smartphone closer to one's ears and voice recognition while extracting features. In [109], Shi et al described a user identification framework called SenGuard which enabled continuous and implicit user identification service in smartphones. A summary of existing literature on sensor based user identification is provided in Table 4.1.

4.3 Problem statement

The aim in this work is to perform privacy aware user recognition using deep learning for smartphone based participatory sensing applications. The problem is formulated in such a manner that the gait-based identity recognition can be performed using smartphone equipped accelerometer sensor data. For this purpose, the accelerometer sensor data of different individuals (while walking) are processed and different features are extracted from them.

Let us consider, $\mathscr{U} = \{u_1, u_2, u_3, \ldots, u_n\}$ is a set of users who have shared their smartphone's accelerometer sensor data. Suppose, $\mathcal{D}_i = \{d_{i1}, d_{i2}, d_{i3}, d_{ik}\}$ denotes the data samples of user $u_i \in \mathscr{U}$. Here, each sample $d_{ij} \in \mathcal{D}_i$ has the following form $d_{ij} = (x_{ij}, y_{ij}, z_{ij}, t_{ij})$, where x_{ij}, y_{ij}, and z_{ij}, represent the accelerometer values of x, y, and z component respectively; t_{ij} is the timestamp when the sample is taken. Note that $\forall_{1<j<k}$, $|t_{ij} - t_{i(j-1)}| = |t_{i(j+1)} - t_{ij}|$. Given a set of such data $\mathscr{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ from the users $u_i \in \mathscr{U}$, our objective is to build a deep convolutional neural network (CNN) based classification model that can predict the users based on the smartphones' accelerometer data.

Specifically, the extracted features are passed through the proposed CNN-based deep learning model. During the training process of the CNN model, the input features pass through the convolutional layers of the CNN and is flattened which ultimately reaches the final dense layers where they are mapped with the final output neurons corresponding to the different individuals providing the accelerometer data/signals. During the testing process, the fitted model will be used to predict the users from previously unseen accelerometer data/signals and analyse its performance by determining the different performance measures like accuracy, loss, etc.

4.4 Proposed methodology

Fig. 6.1 sets out a framework for the proposed approach that outlines the major steps involved in learning the identity of users from smartphone accelerometer data. The framework consists of following major parts: *data collection*, *pre-processing*, *windowing and feature extraction*, and *performance evaluation*.

Sensor data

There are various MEMS sensors available on different smartphones. This work focuses on the use of 3-axis accelerometer data from users' smartphones for predictive model building. An accelerometer measures the acceleration forces acting on an object, in order to monitor the

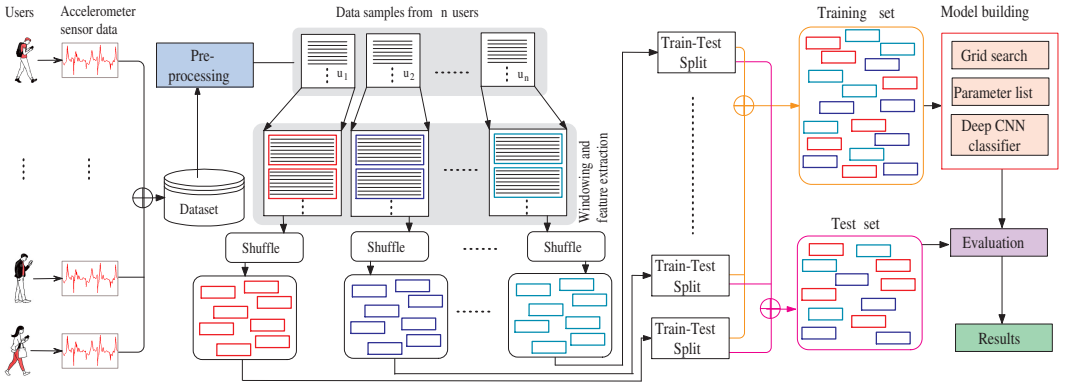


FIGURE 4.2: A framework describing overall process of the proposed approach of user identification. The framework consists of major steps such as accelerometer data collection, pre-processing, windowing and feature extraction, train-test splits, model building, and evaluation

object's movement. This accelerometer sensor can be used to record time series accelerometer data with a user defined sampling frequency. The accelerometer contains two arms, one static arm and one movable arm. A capacitance is formed between the two arms which then changes when the sensor is subjected to acceleration. An accelerometer measures proper acceleration, which is the acceleration it experiences relative to freefall and is the acceleration felt by people and objects. In literature [153], it has been seen that the exact way of how a person walks, i.e Gait varies from individual to individual. During the process of walking the individual goes through a series of repetitive movements which when measured through an accelerometer produces a periodic signal. It is observed that, this signal though periodic is different for different individuals. Due to this reason, the accelerometer data is utilized in the user identification task to extract features. Specifically, the accelerometer data obtained from smartphone equipped accelerometer sensor that can reflect the essence of Gait in a way such that features can be extracted out using the prevalent signal processing techniques. In this context, this work made an attempt to find out whether the accelerometer data signal produced by different users can be classified using a deep learning neural networks and the signal can be treated as a biometric. In this study, an accelerometer biometric competition dataset is used that is provided by Kaggle [165]. The dataset contained over 60 million data samples of 387 users.

Data preparation

In order to prepare the dataset for further analysis, first, the users having an insufficient number of data samples are removed from the dataset. It is because the deep learning models perform well for a large number of data samples [166]. More specifically, in this study, if the number of data samples (n_s) for a user is less than 15000, the data for that user will not be taken for model building purposes. Now, since the data produced by the signal is a time series signal and the whole signal cannot be fed into CNN model, small time series windows are created. In Fig. 6.1, the small rectangular boxes (shown in red, blue, etc.) denote the window of values. The windows were essentially a two dimensional array consisting of the three columns (i.e., X, Y, Z) and number of rows equal to the window size (W_s). Note that

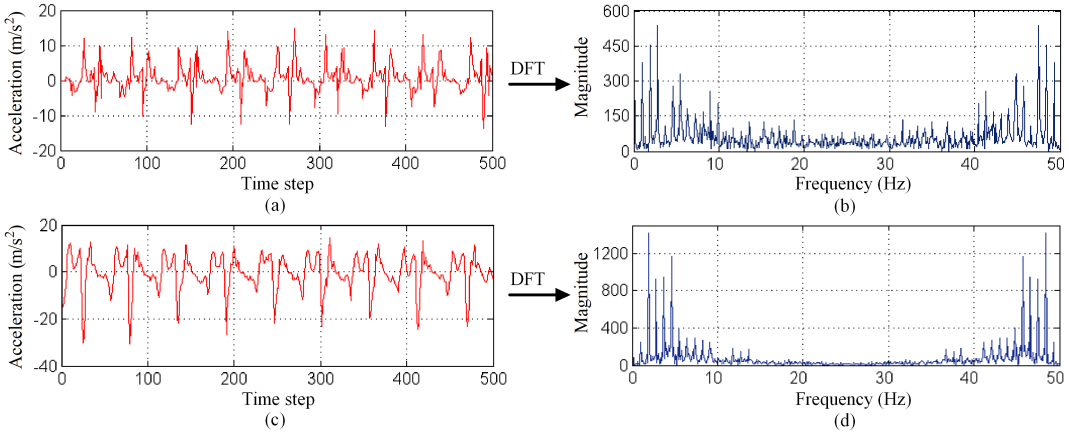


FIGURE 4.3: The magnitude spectrum from the Fourier transform of two typical accelerometer signals from two users. (a) acceleration vs time plot of the first user (b) magnitude spectrum of accelerometer signal of the first user (c) acceleration vs time plot of the second user (d) magnitude spectrum of accelerometer signal of the second user.

TABLE 4.2: Features for user identity detection

Feature	Description
$\mathcal{A}_x$	Acceleration on X-axis
$\mathcal{A}_y$	Acceleration on Y-axis
$\mathcal{A}_z$	Acceleration on Z-axis
$\mathcal{S}_x$	The standard deviation of acceleration on X-axis
$\mathcal{S}_y$	The standard deviation of acceleration on Y-axis
$\mathcal{S}_z$	The standard deviation of acceleration on Z-axis
$\overline{\mathcal{A}_x}$	The mean acceleration on X-axis
$\overline{\mathcal{A}_y}$	The mean acceleration on Y-axis
$\overline{\mathcal{A}_z}$	The mean acceleration on Z-axis
$\mathcal{M}_x$	The magnitude spectrum of acceleration on X-axis
$\mathcal{M}_y$	The magnitude spectrum of acceleration on Y-axis
$\mathcal{M}_z$	The magnitude spectrum of acceleration on Z-axis

the window size could be varied to find out the best W_s for achieving high accuracy. In this context, Angrisano et.al [106] highlighted that the accuracy increment was quite negligible for window sizes of more than 4.5 seconds. Note that they used a sampling frequency of 100Hz. In our case, the accelerometer biometric competition dataset [165] provides us with a sampling frequency of 5Hz. The best performance is achieved for 225 samples per window (i.e. W_s = 225). Here, due to a relatively low sampling frequency (i.e. 5Hz), it corresponds to 45 seconds of time series data. It is quite obvious that increasing window size (number of data samples) will increase the predictive accuracy. Since the pattern of walking or gait will be more prominent with greater window sizes but whilst increasing window size does increase the accuracy it increases the load on the processing device and the complexity of the model.

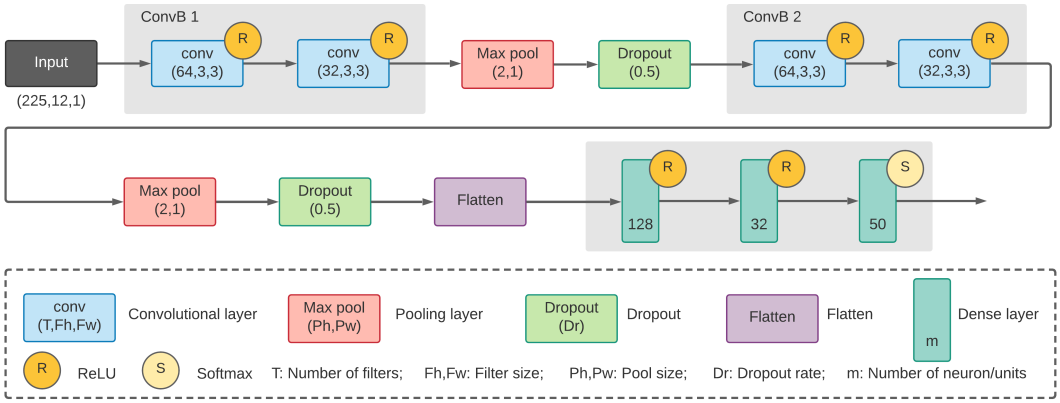


FIGURE 4.4: Architecture of the proposed CNN model. It consists of four convolutional layers, two max-pooling layers, two dropout layers, two fully connected layers, and one output layer.

Feature extraction

The statistical and DFT (Discrete Fourier Transform) based features extracted from the accelerometer data are discussed in this subsection. Statistical features such as mean and standard deviation of each time series window are extracted. The statistical mean and standard deviation of a window of values $a_1, a_2, a_3, \ldots, a_n$ is computed in Eq. 4.1 and Eq. 4.2 respectively.

$$\bar{a} = \frac{1}{W_s} \sum_{n=1}^{W_s} a_n \tag{4.1}$$

$$s = \sqrt{\frac{1}{W_s} \sum_{n=1}^{W_s} (a_n - \bar{a})^2} \tag{4.2}$$

where W_s is the size of the window. Here, $\bar{a}$ denotes the mean, and s is the standard deviation that is a measure of dispersion of the set of values from its mean.

It will now be discussed that the frequency spectrum in the experiment can proved to be a helpful feature. In Fig. 4.3, the magnitude spectrum (Fig. 4.3 (b) and Fig. 4.3 (d)) from the Fourier transform of two typical accelerometer signals (Fig. 4.3 (a) and Fig. 4.3 (c)) from two users is shown. From Fig. 4.3 (a) and Fig. 4.3 (c), it can be seen that there is a typical pattern present in the accelerometer signal. The process of walking involves lot of muscular movements. These movements occur periodically which ultimately adds up to create the walking movement. Here, the accelerometer signal obtained can be though of to be composed of those periodical oscillations. So, if a Fourier transform is performed on the accelerometer signal then the frequencies present in the signal can be found out from the magnitude spectrum. Here, the Fourier magnitude spectrum is used as a feature in the deep-CNN model. Note that the representation of the signal in terms of its frequency components in the frequency domain (i.e. the signal spectrum) could be developed using N-point DFT

(see Eq. 4.3).

$$X_n = \sum_{k=0}^{N-1} x_k e^{-jn(\frac{2\pi k}{N})}, \text{ for } \quad n = 0,1,2,...,N-1 \tag{4.3}$$

where x_k is the input sequence for $k = 0,1,2...N-1$ and X_n is the output magnitude DFT sequence. The N-point DFT will produce complex numbers (magnitude and phase) for each probable frequency present in the accelerometer signal. In summary, a total of 4 features (i.e., mean, standard deviation, Fourier magnitude spectrum, and acceleration) are extracted from the accelerometer signal of each of the 3 axis. Hence, a total of (4×3) = 12 features are extracted as listed in Table 4.2. Note that a train-test split is performed for each user's data after the windowing and feature extraction stage (see Fig. 6.1).

Architecture of Deep CNN

A deep learning model based on a Convolutional Neural Network (CNN) is proposed in this work for user identification. The proposed CNN consists of four convolutional layers, two max pooling layers, two dropout layers, and two fully connected layers. The architecture of the model is presented in Fig. 7.3. An input $\mathbf{X} \in \mathbb{R}^{225\times 12}$ containing the values of a window is fed to the convolutional neural network. It has two convolution blocks (convB1 and convB2) each having two convolution layers. The first, second, third and the fourth Conv2D Layers have 64 ,32, 64 and 32 filters respectively. All the convolution layers have a kernel size of (3,3). Note that, if the input matrix is I and K is a kernel of size $m \times n$, then after convolution the revised estimate $E_{i,j}$ at the matrix location $I_{i,j}$ will be computed as follows:

$$E_{i,j} = (I \times K)_{i,j} = \sum_{p=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{q=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-p,j-q} K_{\frac{m}{2}+p,\frac{n}{2}+q} \tag{4.4}$$

where p and q represent the relative indices of K, and can vary between $-m/2$ to $+m/2$ and $-n/2$ to $+n/2$ respectively. In each of the convolutional layers, rectified linear units (ReLUs) are used as an activation function. After each convolution block, max pooling is applied with filter size of 2×1 and a stride of 1. The first fully connected dense layer consists of 128 neurons and the second fully connected dense layer consists of 32 neurons. Finally, the proposed CNN ends with the Softmax [167] function which is mathematically defined in Eq. 4.5.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{t=1}^{Q} e^{z_t}} \tag{4.5}$$

where $\mathbf{z} = (z_1, z_2, z_3, \ldots, z_Q) \in \mathbb{R}^Q$ is the input vector; e^{z_i} and e^{z_t} are the exponential function for the input and output vectors respectively; Q denotes the total number of classes. In this study, categorical cross-entropy is used as a loss function which is defined as follows (Eq.

4.6)

$$L = -\sum_{k}^{\mathcal{U}} y_i \log(s_k) \tag{4.6}$$

where $\mathcal{U}$ is the set of classes, y_k and s_k denote the ground-truth and convolutional neural network score for each class k in $\mathcal{U}$. Now, the Adam optimizer [168] is used to minimise the output of the loss function. As shown in Fig. 6.1, grid search [169] technique is employed to find optimal hyper-parameter setting in proposed CNN.

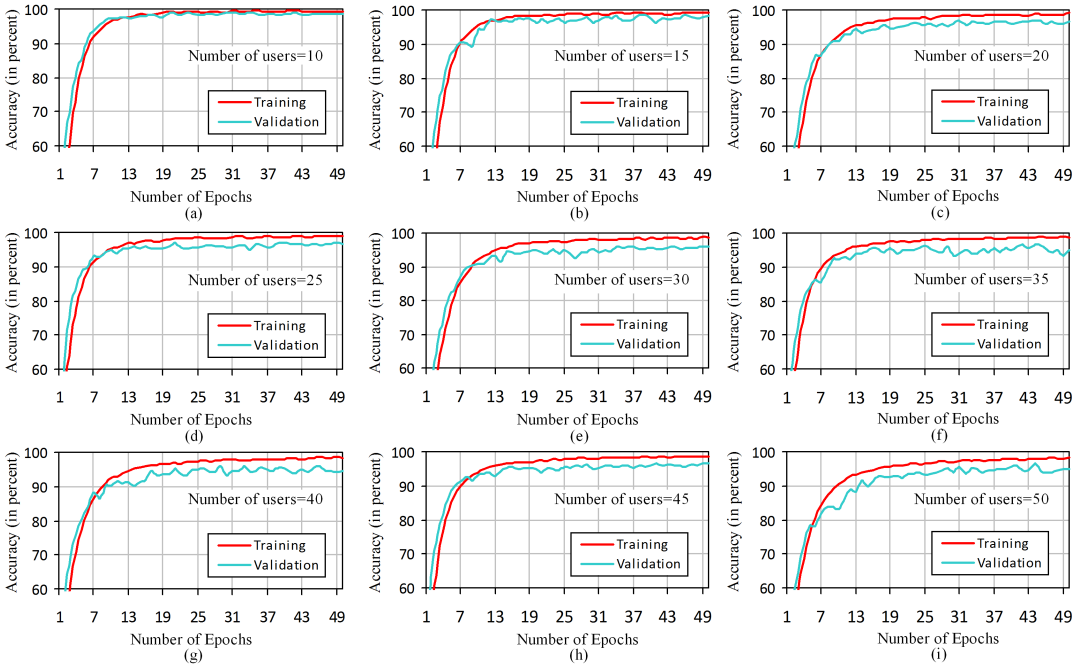


FIGURE 4.5: The plots of training and validation accuracy versus number of epochs for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users.

4.5 Evaluation

In this section the effectiveness of the proposed CNN-based user identity detection model is assessed through rigorous experiments. The performance of the model is compared with various baseline methods in terms of four evaluation metrics namely accuracy, precision, recall, and F1 score. Moreover, the results are also evaluated through training and validation loss curves, accuracy curves, execution times, etc. For each user, the training set contains 85% data (70% for actual training and 15% for validation) and the test set contains remaining 15% data.

4.5.1 Experimental setup

In this study, all the experiments are carried out on a 64-bit Windows PC with 1.8GHz Intel(R) Core(TM) i7 CPU and 16GB of RAM. The proposed model and the baselines are

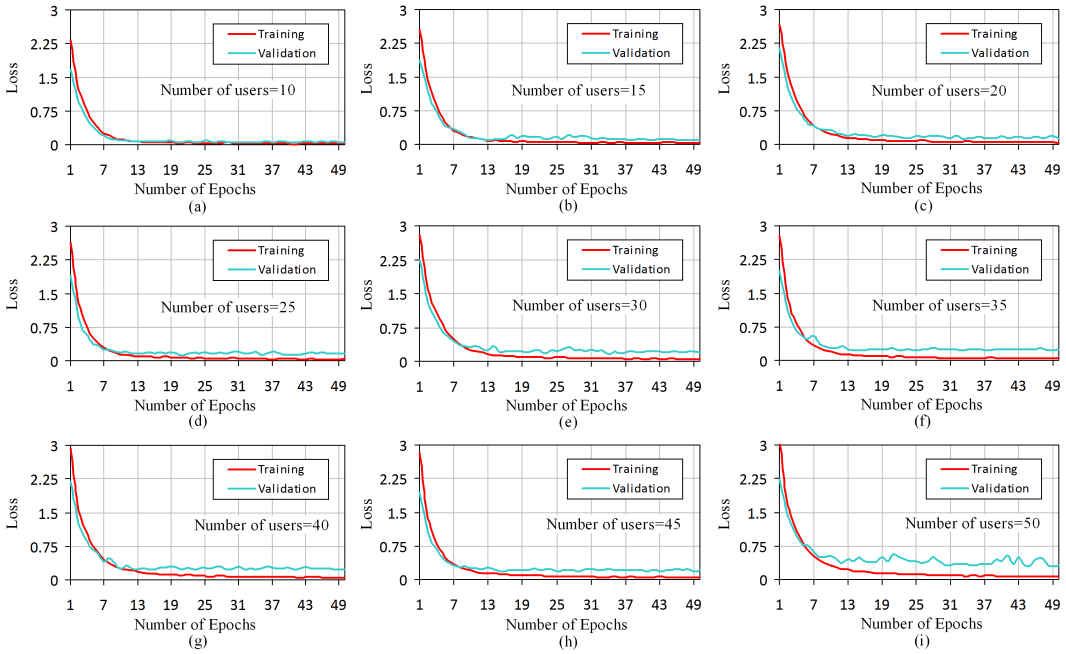


FIGURE 4.6: The plots of training and validation loss versus number of epochs for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users.

implemented in Python 3.7 [170]. The following Python libraries are utilized during model development: (i) Keras (version 2.2.4), Tensorflow (version 1.13.1), scikit learn (version 0.21.2), numpy (1.13.3), pandas (version 0.24.2), matplotlib (version 3.1.0), scipy (version 0.19.1), etc.

4.5.2 Baselines and Evaluation metrics

In order to determine the effectiveness of the proposed CNN based model, it is compared with five baseline methods namely Support Vector Machine (SVM) [171], Decision Tree (DT) [172], Random Forest (RF) [173], Logistic Regression (LR) [174] and K-Nearest Neighbors (KNN) [175]. The models are mainly evaluated based on four performance metrics namely accuracy, precision, recall and F1 score. Accuracy denotes the number of true predictions made by the classifier over all the predictions. It is mathematically defined in Eq. 4.7. In this equation, tp_c, tn_c, fp_c, and fn_c denote the number of true positives, true negatives, false positives, and false negatives for class $c \in \mathcal{U}$. On the other hand, precision, recall, and F1-score are defined in Eq. 4.8, Eq. 4.9, and Eq. 4.10 respectively. Note that the precision (Eq. 4.8) and recall (Eq. 4.9) are computed by macro-averaging. Here, harmonic mean of the macro-averaged recall and precision is the F1-score (Eq. 4.10).

$$Accuracy = \frac{\sum_{c}^{\mathcal{U}} \frac{tp_c + tn_c}{tp_c + tn_c + fp_c + fn_c}}{|\mathcal{U}|} \tag{4.7}$$

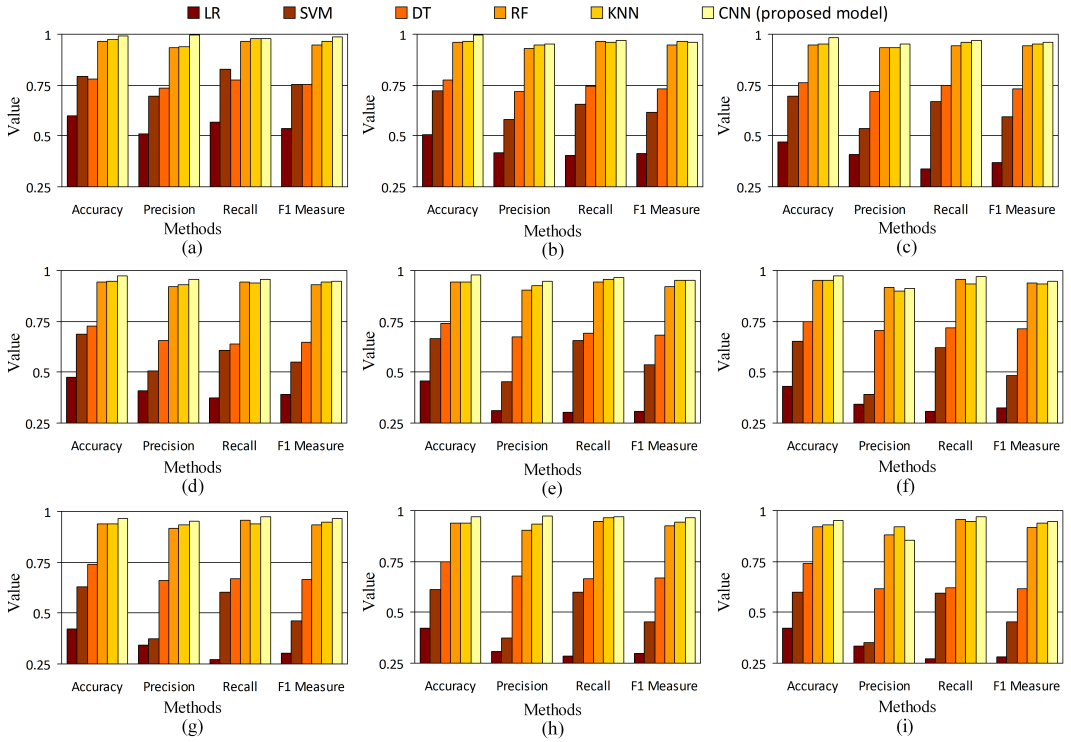


FIGURE 4.7: Performance comparison of the proposed model (CNN) with the baselines (LR, SVM, DT, RF, KNN) in terms of four metrics (accuracy, precision, recall, and F1 measure) for (a) 10 users (b) 15 users (c) 20 users (d) 25 users (e) 30 users (f) 35 users (g) 40 users (h) 45 users and (i) 50 users.

$$Precision = \frac{\sum_{c}^{\mathcal{U}} \frac{tp_c}{tp_c + fp_c}}{|\mathcal{U}|} \tag{4.8}$$

$$Recall = \frac{\sum_{c}^{\mathcal{U}} \frac{tp_c}{tp_c + fn_c}}{|\mathcal{U}|} \tag{4.9}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.10}$$

4.5.3 Results and discussion

In order to evaluate the effectiveness of the proposed CNN the accuracy and loss curves are generated. Fig. 4.5 (a)- Fig. 4.5 (i) denote the training vs validation accuracies over 50 epochs for different numbers of users. The CNN based approach achieves very high predictive accuracy ($accuracy \geq 97.5\%$) for less number of users (i.e. $\#users \leq 25$). However, the predictive accuracy slightly decreases ($accuracy \in [96\%, 97.5\%]$) if the number of users increases ($\#users \in [30, 50]$). On the other hand, Fig. 4.6(a) - Fig. 4.6(i) represent the training vs validation loss for different number of users. It could be used to describe that there is no overfitting of the proposed model. The figures (Fig. 4.6(a) - Fig. 4.6(i)) show that both the

training loss and the validation loss curves decrease to a point of stability, and have a small generalization gap. Further, the training and validation loss curves almost follow each other, which strongly suggests that there has been no overfitting.

The performance of the CNN based approach is also compared with five baselines (SVM, DT, RF, LR, and KNN). A summary of various models' performance in terms of *accuracy, precision, recall,* and *F1-score* are presented in Fig. 4.7 for different number of users (i.e. *#users* = 10, 15, 20, ..., 50). It is to be noted that the values of the evaluation metrics are computed based on separate test set. The results in Fig. 4.7(a) - Fig. 4.7(i) indicate that the CNN based approach achieves better accuracy compared to other methods. Further, the proposed approach also clearly dominates over the baseline methods in terms of other performance metrics (precision, recall, and F1-score). Among the baselines, however, the techniques such as RF and KNN perform very well. In this regard the performance of the three best performing methods (CNN, RF, and KNN), averaged over the different number of users (i.e. *#users* = 10, 15, 20, ..., 50), are presented in Table 4.3. The CNN model achieves an accuracy of 0.98.82% in the separate test set. The Table 4.3 also indicates that the uncertainties in the prediction accuracy (expressed in terms of standard deviation) is relatively less for different number of users.

Now, Table 4.4 presents the training time (in seconds) for various models for different number of users. The training time for CNN is the average per epoch time. It is observed that the training time is relatively high for CNN and SVM. Note that these two models (i.e., CNN and SVM) take more time compared to others models, because they need to estimate a large number of parameters.

TABLE 4.3: Model accuracy

Models	Performance of the models (mean $\pm$ sd)			
	Accuracy	Precision	Recall	F1-score
RF	0.9551 ± 0.003	0.914 ± 0.017	0.954 ± 0.009	0.933 ± 0.011
KNN	0.9614 ± 0.005	0.928 ± 0.014	0.956 ± 0.008	0.941 ± 0.009
CNN (proposed)	$\mathbf{0.9882 \pm 0.004}$	$\mathbf{0.942 \pm 0.040}$	$\mathbf{0.978 \pm 0.027}$	$\mathbf{0.959 \pm 0.030}$

TABLE 4.4: Model training time (in seconds)

Users	LR	SVM	DT	RF	KNN	CNN
10	18.86	413.81	52.72	59.85	4.59	360.02
15	27.85	1125.63	77.18	92.33	5.93	540.14
20	43.11	1753.47	149.09	173.91	9.02	720.57
25	56.22	2754.26	201.01	230.22	11.01	732.55
30	87.94	3882.74	294.66	334.17	16.79	1080.38
35	92.16	5127.17	324.41	374.74	15.93	1200.87
40	134.68	7168.51	486.73	582.08	26.98	1380.73
45	137.92	8793.29	497.99	576.71	21.67	1620.27
50	195.12	10068.93	727.02	848.42	31.84	1800.45

4.6 Summary

In this chapter, an approach using Convolution Neural Network (CNN) is proposed on the data produced by the smartphone equipped accelerometer sensor to recognize a particular user's identity during participatory sensing based data collection. This chapter also attempts to find out if the use of a method of signal processing such as Fourier transform is useful to produce a model that is more robust and accurate than the state-of-the-arts. Based on the findings, it is concluded that the Fourier transform is a simple yet a very powerful tool in the feature extraction process which has improved the model accuracy to a large extent. The performance of the model is evaluated on real world benchmark dataset. In order to investigate the effectiveness of the proposed approach, it is compared with five baseline methods namely SVM, DT, RF, LR, and KNN. The proposed CNN achieves better results (accuracy = 98.8%, precision = 0.94, recall = 0.97, and F1-score =0.95) as compared to the baseline methods.

In this chapter, a critical challenge is identified: ensuring adequate user participation in participatory sensing systems to gather sufficient data. Smartphone users, engaged in participatory sensing tasks, not only face costs related to data sensing and uploading but also encounter other potential privacy issues related to disclosing sensitive information, such as GPS location. The dual concerns significantly demotivate users from actively engaging in sensing tasks. To address this, the next chapter of the thesis proposes an incentive mechanism designed to compensate users and encourage their active participation in sensing tasks.

Publication(s) related to this Chapter:

- *A.I. Middya, S. Roy, S. Mandal and R. Talukdar, 2021. Privacy protected user identification using deep learning for smartphone-based participatory sensing applications. Neural Computing and Applications, 33, pp.17303-17313.*
  (DOI: https://doi.org/10.1007/s00521-021-06319-6)

Chapter 5

Double Auction based Incentive Mechanism

5.1 Introduction

Smart handheld devices (e.g. smartphone, tablet, etc.) are rapidly becoming an integral part of people's daily life. Importantly, today's smart handheld devices are equipped with useful sensors like accelerometer, GPS, gyroscope, microphone, camera, etc. These embedded sensors open up the possibility of monitoring a diverse set of users' activities and ambient environment. Now, participatory sensing (also known as crowdsensing) [17, 176] is a new paradigm of sensing where participants use their smart handheld devices to collect high granular data that can be used to explore different dynamics of a city (e.g. road condition monitoring, pollution monitoring, etc.). The active participation of a large number of smartphone users is extremely important for participatory sensing based systems to acquire sufficient data. But, in participatory sensing tasks, smartphone users not only incur costs due to sensing and uploading data but also suffer from potential privacy issues due to revealing sensitive information (e.g. GPS location of the users). Here, both the privacy and the cost issues greatly demotivate the users to actively participate in the sensing tasks. In this context, a satisfying reward/ incentive will compensate the users and attract them to join in the sensing tasks. Hence, in a participatory sensing based system a good incentive mechanism will play an important role in sensing and overall success of the system.

The main focus of this chapter is to come up with an incentive mechanism for the sensing tasks that are related to different dynamics of a city (e.g. road condition monitoring [177, 178], mobility pattern [179], air pollution monitoring [180, 181], noise pollution monitoring [22, 182], etc.). From this perspective, a city can be viewed as different functional regions such as educational areas, residential areas, business areas, areas with major road segments, etc. These regions can be known as Region Of Interests (ROIs). Now, there are different entities such as local authorities, government agencies, public interest groups, academic or research oriented organizations, individuals who may be interested to explore different dynamics of a city at different ROIs. In order to explore a particular city dynamics at the ROIs, the entities provide sensing tasks related to that. These sensing tasks are then allocated to a set of smartphone users and after task completion incentives are provided to them. In this chapter, the entities who provide the tasks are called Task Providers (TPs) and

the smartphone users who perform the tasks are called Task Executors (TEs). It is important to note that, a particular TP may have sensing tasks at some specific ROIs. For example, in case of air pollution dynamics, a hospital authority (the task provider) may be interested in sensing air pollution levels (the task) only in those areas (the ROIs) where its branches are located. Similarly, a transport organization (another TP) may be interested in sensing road conditions in some areas that are selected for their studies.

Many researchers consider the potential of smartphone equipped sensors to develop various participatory sensing based systems, such as traffic monitoring systems [183, 184], road condition monitoring systems [185, 186], pollution monitoring systems [18, 187], etc. Table 5.1 shows a more detail list of participatory sensing based systems/ applications and the smartphone sensors that they use. These smartphone based participatory sensing systems assume that the participants are always willing to participate in the sensing tasks. But, as discussed earlier, this is not a realistic assumption due to the privacy and cost issues of the participants. Although, there are some research works [58, 61, 63–65, 188] that consider various incentive mechanisms for motivating the smartphone users to join in the participatory sensing tasks. However, these incentive models suffer from several limitations.

- They do not satisfy all the important and critical properties namely *truthfulness, computational efficiency, budget balance, and individual rationality* of incentive based sensing task allocation mechanism. Truthfulness indicates that revealing the true valuation of the tasks will be the dominant strategy for TPs/TEs to gain an optimal utility. Individual rationality ensures that no TPs/TEs will have negative utility if they reveal the true private valuation. The computational efficiency indicates that the auction based incentive mechanism will take polynomial time. Budget balance ensures that the auctioneer (a trust worthy third party who govern the auction based incentive mechanism) doesn't have a deficit due to the monetary transfers between TPs and TEs.

- The existing mechanisms are based on either task provider-centric (i.e. crowdsourcer-centric) or task executor-centric (i.e. smartphone user-centric) approach. In a task provider centric incentive mechanism, TP has full control over the total payments to the TEs. On the other hand, in executor-centric incentive mechanism, the roles of TP and TE are opposite. Here, it is very intuitive to say that both parties (i.e. TP and TE) can not be fairly benefited simultaneously. The party who has more control over the payment process will try to be benefited more.

- Although some attempts have been made in the past to develop incentive mechanisms for multiple task providers (TPs) and multiple task executors (TEs) scenarios [79, 189, 190], none of them take RoI (Region of Interest)-based participatory sensing into account when formulating their problems.

5.1.1 Objective

In order to address the above mentioned limitations, our objective is to design a double auction based fair task allocation mechanism for ROI-based problem formulation. Here, we aim

TABLE 5.1: Smartphone sensors used for different Participatory sensing based applications/ systems

City dynamics	Application and/or reference	Smartphone sensors								Other sensor(s)
		GPS	Magmetometer	Accelerometer	Microphone	Proximity	Camera	Gyroscope	Light sensor	
Noise pollution	Ear-Phone [191]	✓		✓	✓	✓		✓	✓	
	NoiseTube [82]	✓			✓					
	NoiseSense [104]	✓			✓					
Air pollution	CitiSense [192]	✓								✓
	HazeWatch [187]	✓								✓
	AirSense [7]	✓								✓
Road condition	MagTrack [193]	✓	✓	✓						
	SPC [194]	✓		✓		✓				
	RoadSense [195]	✓		✓				✓		
	D&Rsense [196]	✓		✓						
Traffic condition	Vij *et al.* [197]	✓			✓					
	Liu *et al.* [183]	✓		✓	✓					
	Yan *et al.* [184]	✓		✓			✓			✓
Street illumination	LiSense [198]	✓							✓	

✓: The presence of check marks indicate the smartphone equipped sensors and other sensors that are required for some popular participatory sensing based applications/systems.

to use a double sided auction (i.e double auction) based incentive mechanism over a single sided auction based mechanism. This is because, in a double sided auction multiple TPs and multiple TEs can participate in the auction process simultaneously. In other words, both the parties (i.e. TPs and TEs) can be involved actively in the auction process. The double auction-based solution has been used here to solve the assignment problem by finding a bipartite matching between the bids of TPs to the asks of TEs. Specifically, finding a clearing price with assignment constraints can be used to find a bipartite matching. Now, the existing double auction based research suggest that the efficiency/effectiveness of a double auction technique can be assessed through certain important properties such as *truthfulness*, *computational efficiency*, *budget balance*, and *individual rationality* (more details in Section 5.4.3). As a result, the proposed incentive mechanism is intended to be designed in such a way that all desirable properties (see Section 5.4.3) are met.

5.1.2 Contribution

Now, in order to achieve the above mentioned objectives, a double auction based task allocation mechanism called TATA (Truthful Double Auction for Task Allocation) is proposed that yields a mapping between TPs and TEs (see Section 5.3 for more details). TATA can be used to efficiently allocate sensing tasks of the TPs to the TEs for execution. The desirable properties of TATA are validated through the rigorous experiments, and finally the performance of TATA is also compared with some of the existing incentive mechanisms in terms of various performance metrics. The double auction based incentive mechanism fits well for allocating the sensing tasks of TPs to TEs for any city dynamics. Here, a set of TPs, a set of TEs and an auctioneer are involved in the auction. The proposed auction based incentive mechanism TATA is governed by the auctioneer who is a trustworthy third party. The TPs may have sensing tasks at different $ROIs$ for which they place the bids. Similarly, the TEs place the asks (i.e. the offer price for the sensing tasks) for performing the task at different $ROIs$. After that, the proposed algorithm TATA is used to find a set of winning TPs, a set of winning TEs,

a set of successful trades, the price charged to the TPs and the payments/incentives for the TEs. The key contributions of this work are summarized as follows:

- An effective incentive mechanism is proposed for participatory sensing based systems related to different city dynamics. The incentive mechanism is modeled as a double auction game to address multiple TPs and multiple TEs scenario.
- The proposed auction based incentive mechanism TATA is proved truthful, budget balanced, individually rational, and computationally efficient.
- The extensive simulation results show that TATA outperforms the existing auction based incentive mechanisms for participatory sensing and also yields high system efficiency.

5.2 Existing Works

There are several incentive mechanisms available for participatory sensing based systems that encourage participants to perform the sensing tasks [58, 59, 61, 63–65, 78, 79, 189, 190, 199–202]. The details of the existing works are presented in the Table 5.2.

Although various attempts have been made in the past to build incentive mechanisms for participatory sensing systems, the majority of them do not consider multiple task providers (TPs) as well as multiple task executors (TEs) for a given city dynamics. In this context, the proposed incentive mechanism TATA is designed using a double auction game that can handle multiple TPs and TEs while also satisfying all of the desirable properties (e.g., truthfulness, computational efficiency, budget balance, and individual rationality) of an auction-based incentive mechanism. TATA is designed in such a way that neither the TPs nor the TEs have direct control over the incentive mechanism. Instead, a trustworthy third party (i.e. auctioneer) conducts the auction. That helps both parties to be fairly benefited simultaneously. Most of the existing works [58–61, 63–65, 79, 189, 190, 199, 201, 203] do not consider RoI (Region of Interest)-based participatory sensing for their problem formulation. Although the region-specific sensing tasks are considered in SPARC [78], it provides a solution approach by conducting a separate auction for each region. However, conducting a separate auction for each region (i.e., RoI) is inefficient. It is because in such a case, due to many RoIs a TE/TP has to participate in multiple auctions. Therefore a novel auction framework is proposed in our work that allows bidding/asking by considering multiple ROIs for multi-demand heterogeneous sensing task allocation market. Unlike the above-mentioned existing incentive mechanisms in the crowd/participatory sensing systems, the proposed model TATA introduces a novel pricing mechanism. Moreover, in contrast to the existing approaches, a different task allocation mechanism is provided. Specifically, in TATA, the sensing task allocation process is basically a bipartite matching between TEs and TPs that allows to achieve high system efficiency along with all the desirable properties. The existing incentive mechanisms usually suffer from several problem-specific assumptions. For instance, some commonly used assumptions are: the smartphones (i.e. task executors) do not lie about their sensing time, the sensing tasks do not manipulate their demands, etc. However, in the case

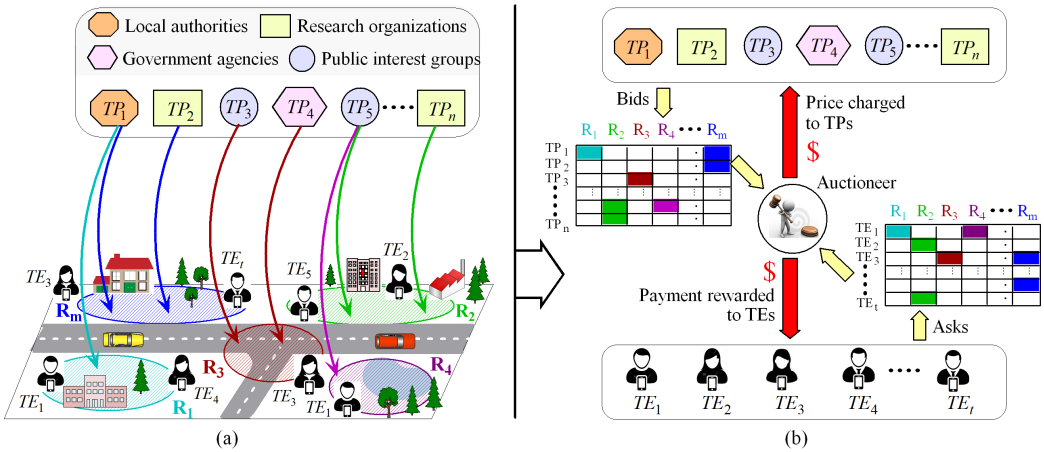


FIGURE 5.1: A market of sensing task allocation between TPs and TEs. TP_i, TE_k, and R_j represent i^{th} task provider, k^{th} task executor and j^{th} ROI respectively. (a) TPs have sensing tasks at different $ROIs$. The TEs are able to execute the tasks at different $ROIs$. (b) The TPs place bids to the auctioneer for those $ROIs$ where the sensing tasks need to be performed. The TEs place asks (offer price for executing the tasks) to the auctioneer for those $ROIs$ where they can perform sensing tasks. Auctioneer decides the price charged to the winning TPs and the incentive/ reward for the winning TEs.

of TATA, no such assumptions are required. Moreover, in contrast to our work, the existing approaches [79, 189] have different problem-specific design objectives. For instance, in DPDA [189], the main objective of the authors is to protect against inference attacks in the MCS auction markets. Also, DPDA is intended to maximize the platform's (i.e., auctioneer) revenue by formulating a platform revenue maximization (PRM) problem. Similarly, Deng et al. [79] build a market mechanism that maximizes the revenue of the market maker (i.e., auctioneer). However, instead of maximizing the auctioneer's revenue, the aim of the proposed incentive model is to come up with the task allocation such that both the TPs and TEs can be fairly benefited simultaneously.

5.3 Problem Statement

In this section, first we provide a system model for an auction based incentive mechanism. After that, based on the presented system model, the problem will be formulated as a double sided auction game.

5.3.1 System Model

As shown in Fig. 5.1, consider a market of sensing task allocation between TPs and TEs for any city dynamics. Also, consider a scenario where the TPs announce sensing tasks for different Region of Interests ($ROIs$). The TEs that are capable to access the $ROIs$ can perform those tasks. It is noted that the TPs and TEs may value differently to the sensing tasks for different $ROIs$. Now, the incentive mechanism design for TEs in this market scenario can be performed using a double auction game. Traditionally, a double auction [210] is a way of trading commodities where there are multiple buyers, multiple sellers, and an auctioneer involved. In a double auction, the auctioneer acts as a trustworthy third party who receives

TABLE 5.2: A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics. (✓: The requirement/characteristic has been met, × : The requirement/characteristic has not been met, IC : Incentive Compatibility, IR: Individual Rationality, BB: Budget Balance, CE: Computational Efficiency, LA: Location awareness, HMTA: Heterogeneous Multi-Task Assignment, MAAIM: Mental accounting auction incentive mechanism)

Ref and year	Method	Contribution	Assumption	Method design	Pricing	Properties				LA
						IC	IR	BB	CE	
Zhang et al. [75], 2015	Double auction	Provide auction-based incentives mechanisms using the three auction-based models	The assumption is- each job comprises of a various small tasks, and the winner can work on a subset of these tasks.	The double auction is designed between requesters and providers	They determined the critical value as the payment of each winner provider.	✓	✓	✓	✓	×
Li et al. [204], 2019	MAAIM	They introduced mental accounting (MA) theory into designing incentive mechanism for crowdsensing.	–	They build an incentive mechanism based on MA theory's reference dependency and loss aversion and study its utility function.	In accordance with the QoD that the winner has provided, the platform decides the winner's payment.	✓	×	✓	✓	×
Wang et al. [205], 2019	HMTA	They proposed an interconnected task assignment strategy that includes both intra- and inter-task assignment phases.	They made an assumption that the reference task can fully divide the spatiotemporal contexts of all the tasks.	They designed a two-stage task distribution process to look for the appropriate users and allocate jobs in accordance with user preferences.	More rewards are given to active users if they are asked to gather sensor readings in more cycle-region pairs as opposed to providing each user the same amount of incentives.	×	×	×	×	✓
Jin et al. [67], 2015	Reverse auction	Proposed a quality-aware incentive mechanism for MCS systems	They made an assumption that the total QoI that all winners have on a particular task is the sum of the QoIs of the winners that execute the task	Create a reverse auction mechanism that, roughly, maximises social welfare.	Pricing mechanism considers both the bidding price and QoI as the factors	✓	✓	×	✓	×
Wen et al. [60], 2015	Reverse Vickrey auction	For MCS, they suggest a quality-aware auction-based incentive mechanism.	The lowest payment that is acceptable by the user will always be the true cost of the user	Create a reverse auction mechanism that maximises social welfare.	A Vickrey-Clarke-Groves-based rule is used to determine user payments.	✓	✓	×	×	×
Zhang et al. [110], 2022	Reverse auction	They developed an incentive mechanism based on nondeterministic mobility for effective crowdsensing data collection.	–	They created an incentive mechanism that maximises the system's social welfare for efficient data collection.	They create greedy algorithms to handle the user selection and price calculation problems.	✓	✓	×	✓	✓
Esmaeilyfard et al. [69], 2022	Stackelberg game	They provide a game-theoretic incentive mechanism for a multi-publisher and multitask mobile crowdsensing system.	–	They use Stackelberg model where the task publishers act as leader and users act as followers.	The suggested model executes the best reward distribution depending on user demands.	×	×	×	✓	×
Li et al. [70], 2022	Stackelberg game	For the strategic interactions in privacy-preserving MCS scenario, they suggest a three-stage Stackelberg incentive system.	They made an assumption that each task requester is limited to one job per time period.	They define the incentive scheme inside the T rounds' interactions, which is characterised as a 3-phase Stackelberg game	The platform suggests that the total incentives should be smaller than the payment made.	×	✓	✓	✓	×
Ni et al. [76], 2021	Double auction	On the basis of the double auction and taking into account worker dependabilities, they suggest a differentially private mobile crowdsensing incentive model.	They make the assumption that the auctioneer can be trusted, and that the adversary won't be able to steal information about bids.	They create the incentive system by applying the exponential process in double auction to pick the clearing price.	They employ uniform pricing where winner workers are paid the same clearing paying price and winner requesters are charged the same clearing charging price.	✓	✓	✓	✓	×
Chen et al. [111], 2021	Reverse auction	They provide an incentive scheme for vehicular crowdsourcing that considers the unpredictable trip duration of the vehicle.	When vehicles enter a data source's transmission range, they assume it transmits at a constant data rate.	The approach is represented as a single reverse auction, with the service requester acting as service buyers and the participating automobiles acting as service sellers.	Every winner is given the crucial value, or the maximum offer price, available to determine payment throughout the payment determination process.	✓	✓	✓	✓	×

TABLE 5.2: (Continued). A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics. (✓: The requirement/characteristic has been met, × : The requirement/characteristic has not been met, IC : Incentive Compatibility, IR: Individual Rationality, BB: Budget Balance, CE: Computational Efficiency, LA: Location awareness, HMTA: Heterogeneous Multi-Task Assignment, MAAIM: Mental accounting auction incentive mechanism)

Ref and year	Method	Contribution	Assumption	Method design	Pricing	Properties				LA
						IC	IR	BB	CE	
Chen et al. [77], 2016	Double auction	For two-sided heterogeneous mobile crowdsensing marketplaces, they presented a truthful double auction.	They make the assumption that there is no user collusion during the auction and that each user's stated expenses are unique.	Design a two-sided auction with an emphasis on creating a system to maximise social welfare	If the requester wins, it must pay a crucial price to the platform. For users, they create a price structure akin to VCG.	✓	✓	✓	✓	×
Chen et al. [78], 2013	Double auction	They developed a double auction that is strategy-proof for mobile participatory sensing.	They have made the assumption that the demand for sensing time cannot be manipulated by the sensing task.	They create auction mechanisms that preserve individual rationality and incentive-compatibility while also maximising overall social welfare.	SPARC greedily distributes sensing time to participative sensing activities, charging them a fee unrelated to their cost or valuation.	✓	✓	×	×	×
Jin et al. [189], 2018	Double auction	Provide a solution in order to protect against inference attacks in the MCS auction markets.	They presume the platform is reliable, i.e., it does not use its information of participants' inputs to influence auction outcome.	Incentive mechanism is design to maximize the platform's (i.e., auctioneer) revenue by formulating a platform revenue maximization (PRM) problem.	Uniform pricing is used to determine payments to winner worker and charges to winner requestor.	✓	✓	✓	✓	×
Deng et al. [79], 2014	Double auction	They investigate a double auction market design in which the auctioneer seeks to maximise its overall revenue	They make the assumption that the probability distributions of buyer and seller valuations are independent.	They take into account the single-dimensional double auction design issue in which all vendors offer the same goods.	They make use of discriminative pricing models	✓	✓	✓	✓	×
Zhang et al. [206], 2021	auction game	They presented a two-stage incentive model for crowdsensing named as group-buying-based online incentive mechanism.	They assumed that actions with a shared necessity for a certain place may be automatically combined to acquire and share user sensing abilities.	First, a price learning algorithm is designed. Next, group agents and users engage in an online auction before a predetermined deadline for recruitment.	They select one price with a specific probability to assess each job from a pool of possible prices.	✓	✓	✓	✓	×
Jin et al. [80], 2017	Double auction	They presented a framework for crowdsensing that includes an incentive model.	They presume that the platform already understands the reliability level matrix and has a history of it.	The incentive mechanism was designed in a manner that will enable the most user engagement.	Every winning worker requester receives the supremum and infimum of her bid from the pricing process.	✓	✓	×	✓	×
Peng et al. [207], 2018	EM and IT	The development of a quality-based reward system is presented, which would directly encourage each user to provide high-quality sensing data for long-term crowdsensing.	–h	To quantify the quality of sensing data, they develop the Expectation Maximization method, and they utilise classical information theory to determine the useful data contributions.	They decide whether rewards are reasonable and appropriate for the users based on the projected quality and contribution.	✓	✓	×	✓	×
Zhan et al. [208], 2020	Reinforcement Learning	They model the problem as a Stackelberg game with many leaders and followers.	It is assumed that the cost function is monotonically growing, strictly convex, and differentiable.	The MCS system is designed as a multi-leader multi-follower Stackelberg game.	The most effective price policy is discovered using the multi-agent dynamic incentive mechanism.	—	—	—	—	×
Zhan et al. [209], 2021	Stackelberg game	They developed a framework for an incentive system for mobile crowdsensing with unpredictable demand.	–	A two-stage Stackelberg game is designed to develop the incentive mechanism.	The greedy and random pricing systems are outperformed by dynamic incentive mechanism.	—	—	—	—	×
Proposed work	Double auction	Come up with an allocation by finding a bipartite matching between TEs and TPs such that both the task provider and task executors can be fairly benefited simultaneously	–	Capture location (i.e., RoI)-based crowd sourcing and crowd sensing market scenario of sensing task allocation between TPs and TEs, a set of TPs,a set of TEs and an auctioneer.	The price charged to the winning TPs and payments for the winning TEs in our proposed approach are determined by the threshold ask and the number of successful trades (see Algorithm 4).	✓	✓	✓	✓	✓

TABLE 5.3: Notation

Symbols	*Description*	*Symbols*	*Description*
TP_i	i^{th} task provider	TE_k	k^{th} task executor
R_j	j^{th} Region of Interest (RoI)	n	Total number of TPs participating in auction
t	Total number of TEs participating in auction	m	Total number of RoIs
$\mathscr{P}$	Set of task providers	$\mathcal{E}$	Set of task executors
$\mathcal{R}$	Set of RoIs	$\mathscr{P}_w$	Set of winning task providers
$\mathcal{E}_w$	Set of winning task executors	$\mathscr{S}^{\mathscr{T}}$	Set of successful trades
D_i^j	Bid of task provider $TP_i \in \mathscr{P}$ for RoI $R_j \in \mathcal{R}$	$\mathbf{D_i}$	Bid vector of task provider TP_i
$\mathbf{D}$	Bid matrix consisting of all bid vectors	A_k^j	Ask of task executor $TE_k \in \mathcal{E}$ for $R_j \in \mathcal{R}$
$\mathbf{A_k}$	Ask vector of TE_k	$\mathbf{A}$	Ask matrix consisting of all ask vectors
V_i^j	Valuation of TP_i for sensing service at R_j	$\mathbf{V_i}$	Valuation vector of TP_i
$\mathbf{V}$	Valuation matrix consisting of all valuation vectors.	C_k^j	Actual sensing cost of TE_k at R_j
$\mathbf{C_k}$	Cost vector of all TE_k	$P_i^{(p)}$	Total price that the $TP_i \in \mathscr{P}_w$ is charged
$P_k^{(e)}$	Total payment that the $TE_k \in \mathcal{E}_w$ is rewarded	$P_{ij}^{(p)}$	Price charged to TP_i for a sensing service at R_j
$P_{kj}^{(e)}$	Payment rewarded to TE_k for the sensing task at R_j	$\mathcal{U}_i^{(p)}$	Total utility of $TP_i \in \mathscr{P}_w$
$\mathcal{U}_k^{(e)}$	Total utility of $TE_k \in \mathcal{E}_w$		

the bids and asks from the buyers and sellers, respectively. The auctioneer is responsible for organising and managing the auction, including choosing the clearing price and payment as well as how the trade's commodities will be distributed. To the best of our knowledge, TATA could be incorporated in any of the practical scenario (online or offline) for allocation of sensing tasks of the task providers to the task executors in participatory sensing systems. For instance, the auctioneer can call for the asks and the bids for a specified time frame. After the time is over the collected asks and the bids can be used for candidate trade determination, successful trade determination and price calculation. Again for the next time frame, auctioneer may collect the asks and the bids in order to conduct another auction mechanism.

Now, in our market scenario of sensing task allocation between TPs and TEs, a set of TPs, a set of TEs and an auctioneer participate in the double auction. The auction process is driven entirely by the auctioneer. Here, the auctioneer functions between TPs and TEs as a trusted third party. The TPs place the bids to the auctioneer for the sensing tasks at different $ROIs$ in the form of a bid matrix. Similarly, TEs place the asks (i.e. offer price) to the auctioneer for sensing task execution at different $ROIs$ in the form of an ask matrix. These bids and asks are collected by the auctioneer. Note that the bidding price of a TP for a particular task indicates the valuation of that task to the TP. On the other hand the asking price of a TE for a particular task indicates the cost of the TE for executing the task. However, different TPs may value the same task differently. Similarly, different TEs may also calculate the cost of a task differently. It is because, in the participatory sensing systems, it is not always be the case that the price came from a generalized and physically realized fashion. Here, the assumption is that the TPs, TEs and the auctioneer do not collude [211]. Now, the auctioneer comes up with an allocation between the TPs and TEs from the submitted asks and bids. In addition, the auctioneer determines the price charged to the winning TPs and payments/ rewards/ incentives for the winning TEs.

Definitely, the trading between the TPs and TEs should fulfill certain requirements so

that both TPs and TEs can be benefited simultaneously. The task executors need to be incentivized for executing the sensing tasks. In particular, a task executor cannot be rewarded less than its actual cost for task execution and a task provider cannot be charged more than its valuation to the sensing task.

5.3.2 Problem Formulation

The proposed auction involves three parties namely the task providers who announce a set of sensing tasks at different *ROIs*, task executors who execute the sensing task and an auctioneer (a trustworthy third party) who hosts the auction and allocates the task of the TPs to TEs.

Task Providers

The set of task providers are denoted as $\mathscr{P} = \{TP_1, TP_2, TP_3, \ldots, TP_n\}$. Let us consider, $\mathcal{R} = \{R_1, R_2, R_3, \ldots, R_m\}$ is the set of all region of interests (*ROIs*). For each task provider $TP_i \in \mathscr{P}$, its bids for the region of interests (ROI) $R_j \in \mathcal{R}$ are denoted by a bid vector $\mathbf{D}_i = (D_i^1, D_i^2, \ldots, D_i^m)$, where D_i^j is the bid of $TP_i \in \mathscr{P}$ for a sensing task at $R_j \in \mathcal{R}$. In other words, $D_i^j \in \mathbf{D}_i$ denotes the maximum price that can be spent by $TP_i \in \mathscr{P}$ for the sensing task at $R_j \in \mathcal{R}$ for any task executor. Here, $D_i^j \in \mathbf{D}_i$ is defined as follows:

$$D_i^j = \begin{cases} 0, & \text{if } TP_i \in \mathscr{P} \text{ has no sensing task at } R_j \\ b_{ij} \in \mathbb{R}^+ - \{0\}, & \text{otherwise} \end{cases} \tag{5.1}$$

Let $\mathbf{D}$ denotes a bid matrix consisting of all $\mathbf{D}_i$'s i.e $\mathbf{D} = (\mathbf{D}_1; \mathbf{D}_2; \ldots; \mathbf{D}_n)$. For each $TP_i \in \mathscr{P}$, its valuation vector is represented as $\mathbf{V}_i = (V_i^1, V_i^2, \ldots, V_i^m)$, where, V_i^j is the valuation of the sensing task at *ROI* $R_j \in \mathcal{R}$ to TP_i. Here, $V_i^j \in \mathbf{V}_i$ is defined as follows:

$$V_i^j = \begin{cases} 0, & \text{if } D_i^j = 0 \\ d_{ij} \in \mathbb{R}^+ - \{0\}, & \text{otherwise} \end{cases} \tag{5.2}$$

It should be noted that, $\forall_{i,j}\ V_i^j \geq D_i^j$. For each $TP_i \in \mathscr{P}$, the need vector is denoted as $\mathbf{N}_i = (N_i^1, N_i^2, \ldots, N_i^m)$, where, N_i^j represents the number of task executors required to perform the task at $R_j \in \mathcal{R}$ for task provider TP_i's sensing task. Let the need matrix consisting of the need vector of all $TP_i \in \mathscr{P}$ is denoted as $\mathbf{N} = (\mathbf{N}_1; \mathbf{N}_2; \ldots; \mathbf{N}_n)$. Here, $N_i^j \in \mathbf{N}_i$ is defined as follows:

$$N_i^j = \begin{cases} 0, & \text{if } D_i^j = 0 \\ e_{ij} \in \mathbb{Z}^+ - \{0\}, & \text{otherwise} \end{cases} \tag{5.3}$$

Task Executors

The set of task executors is denoted as $\mathcal{E} = \{TE_1, TE_2, TE_3, \ldots, TE_t\}$. The task executors submit the minimum price at which they can perform the sensing tasks. This minimum price is known as Ask (or offer price). The asks of $TE_k \in \mathcal{E}$ for all $R_j \in \mathcal{R}$ is denoted by an ask

vector $\mathbf{A}_k = (A_k^1, A_k^2, \ldots, A_k^m)$, where A_k^j denotes the ask (i.e. offer price) of $TE_k \in \mathcal{E}$ for sensing task at $R_j \in \mathcal{R}$. Here, A_k^j is defined as follows:

$$A_k^j = \begin{cases} 0, & \text{if } TE_k \in \mathscr{E} \text{ has no ask for } R_j \in \mathcal{R} \\ a_{kj} \in \mathbb{R}^+ - \{0\}, & \text{otherwise} \end{cases} \tag{5.4}$$

The ask matrix that consist of all the ask vectors of all task executors is denoted as $\mathbf{A} = (\mathbf{A}_1; \mathbf{A}_2; \ldots; \mathbf{A}_t)$. For each task executor $TE_k \in \mathcal{E}$, C_k^j denotes the actual cost of sensing at *ROI* $R_j \in \mathcal{R}$. The actual sensing costs of TE_k for all the *ROIs* in $\mathcal{R}$ is denoted by the cost vector $C_k = (C_k^1, C_k^2, \ldots, C_k^m)$. It should be noted that, $\forall_{k,j}\, C_k^j \leq A_k^j$.

Auctioneer

The auctioneer is a trustworthy third party who host the auction. Given, $\mathscr{P}, \mathcal{E}, \mathbf{D}$ and $\mathbf{A}$ the auctioneer determines the winning task provider set $\mathscr{P}_w \subseteq \mathscr{P}$, the winning task executor set $\mathcal{E}_w \subseteq \mathcal{E}$, a set of successful trades $\mathscr{S}^{\mathscr{T}}$ (see Definition 1 and Definition 2) between $\mathscr{P}_w$ and $\mathcal{E}_w$, the total price $P_i^{(p)}$ that the winning task provider $TP_i \in \mathscr{P}$ is charged, and the total payment $P_k^{(e)}$ that the winning task executor $TE_k \in \mathcal{E}$ is rewarded. The term $P_{ij}^{(p)}$ is used to denote the price that the winning task provider $TP_i \in \mathscr{P}$ is charged for having sensing service for *ROI* $R_j \in \mathcal{R}$. Similarly, $P_{kj}^{(e)}$ is used to denote the price that the winning task executor $TE_k \in \mathcal{E}$ is rewarded for performing the sensing task for *ROI* $R_j \in \mathcal{R}$.

Definition 1 *(Trade) A trade is defined as a triplet (i, j, k), if the bid D_i^j of $TP_i \in \mathscr{P}$ for $R_j \in \mathcal{R}$ and the ask A_k^j of $TE_k \in \mathcal{E}$ for $R_j \in \mathcal{R}$ are non-zero positive values. In other words, if TP_i has a sensing task at ROI R_j and there is a task executor TE_k who is also willing to execute task at ROI R_j, then trade can take place between them.*

Definition 2 *(Successful Trade) Suppose we are given a set of winning task executors $\mathcal{E}_w \subseteq \mathcal{E}$, a set of winning task providers $\mathscr{P}_w \subseteq \mathscr{P}$, and a set of Region of Interests $\mathcal{R}$. A successful trade is a trade (i, j, k), where, the auctioneer assigns a task executor $TE_k \in \mathcal{E}_w$ to perform a task at $R_j \in \mathcal{R}$ for the task provider $TP_i \in \mathscr{P}_w$. A set of all such successful trades is denoted as $\mathscr{S}^{\mathscr{T}}$, i.e. $\mathscr{S}^{\mathscr{T}} = \{(i, j, k) \mid TP_i \in \mathscr{P}_w, TE_k \in \mathcal{E}_w, R_j \in \mathcal{R}$ and TP_i takes sensing services from TE_k for ROI $R_j\}$*

Now, the utilities of the *TPs* and the *TEs* not only depend on the price charged and the payment rewarded but also depend on the valuations of the *TPs* toward the acquired sensing services and the cost for performing such sensing tasks by the *TEs*. The *total utility* of a task provider $TP_i \in \mathscr{P}_w$ is defined as the difference between TP_i's total valuation toward the acquired sensing services and the total price charged:

$$\mathcal{U}_i^{(p)} = \begin{cases} \sum\limits_{(i,j,k) \in \mathscr{S}^{\mathscr{T}}} V_i^j - \sum\limits_{(i,j,k) \in \mathscr{S}^{\mathscr{T}}} P_{ij}^{(p)}, & if\ TP_i \in \mathscr{P}_w \\ 0, & otherwise \end{cases} \tag{5.5}$$

Similarly, the *total utility* of a task executor $TE_k \in \mathcal{E}_w$ is defined as the difference between TE_k's total payment received and its the total cost for the sensing task:

$$\mathcal{U}_k^{(e)} = \begin{cases} \sum_{(i,j,k)\in\mathscr{S}^{\mathscr{T}}} P_{kj}^{(e)} - \sum_{(i,j,k)\in\mathscr{S}^{\mathscr{T}}} C_k^j, & if\ TE_k \in \mathcal{E}_w \\ 0, & otherwise \end{cases} \tag{5.6}$$

The utility calculation for TPs and TEs may be required in the candidate trade determination phase (see Figure 4 and the example given in Section 4.2), the performance evaluation of the incentive mechanism (see Section 5.3), and the proof of truthfulness/incentive compatibility property of TATA (see Section 4.3).

Given the details of the bids and the asks, the goal of this work is to build an incentive mechanisms for crowd/participatory sensing systems that allows to perform a fair allocation of tasks (i.e., no market manipulation) and also satisfy some desirable properties (see Section 5.4.3 for more details). In order to do that, we need to have a sensing task allocation process, which is basically an assignment/allocation/mapping between TEs and TPs. As shown in Fig. 5.2, first the bid matrix (**D**) and the ask matrix (**A**) are used to find non-zero positive asks ($\mathscr{N}$) and bids ($\mathscr{M}$). Next, from the possible trades (i,j,k), TATA shortlists a set of candidate trades ($\mathscr{T}$). These candidate trades involve those TPs and TEs that are satisfying some criteria/condition of the auction (more details are provided later in this section). The set of candidate trades are further refined based on each provider's need value (obtained from **N**) to find the set of successful trades ($\mathscr{S}^{\mathscr{T}}$). Finally, the price charged to TPs ($P_w^{(p)}$) and the payment/ reward/ incentive given to TEs ($P_w^{(e)}$) are decided.

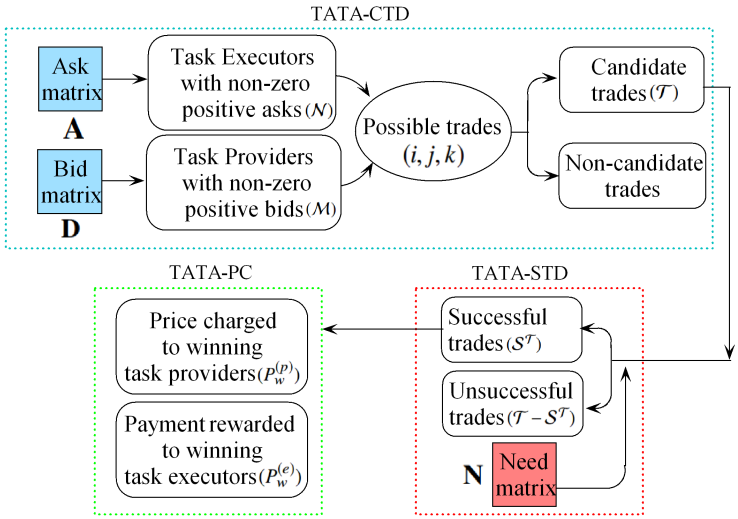


FIGURE 5.2: Three major stages of TATA: TATA-CTD, TATA-STD and TATA-PC.

Algorithm 2 TATA($\mathscr{P}$,$\mathcal{E}$)

Input: Set of task providers $\mathscr{P}$, Set of task executors $\mathcal{E}$.

Output: Set of winning task providers $\mathscr{P}_w$, Set of winning task executors $\mathcal{E}_w$, Set of successful trades $\mathscr{S}^{\mathscr{T}}$, $P_w^{(p)}$, $P_w^{(e)}$.

1: **for** each $TP_i \in \mathscr{P}$ **do**
2: Collect bids in the form of maximum price that can be spent by $TP_i \in \mathscr{P}$ for the sensing task at various ROIs $R_j \in \mathcal{R}$.
3: Create bid vector $\mathbf{D}_i = (D_i^1, D_i^2, \ldots, D_i^m)$, where D_i^j is the bid of $TP_i \in \mathscr{P}$ for a sensing task at $R_j \in \mathcal{R}$.
4: Create need vector $\mathbf{N}_i = (N_i^1, N_i^2, \ldots, N_i^m)$, where, N_i^j is the number of TEs required to perform the task at $R_j \in \mathcal{R}$.
5: **end for**
6: Create bid matrix $\mathbf{D} = (\mathbf{D}_1; \mathbf{D}_2; \ldots; \mathbf{D}_n)$ and need matrix $\mathbf{N} = (\mathbf{N}_1; \mathbf{N}_2; \ldots; \mathbf{N}_n)$.
7: **for** each $TE_k \in \mathscr{E}$ **do**
8: Collect ask values for $TE_k \in \mathscr{E}$ denoting the cost of sensing at various ROIs $R_j \in \mathcal{R}$.
9: **end for**
10: Create ask matrix $\mathbf{A} = (\mathbf{A}_1; \mathbf{A}_2; \ldots; \mathbf{A}_n)$
11: $(\mathcal{B}, \mathscr{P}_w, A_\phi, \mathscr{T}) \leftarrow$ TATA-CTD($\mathscr{P}$,$\mathcal{E}$, $\mathbf{D}$, $\mathbf{A}$);
12: $(\mathcal{E}_w, \mathscr{S}^{\mathscr{T}}) \leftarrow$ TATA-STD($\mathcal{B}, \mathscr{T}, \mathbf{N}$);
13: $(P_w^{(p)}, P_w^{(e)}) \leftarrow$ TATA-PC($\mathscr{P}_w, \mathcal{E}_w, \mathscr{S}^{\mathscr{T}}, A_\phi$);
14: **return** $(\mathscr{P}_w, \mathcal{E}_w, \mathscr{S}^{\mathscr{T}}, P_w^{(p)}, P_w^{(e)})$;

5.4 Truthful Double Auction Based Incentive Mechanism for Task Allocation

In this section, a truthful double auction based incentive mechanism TATA is presented for sensing tasks allocation in participatory sensing environment. The details of the algorithm TATA followed by an illustrative example is provided. The desirable properties (namely budget balance, individual rationality, computational efficiency and truthfulness/ incentive compatibility) of the proposed double auction TATA are also analyzed.

5.4.1 Details of TATA

The proposed double auction mechanism TATA (Algorithm 1) consists of three major stages (Fig. 5.2), namely *candidate trade determination* (TATA-CTD), *successful trade determination* (TATA-STD) and *price calculation* (TATA-PC).

The bid and ask matrices are created in Algorithm 1 after collecting the bids and asks from the TPs and TEs (line 1-10). Note that the ROIs could be important for the TPs and TEs in order to decide the bid and the ask prices respectively. In other words, depending on the ROIs, the selection of the bids and asks values for any task could be varied. However, the aspects of the ROIs that influence the bidding and asking process are task-specific. For example, in the case of participatory sensing based pollution data collection task, depending on the proximity of the ROIs, a TE may have different sensing cost, hence the ask values could be different. Particularly, if an ROI is located geographically close to a TE's location, the TE is expected to place a lower ask value for that ROI. On the other hand, a TP, e.g., a research

group, that is providing the sensing task for pollution monitoring, may place different bids for different functional regions (like educational areas, residential areas, etc.) depending on their research requirements.

TATA invokes TATA-CTD($\mathscr{P}$,$\mathcal{E}$, **D**, **A**) to determine a set of candidate trades ($\mathscr{T}$), and the winning task providers (see Algorithm 2). TATA-CTD() takes a set of task providers $\mathscr{P}$, a set of task executors $\mathcal{E}$, the bid matrix **D** and the ask matrix **A** as input to the function. Algorithm 2 first creates a new set of task providers $\mathscr{M}$ from the original task providers' set $\mathscr{P}$. For any task provider $TP_i \in \mathscr{P}$, the set $\mathscr{M}$ will contain TP_i^j if and only if $D_i^j > 0$. It means a task provider $TP_i \in \mathscr{P}$ appears in $\mathscr{M}$ for those $R_j \in \mathcal{R}$ for which it has non-zero positive bids. After that, the task providers in $\mathscr{M}$ are sorted in non-increasing order of non-zero positive bids to construct a list $\mathscr{M}' = \langle TP_{x_1}^{y_1}, TP_{x_2}^{y_2}, \ldots, TP_{x_v}^{y_v} \rangle$, where $v = |\mathscr{M}|$. Similarly, for task executors, a new set $\mathscr{N}$ of executors are created from the original task executors' set $\mathcal{E}$. For $TE_k \in \mathcal{E}$, the set $\mathscr{N}$ will contain TE_k^j if and only if $A_k^j > 0$. Then, the task executors in $\mathscr{N}$ are sorted in non-decreasing order of non-zero positive asks to construct a list $\mathscr{N}' = \langle TE_{l_1}^{m_1}, TE_{l_2}^{m_2}, \ldots, TE_{l_r}^{m_r} \rangle$, where $r = |\mathscr{N}|$. Next, the threshold ask is determined which is nothing but the ask value of the median task executor in $\mathscr{N}'$, represented by $A_\phi = TE_{l_\phi}^{m_\phi}$ where, $\phi = \lceil (r+1)/2 \rceil$. This threshold ask A_ϕ is used to construct a set $\mathscr{M}'_q$ having first q task providers in $\mathscr{M}'$ such that $D_{x_q}^{y_q} \geq A_\phi$. Now, for each $TP_i^j \in \mathscr{M}'_q$, if there is a A_k^j such that $A_k^j < A_\phi$, we add the task provider TP_i in winning task providers' set $\mathscr{P}_w$. Here, we create a set $\mathfrak{B}_i^j$ that contains those $TE_k^j \in \mathscr{N}$ for which the condition $A_k^j < A_\phi$ is satisfied. The set $\mathfrak{B}_i^j$ consists of all the task executors that are won by task provider TP_i^j. Also, for each combination $(TP_i^j \in \mathscr{M}'_q\ ,\ TE_k^j \in \mathscr{N})$ that satisfies above condition a trade (i,j,k) is created and added to the set of candidate trades $\mathscr{T}$. After the for loop, all $\mathfrak{B}_i^j$'s are combined to form a new set $\mathcal{B}$.

After candidate trade determination phase, TATA invokes TATA-STD($\mathcal{B}, \mathscr{T}, \mathbf{N}$) for successful trade determination (see Algorithm 3). If the number of task executors in $\mathfrak{B}_i^j$ that are won by the task provider TP_i^j is more than its need (line 4 of Algorithm 3), then the executor elimination is required. The auctioneer keeps only those N_i^j task executors for which the total utility achieved by them is maximized. To do this, first the executors in $\mathfrak{B}_i^j$ are sorted according to the non-decreasing order of their ask values (line 5). As we approach the end of the list $\mathfrak{B}_{ij}^{(s)}$ the utility (i.e. $A_\phi - A_k^j$ values) by the executors decrease in truthful auction. Eventually, the auctioneer keeps the first $r = N_i^j$ executors of the sorted list $\mathfrak{B}_{ij}^{(s)}$ and rest of the executors are removed (line 6-line 9). After candidate elimination, the remaining trades in $\mathscr{T}$ are considered as successful trades $\mathscr{S}^{\mathscr{T}}$ (line 12). Next, the set of winning task executors are found (line 13-line 17) for which there is at least one trade in $\mathscr{S}^{\mathscr{T}}$.

After successful trade determination stage, TATA calls TATA-PC($\mathscr{P}_w, \mathcal{E}_w, \mathscr{S}^{\mathscr{T}}, A_\phi$) to perform price calculation (see Algorithm 4). For each winning task provider $TP_i \in \mathscr{P}_w$, the total number of successful trades c_i involving provider TP_i is found. Then total price charged to $TP_i \in \mathscr{P}_w$ is calculated as $P_i^{(p)} = c_i \times A_\phi$, where the threshold ask A_ϕ is considered as the clearing price. Similarly, for each winning task executor $TE_k \in \mathcal{E}_w$, the payments are calculated (line 12-line 20). It is worth noting that, in the pricing stage, the process of determining the threshold ask is also the process of determining the critical value. The

Algorithm 3 TATA-CTD($\mathscr{P}$,$\mathcal{E}$, **D**, **A**)

Input: Set of task providers $\mathscr{P}$, Set of task executors $\mathcal{E}$, Bid matrix **D**, Ask matrix **A**

Output: Set of winning task providers $\mathscr{P}_w$, Set $\mathcal{B}$ that contains all $\mathfrak{B}_x^w$, The threshold ask A_ϕ, A set of candidate trades $\mathscr{T}$.

1: $\mathscr{P}_w \leftarrow \emptyset$, $\mathscr{T} \leftarrow \emptyset$;
2: Construct a set $\mathscr{M} = \{TP_x^y \mid D_x^y > 0,\ TP_x \in \mathscr{P},\ R_j \in \mathcal{R}\}$ based on **D**;
3: Sort the task providers in $\mathscr{M}$ to obtain a list $\mathscr{M}' = \langle TP_{x_1}^{y_1}, TP_{x_2}^{y_2}, \ldots, TP_{x_v}^{y_v} \rangle$ such that $D_{x_1}^{y_1} \geq D_{x_2}^{y_2} \geq \ldots \geq D_{x_v}^{y_v}$
4: Construct a set $\mathscr{N} = \{TE_l^m \mid A_l^m > 0,\ TE_l \in \mathcal{E},\ R_m \in \mathcal{R}\}$ based on **A**;
5: Sort the task executors in $\mathscr{N}$ to obtain a list $\mathscr{N}' = \langle TE_{l_1}^{m_1}, TE_{l_2}^{m_2}, \ldots, TE_{l_r}^{m_r} \rangle$ such that $A_{l_1}^{m_1} \leq A_{l_2}^{m_2} \leq \ldots \leq A_{l_r}^{m_r}$
6: Find the threshold ask $A_\phi = A_{l_\phi}^{m_\phi}$ where, $\phi = \lceil (r+1)/2 \rceil$;
7: Find the largest value of q such that $D_{x_q}^{y_q} \geq A_\phi$ and $D_{x_{q+1}}^{y_{q+1}} < A_\phi$.
8: Construct a set $\mathscr{M}'_q$ with first q providers in $\mathscr{M}'$.
9: **for** each $TP_i^j \in \mathscr{M}'_q$ **do**
10: **for** each $TE_k^j \in \mathscr{N}$ **do**
11: **if** $A_k^j < A_\phi$ **then**
12: **if** $\mathfrak{B}_i^j$ doesn't exist **then**
13: Create a set $\mathfrak{B}_i^j$ and initialize $\mathfrak{B}_i^j \leftarrow \emptyset$
14: $\mathscr{P}_w \leftarrow \mathscr{P}_w \cup \{TP_i\}$
15: **end if**
16: $\mathfrak{B}_i^j \leftarrow \mathfrak{B}_i^j \cup \{TE_k^j\}$
17: $\mathscr{T} \leftarrow \mathscr{T} \cup \{(i,j,k)\}$
18: **end if**
19: **end for**
20: **end for**
21: Construct a set $\mathcal{B} = \{\mathfrak{B}_i^j \mid TP_i \in \mathscr{P}_w,\ TP_i^j \in \mathscr{M}'_q\}$
22: **return** $(\mathcal{B}, \mathscr{P}_w, A_\phi, \mathscr{T})$

threshold ask pays unit price as the critical value.

As long as the bidding and asking of the TPs and TEs are performed in the same currency, there is no need to worry about the outcome of the auction process. For instance, if the bidding and asking are performed in currency dollar ($) and providers' offer prices in the range of [0-1], whereas executors ask for prices in the range of [10-1000], then in that auction there will be no mapping between TPs and TEs because the TPs and TEs are not agreed upon. In this scenario, the TPs and TEs can revisit their bidding and asking strategy and can again participate in the next auction by updating their bid and ask values.

5.4.2 A Walk-Through Example

In order to illustrate the working procedure of TATA, we consider a bid matrix (see Table 5.5), a need matrix (see Table 5.6) and an ask matrix (see Table 5.4). The bid matrix in Table 5.5 consists of bids of 5 task providers for 7 *ROIs*. A zero bid value for a *ROI* indicates that the provider doesn't have a sensing task for that *ROI*. The need matrix in Table 5.6 contains the values that indicate the number of executors required for the providers at different *ROIs*.

Algorithm 4 TATA-STD($\mathcal{B}, \mathscr{T}, \mathbf{N}$)

Input: Set $\mathcal{B}$ that contains all $\mathfrak{B}_i^j$, Set of trades $\mathscr{T}$, Need matrix $\mathbf{N}$
Output: Set of winning task executors $\mathcal{E}_w$, Set of successful trades $\mathscr{S}^{\mathscr{T}}$.

$\mathcal{E}_w \leftarrow \emptyset$;
for $\mathfrak{B}_i^j \in \mathcal{B}$ **do**
  $s = |\mathfrak{B}_i^j|$
  if $s > N_i^j$ **then**
    sort all executors in $\mathfrak{B}_i^j$ to obtain a list $\mathfrak{B}_{ij}^{(s)} = \langle TE_{k_1}^{j_1}, TE_{k_2}^{j_2}, \ldots, TE_{k_s}^{j_s} \rangle$ such that $A_{k_1}^{j_1} \leq A_{k_2}^{j_2} \leq \ldots \leq A_{k_s}^{j_s}$
    for $r = N_i^j + 1$ to s **do**
      $\mathfrak{B}_i^j = \mathfrak{B}_i^j - \{TE_{k_r}^{j_r}\}$
      $\mathscr{T} = \mathscr{T} - \{(i, j_r, k_r)\}$
    end for
  end if
end for
$\mathscr{S}^{\mathscr{T}} = \mathscr{T}$
for each $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$ **do**
  if $TE_k \notin \mathcal{E}_w$ **then**
    $\mathcal{E}_w \leftarrow \mathcal{E}_w \cup \{TE_k\}$
  end if
end for
return $(\mathcal{E}_w, \mathscr{S}^{\mathscr{T}})$

The ask matrix in Table 5.4 consists of the ask values of 5 task executors for 7 *ROIs*. The following illustrates how TATA works for the auctioneer to generate the auction outcome.

- Create a new task providers' set from original set $\mathscr{P}$: $\mathscr{M} = \{TP_1^4,\ TP_1^7,\ TP_2^6,\ TP_3^2,\ TP_3^3,\ TP_4^1,\ TP_5^5,\ TP_5^6\}$ having non-zero bids. Here, TP_i^j represent the bid of task provider $TP_i \in \mathscr{P}$ for ROI $R_j \in \mathcal{R}$.

- Sort all the task providers of the set $\mathscr{M}$ in non-increasing order of their bids to construct an ordered list $\mathscr{M}' = \{TP_1^7,\ TP_1^4,\ TP_3^3,\ TP_5^5,\ TP_4^1,\ TP_3^2,\ TP_2^6,\ TP_5^6\}$. These are shown using TP_i^j with circular nodes in Figure 5.3.

- Create a new task executors' set from original set $\mathcal{E}$: $\mathscr{N} = \{TE_1^1,\ TE_1^5,\ TE_1^6,\ TE_2^1,\ TE_2^3,\ TE_2^4,\ TE_2^7,\ TE_3^3,\ TE_3^4,\ TE_3^6,\ TE_4^2,\ TE_4^5,\ TE_4^7,\ TE_5^2,\ TE_5^3,\ TE_5^4,\ TE_5^7\}$ having non-zero ask. Here, TE_k^j represents the ask of task executor $TE_k \in \mathcal{E}$ for $R_j \in \mathcal{R}$.

- Sort all the task executors of the set $\mathscr{N}$ in non-decreasing order of their asks to construct an ordered list $\mathscr{N}' = \langle TE_5^2,\ TE_2^4,\ TE_2^3,\ TE_2^7,\ TE_1^5,\ TE_3^3,\ TE_3^4,\ TE_1^1,\ TE_5^3,\ TE_4^7,\ TE_2^1,\ TE_5^4,\ TE_3^6,\ TE_4^5,\ TE_4^2,\ TE_1^6,\ TE_5^7 \rangle$. These are shown using TE_k^j with circular nodes in Figure 5.3.

- Based on $\mathscr{M}'$ and $\mathscr{N}'$, a bipartite graph is created as shown in Figure 5.3.

- Determine the threshold ask:$\phi = \lceil (|\mathscr{N}'| + 1)/2 \rceil = \lceil (17+1)/2 \rceil = 9$, $A_\phi = A_9 = A_{l_9}^{m_9} = 7$

Algorithm 5 TATA-PC($\mathscr{P}_w, \mathcal{E}_w, \mathscr{S}^{\mathscr{T}}, A_\phi$)

Input: Set of winning task providers $\mathscr{P}_w$, Set of winning task executors $\mathcal{E}_w$, Set of successful trades $\mathscr{S}^{\mathscr{T}}$, Threshold ask A_ϕ

Output: Set of payments of all winning executors $P_w^{(e)}$, Set of charges for all winning providers $P_w^{(p)}$.

$P_w^{(p)} \leftarrow \emptyset, P_w^{(e)} \leftarrow \emptyset$
for $TP_i \in \mathscr{P}_w$ **do**
  $c_i = 0$
  for each $(x,y,z) \in \mathscr{S}^{\mathscr{T}}$ **do**
    if $i == x$ **then**
      $c_i = c_i + 1$ //Count number of successful trades for $TP_i \in \mathscr{P}_w$
    end if
  end for
  $P_i^{(p)} = c_i \times A_\phi$ //Total price charged to $TP_i \in \mathscr{P}_w$
  $P_w^{(p)} \leftarrow P_w^{(p)} \cup P_i^{(p)}$
end for
for $TE_k \in \mathcal{E}_w$ **do**
  $c_k = 0$
  for each $(x,y,z) \in \mathscr{S}^{\mathscr{T}}$ **do**
    if $k == z$ **then**
      $c_k = c_k + 1$ //Count number of successful trades for $TE_i \in \mathcal{E}_w$
    end if
  end for
  $P_k^{(e)} = c_k \times A_\phi$ //Total payment rewarded to $TE_i \in \mathcal{E}_w$
  $P_w^{(e)} \leftarrow P_w^{(e)} \cup P_k^{(e)}$
end for
return $(P_w^{(p)}, P_w^{(e)})$

- The set $\mathscr{M}'_q = \{TP_1^7,\ TP_1^4,\ TP_3^3,\ TP_5^5\}$, since $D_5^5 \geq 7$ and $D_4^1 < 7$ for the ordered sequence in $\mathscr{M}'$.
- $\mathscr{P}_w = \{TP_1,\ TP_3,\ TP_5\}$, $\mathfrak{B}_1^7 = \{TE_2^7\}$, $\mathfrak{B}_1^4 = \{TE_2^4,\ TE_3^4\}$, $\mathfrak{B}_3^3 = \{TE_2^3,\ TE_3^3\}$, $\mathfrak{B}_5^5 = \{TE_1^5\}$, $\mathscr{T} = \{(1,7,2),\ (1,4,2),\ (1,4,3),\ (3,3,2),\ (3,3,3),\ (5,5,1)\}$, $\mathcal{B} = \{\mathfrak{B}_1^7,\ \mathfrak{B}_1^4,\ \mathfrak{B}_3^3,\ \mathfrak{B}_5^5\}$. The Figure 5.4 shows a bipartite graph (which is essentially a sub-graph of the bipartite graph in Figure 5.3) using these selected task providers, executors and the trades involving them.
- Since, $|\mathfrak{B}_3^3| > N_3^3$, it is required to eliminate $|\mathfrak{B}_3^3| - N_3^3 = 2 - 1 = 1$ executor from $\mathfrak{B}_3^3$. After elimination $\mathfrak{B}_3^3 = \{TE_2^3,\ TE_3^3\} - \{TE_3^3\} = \{TE_2^3\}$. Here, TE_3^3 is eliminated as it has less utility as compared to TE_2^3. $\mathscr{S}^{\mathscr{T}} = \{(1,7,2),\ (1,4,2),\ (1,4,3),\ (3,3,2),\ (5,5,1)\}$. $\mathcal{E}_w = \{TE_1,\ TE_2,\ TE_3\}$
- Price charged to task providers:$P_1^{(p)} = 3 \times A_\phi = 3 \times 7 = 21$, $P_3^{(p)} = 1 \times 7 = 7$, $P_5^{(p)} = 1 \times 7 = 7$. Payment rewarded to task executors: $P_1^{(e)} = 1 \times 7 = 7$, $P_2^{(e)} = 3 \times 7 = 21$, $P_3^{(e)} = 1 \times 7 = 7$. $P_w^{(p)} = \{P_1^{(p)},\ P_3^{(p)}, P_5^{(p)}\}$, $P_w^{(e)} = \{P_1^{(e)},\ P_2^{(e)}, P_3^{(e)}\}$.

A pictorial summary of the overall working process of TATA with respect to the above example is shown in Fig. 5.5.

TABLE 5.4: Ask matrix of 5 task Executors

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
TE_1	6	0	0	0	5	10	0
TE_2	8	0	4	3	0	0	4
TE_3	0	0	6	6	0	9	0
TE_4	0	10	0	0	9	0	8
TE_5	0	2	7	9	0	0	11

TABLE 5.5: Bid matrix of 5 Task Providers

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
TP_1	0	0	0	9	0	0	10
TP_2	0	0	0	0	0	4	0
TP_3	0	5	8	0	0	0	0
TP_4	6	0	0	0	0	0	0
TP_5	0	0	0	0	7	3	0

TABLE 5.6: Need matrix of 5 Task Providers

	R_1	R_2	R_3	R_4	R_5	R_6	R_7
TP_1	0	0	0	2	0	0	2
TP_2	0	0	0	0	0	3	0
TP_3	0	1	1	0	0	0	0
TP_4	1	0	0	0	0	0	0
TP_5	0	0	0	0	3	3	0

5.4.3 Proofs of Desirable Properties

This section first describes four theoretical properties [15, 212] of an auction based incentive mechanism namely budget balance, individual rationality, truthfulness/incentive compatibility, and computational efficiency. After that those properties are analyzed with respect to the proposed mechanism TATA.

- *Budget Balance*: An auction based incentive mechanism is called budget balanced if the auctioneer doesn't have any deficit. In other words, the total price charged to all the winning task providers must be greater or equal to the total payment given to all the task executors. Mathematically, following condition should be satisfied if an incentive mechanism holds budget balance property.

$$\sum_{TP_i \in \mathscr{P}_w} P_i^{(p)} \geq \sum_{TE_k \in \mathcal{E}_w} P_k^{(e)}$$

- *Individual Rationality*: An auction based incentive mechanism will be individually rational if no winning task provider is charged more than its bid and no winning task

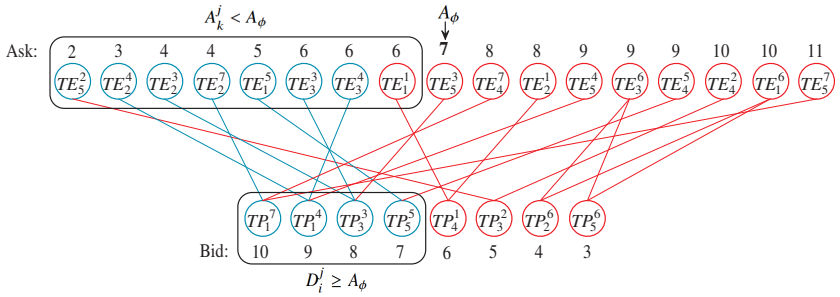


FIGURE 5.3: Bipartite graph between the set of task executors and the set of task providers. TPs are arranged in non-increasing order of the submitted bids and TEs are arranged in non-decreasing order of the submitted asks. The asks less than the threshold asks (i.e. $A_k^j < A_\phi$) and the bids greater or equal to the threshold asks (i.e. $D_i^j \geq A_\phi$) are shown within rounded rectangles. The blue and the red edges represent the candidate trades ($\mathscr{T}$) and non-candidate trades respectively.

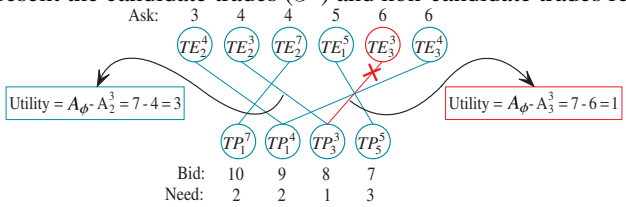


FIGURE 5.4: Bipartite graph between the candidates (executors and providers) selected by Algorithm 2. The blue and the red edges represent the successful trades ($\mathscr{S}^{\mathscr{T}}$) and unsuccessful trades respectively.

executor is offered a payment less than its ask,i.e., $\forall(i,j,k) \in \mathscr{S}^{\mathscr{T}}$,we have $P_{ij}^{(p)} \leq D_i^j$ and $\forall(i,j,k) \in \mathscr{S}^{\mathscr{T}}$,we have $P_{kj}^{(e)} \geq A_k^j$. Note that "individual rationality" is not an assumption in this research. Instead, "individual rationality" is a desirable property that our designed auction mechanism (i.e., TATA) holds. TATA enforces individual rationality to guarantee that no winning TP is charged greater than its bid and that no winning TE is offered a payment lesser than its ask. TATA satisfies "individual rationality" and the proof of the corresponding lemma (i.e., lemma 1) is presented in a subsequent section. The proof does not set any assumption on the bids of TPs or the asks of TEs. Thus, the conclusions are valid for a practical system with any possible data sets of the bids and asks. Additionally, we also provide experimental results in Section 5.5.2 for validating the "individual rationality" property.

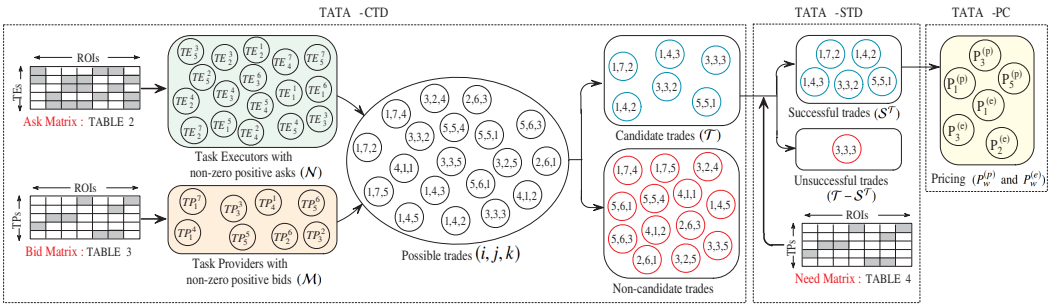


FIGURE 5.5: A pictorial summary of the overall working process of TATA for the walk-through example of Section 5.4.2. TP_i^j and TE_k^j indicate non-zero bids and asks respectively. From the possible trades, TATA-CTD first identifies a set of candidate trades. Next, TATA-STD finds a set of successful trades from the set of candidate trades. Finally, TATA-PC calculates the price charged to the winning TPs and payment/ incentive/ reward given to the winning TEs.

- *Incentive compatibility /Truthfulness*: If users (i.e., TPs and TEs) are irrational, then they may play untruthfully in the auction game. In other words, for irrational users, untruthful bidding or asking may take place. In order to counter this problem, we have designed TATA in such a way that it can satisfy the truthfulness/incentive compatible property. TATA enforces "truthfulness" property in order to ensure that no TP can increase its utility by placing a bid other than its actual valuation, and no TE can increase its utility by placing an ask other than its actual cost. In order to better understand the property of incentive compatibility or truthfulness we must first know the concept of *dominant strategy*.

  Definition 3 *A strategy S_l is the player (task executor or task provider) l's dominant strategy, if for any strategy (i.e. asking or bidding) $S'_l \neq S_l$ and rest of the players' strategy profile S_{-l}, the utilities satisfy the following condition:*

  $$U_l(S_l, S_{-l}) \geq U_l(S'_l, S_{-l})$$

  The definition of dominant strategy is the foundation of truthfulness or incentive compatibility property. It implies that there is no motive for any player (task executors and task providers) to lie about their private information. Thus disclosing truthful information becomes the dominant strategy for each player. Therefore, a player cannot improve its utility by bidding or asking untruthfully. More specifically, in the context of our proposed auction based incentive mechanism, a task provider $TP_i \in \mathscr{P}$ achieves maximum utility by bidding $\mathbf{D_i} = \mathbf{V_i}$. Similarly, a task executor $TE_k \in \mathcal{E}$ achieves maximum utility by asking $\mathbf{A_k} = \mathbf{C_k}$.

- *Computational Efficiency*: An auction based incentive mechanism expected to have polynomial time complexity.

Lemma 1 *TATA is individually rational.*

Proof. The proposed double auction based task allocation mechanism TATA will be individually rational, if the following two conditions are met, (i) For a successful trade $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$ the task provider is not charged more than its bid. It means the condition $P_{ij}^{(p)} \leq D_i^j$ must hold. (ii) Similarly, for a successful trade $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$, the task executor is not paid less than its ask. It means the condition $P_{kj}^{(e)} \geq A_k^j$ must hold. Now, in TATA, for each $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$, the price charged to the winning task provider $TP_i \in \mathscr{P}_w \subseteq \mathscr{P}$ is given as $P_{ij}^{(p)} = A_\phi \leq D_i^j$. Hence, the first condition is satisfied. On the other hand, for each $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$, the payment rewarded to the winning task executor $TE_k \in \mathcal{E}_w \subset \mathcal{E}$ is given as $P_{kj}^{(e)} = A_\phi > A_k^j$. Since both conditions are met, the proposed double auction TATA is individually rational.

Lemma 2 *The proposed auction TATA is budget balanced.*

Proof. An auction mechanism will be budget balanced if the auctioneer should not have any deficit. In TATA, for each trade $(i, j, k) \in \mathscr{S}^{\mathscr{T}}$, *the price charged to task provider* $TP_i \in \mathscr{P}_w =$ *the payment rewarded to the task executor* $TE_k \in \mathcal{E}_w = A_\phi$. Eventually, *the total price*

charged to the winning task providers $= \sum_{(i,j,k)\in\mathscr{S}^{\mathscr{T}}} P_{ij}^{(p)} = |\mathscr{S}^{\mathscr{T}}| \times A_\phi$. On the other hand, *the total payments rewarded to the winning task executors* $= \sum_{(i,j,k)\in\mathscr{S}^{\mathscr{T}}} P_{kj}^{(e)} = |\mathscr{S}^{\mathscr{T}}| \times A_\phi$. Since the total price charged to the winning task providers is same as the total payments rewarded to the task executors, we can say that the auctioneer does not have any deficit, which completes the proof.

Lemma 3 *The proposed auction TATA is incentive compatible /truthful for task providers.*

Proof. We use tilde in the notations for the outcomes with untruthful bidding by the task providers $TP_i \in \mathscr{P}$. In order to prove this lemma, it is required to show that no $TP_i \in \mathscr{P}$ can improve its utility by placing untruthful bids (i.e. $\widetilde{D}_i^j \neq V_i^j$) when other users' bids and asks are unchanged. Here we examine two possible cases.
Case 1: *If* $TP_i \in \mathscr{P}$ *wins in the auction for bid* $D_i^j \in \mathbf{D_i}$.
Here, two sub cases may occur.

- *If* $TP_i \in \mathscr{P}$ *places a bid* D_i^j *which is a truthful bid.*
  Since $D_i^j \in \mathbf{D_i}$ is a truthful bid, we can write $D_i^j = V_i^j$. Also, it is given that $TP_i \in \mathscr{P}$ wins for $D_i^j \in \mathbf{D_i}$. It means, $D_i^j \geq A_\phi$. From $D_i^j = V_i^j$ and $D_i^j \geq A_\phi$, we get $V_i^j \geq A_\phi$. Hence, for truthful bid $D_i^j \in \mathbf{D_i}$, the utility achieved by $TP_i \in \mathscr{P}$ can be given as follows:

$$\mathfrak{U}_{ij}^{(p)} = |\mathfrak{B}_i^j| \times (V_i^j - A_\phi) \geq 0 \tag{5.7}$$

- *If* $TP_i \in \mathscr{P}$ *places a bid* $\widetilde{D}_i^j$ *which is an untruthful bid:*
  Since $\widetilde{D}_i^j$ is an untruthful bid, we can write $\widetilde{D}_i^j \neq V_i^j$. Now, in case of the untruthful bid $\widetilde{D}_i^j$, two scenarios may occur:
  - *If* $V_i^j \geq A_\phi$: Since $TP_i \in \mathscr{P}$ wins in the auction by placing untruthful bid $\widetilde{D}_i^j$, we can say $\widetilde{\mathfrak{B}}_i^j = \mathfrak{B}_i^j$. Hence, the utility due to untruthful bidding is

$$\widetilde{\mathfrak{U}}_{ij}^{(p)} = |\widetilde{\mathfrak{B}}_i^j| \times (V_i^j - A_\phi) = |\mathfrak{B}_i^j| \times (V_i^j - A_\phi) \geq 0 \tag{5.8}$$

  - *If* $V_i^j < A_\phi$: Also, in this scenario, $TP_i \in \mathscr{P}$ wins in the auction by placing untruthful bid $\widetilde{D}_i^j$. Hence, $\widetilde{\mathfrak{B}}_i^j = \mathfrak{B}_i^j$. But, because of the condition $V_i^j < A_\phi$, the utility due to untruthful bidding becomes negative as given below:

$$\widetilde{\mathfrak{U}}_{ij}^{(p)} = |\widetilde{\mathfrak{B}}_i^j| \times (V_i^j - A_\phi) = |\mathfrak{B}_i^j| \times (V_i^j - A_\phi) < 0 \tag{5.9}$$

Case 2: *If* $TP_i \in \mathscr{P}$ *loses in the auction for bid* $D_i^j \in \mathbf{D_i}$.
In this case, $TP_i \in \mathscr{P}$ loses in the auction for both truthful bid (i.e. D_i^j) and untruthful bid (i.e. $\widetilde{D}_i^j$). Hence, for both truthful and untruthful bidding the utility will be zero, i.e.

$$\mathfrak{U}_{ij}^{(p)} = \widetilde{\mathfrak{U}}_{ij}^{(p)} = 0 \tag{5.10}$$

From Eq. 5.7, Eq. 5.8, Eq. 5.9 and Eq. 5.10, we can conclude that $\mathfrak{U}_{ij}^{(p)} \geq \widetilde{\mathfrak{U}}_{ij}^{(p)}$, i.e. the task providers cannot improve their utility by bidding untruthfully. Hence, TATA is truthful/ incentive compatible for task providers.

Lemma 4 *The proposed auction TATA is incentive compatible /truthful for task executors.*

Proof. We use tilde in the notations for the outcome of the untruthful asking by the task executors $TP_i \in \mathscr{P}$. In order to prove this lemma, it is required to show that no $TE_k \in \mathcal{E}$ can improve its utility by placing untruthful asks (i.e. $\widetilde{A}_k^j \neq C_k^j$) when other users' bids and asks are unchanged. Here we examine two possible cases

Case 1: *If $TE_k \in \mathcal{E}$ wins for the ask $A_k^j \in \mathbf{A_k}$:*

Here, two sub-cases may occur:

- *If the ask $A_k^j \in \mathbf{A_k}$ is a truthful ask:*
  Since $A_k^j \in \mathbf{A_k}$ is a truthful ask we can write $A_k^j = C_k^j$. Also, in this case, it is given that $TE_k \in \mathcal{E}$ wins for the ask $A_k^j \in \mathbf{A_k}$. It means $A_k^j < A_\phi$. From $A_k^j = C_k^j$ and $A_k^j < A_\phi$, we get $C_k^j < A_\phi$. Hence, for truthful ask $A_k^j \in \mathbf{A_k}$ the utility achieved by $TE_k \in \mathcal{E}$ can be given as follows:

$$\mathcal{U}_{kj}^{(e)} = A_\phi - C_k^j > 0 \tag{5.11}$$

- *If $TE_k \in \mathcal{E}$ places an ask $\widetilde{A}_k^j$ which is an untruthful ask:*
  Since $\widetilde{A}_k^j$ is an untruthful ask, we can write $\widetilde{A}_k^j \neq C_k^j$. Now, two possible scenarios may occur
  - *If $C_k^j < A_\phi$*: Task executor $TE_k \in \mathcal{E}$ wins in the auction by placing untruthful ask $\widetilde{A}_k^j$. Here, utility due to untruthful bidding is:

$$\widetilde{\mathcal{U}}_{kj}^{(e)} = A_\phi - C_k^j > 0 \tag{5.12}$$

  - *If $C_k^j \geq A_\phi$*: Also, in this scenario, $TE_i \in \mathcal{E}$ wins in the auction by placing untruthful ask $\widetilde{A}_i^j$. But, because of the condition $C_k^j \geq A_\phi$, the utility due to untruthful asking may become negative as given below:

$$\widetilde{\mathcal{U}}_{kj}^{(e)} = A_\phi - C_k^j \leq 0 \tag{5.13}$$

Case 2: *If $TE_k \in \mathcal{E}$ loses in the auction for the ask value $A_k^j \in \mathbf{A_i}$.*

In this case, $TE_k \in \mathcal{E}$ loses in the auction for both truthful ask (i.e. A_k^j) and untruthful ask (i.e. $\widetilde{A}_k^j$). Hence, for both truthful and untruthful asking the utility will be zero, i.e.

$$\mathcal{U}_{kj}^{(e)} = \widetilde{\mathcal{U}}_{kj}^{(e)} = 0 \tag{5.14}$$

From Eq. 5.11, Eq. 5.12, Eq. 5.13 and Eq. 5.14, it is clear that $\mathcal{U}_{kj}^{(e)} \geq \widetilde{\mathcal{U}}_{kj}^{(e)}$, i.e. the task executors cannot improve their utility by asking untruthfully. Hence, TATA is truthful/ incentive compatible for task executors.

Theorem 1 *The proposed auction TATA is truthful or incentive compatible.*

Proof. *Lemma 3* shows that TATA is truthful for task providers. *Lemma 4* shows that TATA is truthful for task executors. *Lemma 3* and *Lemma 4* together prove that TATA is truthful or incentive compatible.

Lemma 5 *The proposed auction TATA is computationally efficient.*

Proof. In the candidate trade determination stage (i.e. Algorithm 2), the maximum possible task providers in set $\mathscr{M}$ is mn, where $n = |\mathscr{P}|$ and $m = |\mathcal{R}|$. The sorting in line 3 takes $O(mn\ log\ mn)$ time in the worst case. Now, the maximum number of possible task executors in set $\mathscr{N}$ is mt and eventually the sorting in line 5 takes $O(mt\ log\ mt)$ time in the worse case, where $t = |\mathcal{E}|$. There can be maximum of mn elements in set $\mathscr{M}'_q$. Hence, the for loop in line 9 iterates over maximum of mn times. On the other hand the for loop in line 10 iterates over maximum of mt times. Note that, in this study, the total number of *ROIs* (i.e. the value of m) is assumed to be constant. Hence, in worst case the Algorithm 2 takes $O(mn\ log\ mn)\ +\ O(mt\ log\ mt)\ +\ O(mn.mt) \approx O(n\ log\ n\ +\ t\ log\ t\ +\ nt)$. Now, in the successful trade determination stage (i.e. Algorithm 3), maximum cardinality of set $\mathcal{B}$ can be n. So, the loop in line 2 can iterate maximum of n times. Now, the sorting in line 5 takes $O(mt\ log\ mt)$ times. The for loop in line 6 can iterate maximum of mt times. Also, the for loop in line 13 can iterate over maximum of $mn(\lceil (mt+1)/2 \rceil - 1)$ times. Hence, in worst case Algorithm 3 takes $O(n(mt\ log\ mt\ +\ mt)) + O(mn(\lceil (mt+1)/2 \rceil - 1)) \approx O(nt\ log\ t\ +\ nt) \approx O(nt\ log\ t)$. Now, in the price calculation stage (i.e. Algorithm 4), the for loop in line 2 can iterate over at most mn times. The for loop in line 12 can iterate over maximum of $(\lceil (mt+1)/2) \rceil - 1)$ times. Both the for loops in line 4 and line 14 can iterate over maximum of $mn(\lceil (mt+1)/2 \rceil - 1)$ times. Hence, in worst case, Algorithm 4 takes $O(mn.mn(\lceil (mt+1)/2 \rceil - 1)\ +\ mn.(\lceil (mt+1)/2 \rceil - 1)^2) \approx O(n^2t\ +\ nt^2)$. Hence, overall time complexity of TATA is $O(n\ log\ n\ +\ t\ log\ t\ +\ nt)\ +\ O(nt\ log\ t)\ +\ O(n^2t\ +\ nt^2) \approx O(n^2t\ +\ nt^2)$.

Lemma 6 *TATA is incentive compatible, individually rational, computationally efficient, and budget balanced.*

Proof. This conclusion can be drawn from Lemma 1-5.

Lemma 7 *The maximum number of successful trades in TATA can be* $mn(\lceil \frac{mt+1}{2} \rceil - 1)$ *where,* $n = |\mathscr{P}|$, $m = |\mathcal{R}|$ *and* $t = |\mathcal{E}|$.

Proof. Let us consider, for each $R_j \in \mathcal{R}$, all the $TP_i \in \mathscr{P}$ and $TE_k \in \mathcal{E}$ place non-zero positive bids and asks respectively. In this case, the total number of non-zero positive asks (i.e. maximum $|\mathscr{N}|$ value) $= mt$. Similarly, the total number of non-zero positive bids (i.e. maximum $|\mathscr{M}|$ value) $= mn$. Hence, the maximum number of elements in $\mathscr{N}'$ is $= mt$. The median ask A_ϕ occurs for the task executor who is at position $\phi = \lceil \frac{mt+1}{2} \rceil$ of the list $\mathscr{N}'$. Clearly, the maximum number of elements in M'_q is $= mn$ that occurs when $\forall_{i,j}\ D_i^j \leq A_\phi$ and $|\mathscr{M}| = mn$. Now, as given in line 9-19 of Algorithm 2, for each $TP_i^j \in \mathscr{M}'_q$ if $TE_k^j \in \mathscr{N}$ satisfies the condition $A_k^j < A_\phi$ then there will be a candidate trade (i, j, k). Now, the maximum number of $TE_k^j \in \mathscr{N}$ that satisfies $A_k^j < A_\phi$ is $\phi - 1 = \lceil \frac{mt+1}{2} \rceil - 1$. Hence, the maximum number of candidate trades in $\mathscr{T}$ is $= |\mathscr{M}'_q| \times (\phi - 1) = mn \times (\lceil \frac{mt+1}{2} \rceil - 1)$. Now, as discussed in Section 5.4.1, the TATA-STD() stage may eliminate some of the trades in $\mathscr{T}$ to produce the set of successful trades ($\mathscr{S}^T$) . In TATA-STD() stage, if the condition $s > N_i^j$ (line 4, Algorithm 3) is not satisfied for $\mathfrak{B}_i^j \in \mathcal{B}$, no trades will be eliminated from $\mathscr{T}$. Hence, the maximum number of possible successful trades $= mn \times (\lceil \frac{mt+1}{2} \rceil - 1)$.

Lemma 8 *In case of truthful bidding, the maximum and the minimum utility achieved by $TP_i \in \mathscr{P}$ for $R_j \in \mathcal{R}$ are $N_i^j \times (D_i^j - A_\phi)$ and 0 respectively.*

Proof. $TP_i \in \mathscr{P}$ places truthful bid D_i^j for $R_j \in \mathcal{R}$. Since, D_i^j is a truthful bid, the condition $V_i^j = D_i^j > 0$ will be satisfied. Now, if $TP_i \in \mathscr{P}$ wins in the auction for bid D_i^j, then the utility achieved by TP_i for R_j is given as:

$$\mathcal{U}_{ij}^{(p)} = |\mathfrak{B}_{\mathfrak{i}}^{\mathfrak{j}}| \times (V_i^j - A_\phi) = |\mathfrak{B}_{\mathfrak{i}}^{\mathfrak{j}}| \times (D_i^j - A_\phi) \tag{5.15}$$

where, $\mathfrak{B}_{\mathfrak{i}}^{\mathfrak{j}}$ contains the executors $TE_k^j \in \mathscr{N}$ that are won by TP_i for R_j. Now, after TATA-STD() stage (i.e. Algorithm 3), we can say, $1 \leq |\mathfrak{B}_{\mathfrak{i}}^{\mathfrak{j}}| \leq N_i^j$. Hence, the maximum utility achieved by TP_i for R_j is found by putting $|\mathfrak{B}_{\mathfrak{i}}^{\mathfrak{j}}| = N_i^j$ in Eq. 5.15 as follows:

$$\mathcal{U}_{ij}^{(p)} = N_i^j \times (D_i^j - A_\phi) \tag{5.16}$$

On the other hand, if $TP_i \in \mathscr{P}$ loses in the auction for bid D_i^j, then the utility is: $\mathcal{U}_{ij}^{(p)} = 0$. Hence, the maximum and the minimum utility achieved by TP_i for R_j are $N_i^j \times (D_i^j - A_\phi)$ and 0 respectively.

Lemma 9 *In case of truthful asking, the maximum and the minimum utility achieved by $TE_k \in \mathcal{E}$ for $R_j \in \mathcal{R}$ are $(A_\phi - A_k^j)$ and 0 respectively.*

Proof. $TE_k \in \mathcal{E}$ places truthful ask A_k^j for $R_j \in \mathcal{R}$. Since, A_k^j is a truthful ask, $A_k^j = C_k^j > 0$ will be satisfied. Now, if $TE_k \in \mathcal{E}$ wins in the auction for ask A_k^j, then the utility achieved by TE_k for R_j is given as follows: $\mathcal{U}_{kj}^{(e)} = A_\phi - C_k^j = A_\phi - A_k^j$. On the other hand, if $TE_k \in \mathcal{E}$ loses in the auction for ask A_k^j, then the utility is: $\mathcal{U}_{kj}^{(e)} = 0$. Hence, the maximum and the minimum utility achieved by TE_k for R_j are $A_\phi - A_k^j$ and 0 respectively.

Lemma 10 *The expected number of successful trades after k rounds (i.e. if TATA is repeated k times) will be $\approx \frac{nt}{4} \times [\sum_{i=0}^{m} i f_i]^2$, where $n = |\mathscr{P}|$, $m = |\mathcal{R}|$, $t = |\mathcal{E}|$, and f_i denotes the probability of having i number of non-zero bids/asks for any TP/TE.*

Proof. Let us consider, for each $R_j \in \mathcal{R}$, the probability of placing a bid/ask (non-zero) is $p \in (0,1)$. Hence, for any TP/TE, the probability of having i number of non-zero bids/asks in $m = |\mathcal{R}|$ regions is $f_i = \binom{m}{i} p^i (1-p)^{m-i}$. Suppose, T_b and T_a are the variables that denote the total number of non-zero positive bids and asks respectively. Then the expected value of T_b and T_a are $E(T_b) = n \times \sum_{i=0}^{m} i f_i$ and $E(T_a) = t \times \sum_{i=0}^{m} i f_i$ respectively. In this case, the value of ϕ will be $\lceil \frac{E(T_a)+1}{2} \rceil$. If the asks and the bids are drawn from the same non-skewed (symmetric) distribution, then the number of elements in M_q' will be approximately half of the expected value of T_b, i.e., $|M_q'| \approx \frac{E(T_b)}{2}$. Now, if no trades are eliminated, then the possible number of successful trades will be $\approx \frac{E(T_b)}{2} \times (\phi - 1) \approx \frac{E(T_b)}{2} \times (\lceil \frac{E(T_a)+1}{2} \rceil - 1)$ $\approx \frac{1}{4} \times E(T_a) E(T_b) \approx \frac{nt}{4} \times \sum_{i=0}^{m} i f_i \times \sum_{i=0}^{m} i f_i \approx \frac{nt}{4} \times [\sum_{i=0}^{m} i f_i]^2$ Now, note that if we repeat the auction several times, each of the rounds will be independent. Hence, the average number of successful trades in k rounds will be $\approx \frac{nt}{4} \times [\sum_{i=0}^{m} i f_i]^2$.

5.5 Evaluation Results

The proposed double auction based incentive mechanism TATA is implemented and a set of simulation experiments are performed to investigate its effectiveness. In this section, the experimental setup is first discussed, next different properties of TATA are validated through simulations, and finally the performance of TATA is compared with some of the existing incentive mechanisms.

5.5.1 Experimental Setup

For experiment, the real-world payments to the crowdworkers in Amazon Mechanical Turk (MTurk) [213] are considered as the bids of TPs. A maximum of 200 TPs are considered and their bids are normalized between the interval $D_i^j \in (0,1]$. Since there are no existing statistics on task executors' ask values, we randomly generate the asks of the task executors that are uniformly distributed over $A_k^j \in (0,1]$. Here, a maximum of 1000 task executors' ask values are generated. On the other hand, the need values of TPs are randomly generated from the uniform distribution within $N_i^j \in (1, N_{max}]$. In TATA, N_{max} influences the winner satisfaction level (more details in Section 5.5.3) of task providers. All the asks and bids are generated by considering a total of 100 ROIs. Note that the unavailability of problem-specific open source real-world data is not unexpected as it is an emerging field of research. The proposed incentive mechanism TATA is implemented using Python 3.6 (`https://docs.python.org/3.6/`). The following three Python libraries are mainly utilized to implement TATA: (i) Pandas (version 1.0.2) (ii) NumPy (version 1.13.3) (iii) SciPy (version 0.19.1). The experiments are carried out on a 64-bit Windows PC with 3.30 GHz Intel(R) Xeon(R) CPU E3-1226 v3 processor and 128 GB of RAM. Experiments are conducted to validate the four desirable properties: *truthfulness, individual rationality, budget balance* and *computational efficiency*. The impact of N_{max} on the number of successful trades are explored. Also, TATA is compared with two existing incentive mechanisms- IMRAL [71] and LBS [63] in terms of some performance metrics (see Section 5.5.3). A theoretical comparison among TATA, IMRAL, and LBS is also provided.

5.5.2 Analysis of Results

In this section, the experimental results are provided to validate the properties of TATA that are already proved in Section 5.4.3.

Truthfulness

To verify incentive compatibility/ truthfulness of the proposed incentive mechanism TATA, 9 TPs and 9 TEs are randomly selected from the total sets of 200 TPs and 1000 TEs respectively. The asks and the bids of these selected TEs and TPs are examined to see how their utilities change with the change of their asks and bids. The truthfulness property is verified under the possible cases such as when a task executor/provider truthfully wins or truthfully loses in the auction. Fig. 5.6(a) shows the case when the TEs truthfully win in the auction. For example, consider TE_{922} of Fig. 5.6(a). Here, TE_{922} achieves a utility of 0.15 when it

truthfully places the ask value (the truthful ask value was 0.18). It can be observed from the Fig. 5.6(a) that TE_{922} unable to improve its utility by manipulating (i.e. changing) its ask value. The Fig. 5.6(a) also shows a similar trend for rest of the TEs. Now, Fig. 5.6(b) shows the case when the TEs truthfully lose in the auction. For example, consider TE_{816} of Fig. 5.6(b). Here, TE_{816} loses in the auction and have a utility of 0 when it truthfully places the ask value (in this case the truthful ask value was 0.67). Further, it can be seen that, TE_{816} unable to improve its utility by manipulating its ask value. Here, for the asks other than the true value, the utility of the executor is either 0 or negative. In Fig. 5.6(b), a similar trend is found for rest of the TEs.

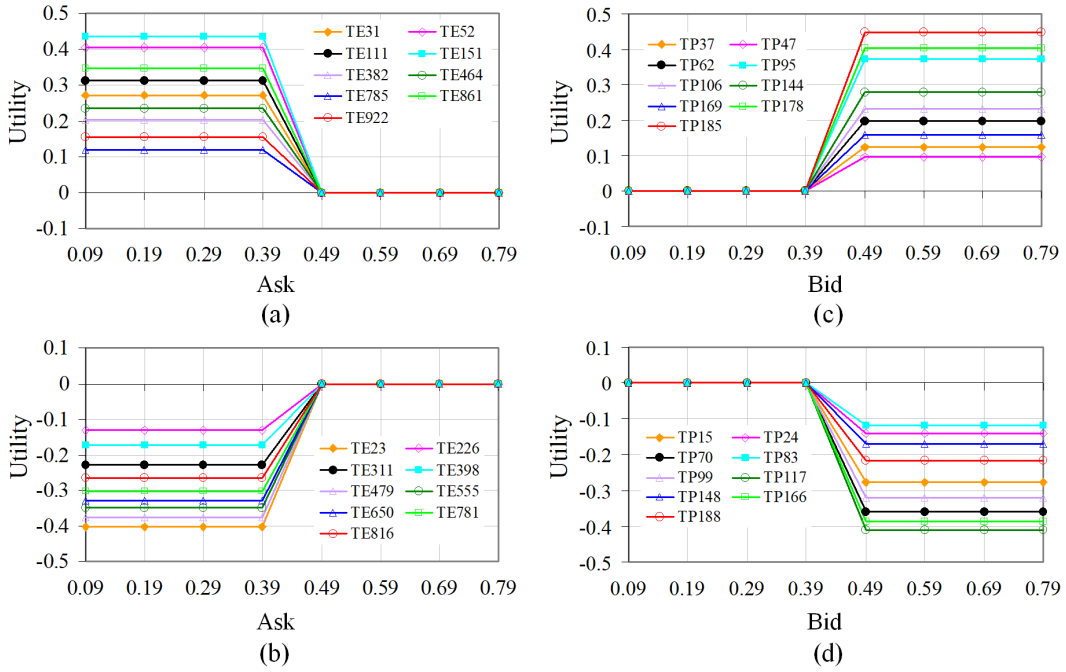


FIGURE 5.6: Verification of truthfulness property when (a) task executors win task providers by placing truthful asks (b) task executors lose in the auction by placing truthful asks (c) task providers win task executors by placing truthful bids (d) task providers lose in the auction by placing truthful bids

Fig. 5.6(c) shows the case when the TPs truthfully win in the auction. It is observed that none of the TPs are able to improve their utility by manipulating their bids. For example, as shown in Fig. 5.6(c), the utility of TP_{62} is 0.2 when bidding truthfully (in this case true bid was 0.65). It is clear that TP_{62} is not able to improve its utility by placing untruthful bids. The last case is shown in Fig. 5.6(d), where TPs lose in the auction by bidding truthfully and gain a utility of 0. It is observed from the Fig. 5.6(d) that, the TPs are unable to achieve more utility by bidding untruthfully. From the above results, it is verified that TATA holds truthfulness property for both task executors and task providers.

Individual rationality

To verify the individual rationality property (Lemma 1) of TATA, for different threshold settings, some randomly selected bids and the price charged to the task providers $TP_i \in \mathscr{P}_w \subseteq \mathscr{P}$ are presented in Fig. 5.7(a, c, e), whereas some randomly selected asks and the

payments to the winning $TE_k \in \mathcal{E}_w \subseteq \mathcal{E}$ are presented in Fig. 5.7(b, d, f). From Fig. 5.7(a, c, e), it is clear that for each $TP_i \in \mathscr{P}_w$ the charged price is always less than TP_i's bid value. Additionally, from Fig. 5.7(b, d, f), it is clear that for each $TE_k \in \mathcal{E}_w$ the payment received by TE_k is always more than TE_k's ask values. Hence, it is verified that TATA is individually rational.

Three different threshold settings in this experiment are as follows. (i) The actual threshold $A_\phi = 0.5097$, where $\phi = \lceil (r+1) \times \frac{1}{2} \rceil$ and r denotes total number of non-zero positive asks. (ii) A threshold value smaller than the actual threshold, i.e., $A'_\phi = 0.342$, where $\phi = \lceil (r+1) \times \frac{1}{3} \rceil$ (iii) A threshold value larger than the actual threshold, i.e., $A''_\phi = 0.651$, where $\phi = \lceil (r+1) \times \frac{2}{3} \rceil$. If the threshold value becomes smaller (i.e., $A'_\phi = 0.342$) than the actual threshold A_ϕ, then the number of wining TEs decreases. On the other hand, if the threshold value becomes larger ($A''_\phi = 0.651$) than the actual threshold A_ϕ, then the number of TPs decreases. For both the above-mentioned cases, the number of trades will be decreased. Due to this reason, as shown in Fig. 7(c-f) there is an overall decrease in the winning bid/ask values compared to Fig. 7(a,b). Some other interesting observations can also be made from the results. When the threshold value becomes smaller, less number of TEs wins in the auction, which can be verified from Fig. 7(d). On the other hand, when the threshold value becomes larger, a relatively less number of TPs wins in the auction, which can be verified from Fig. 7(e).

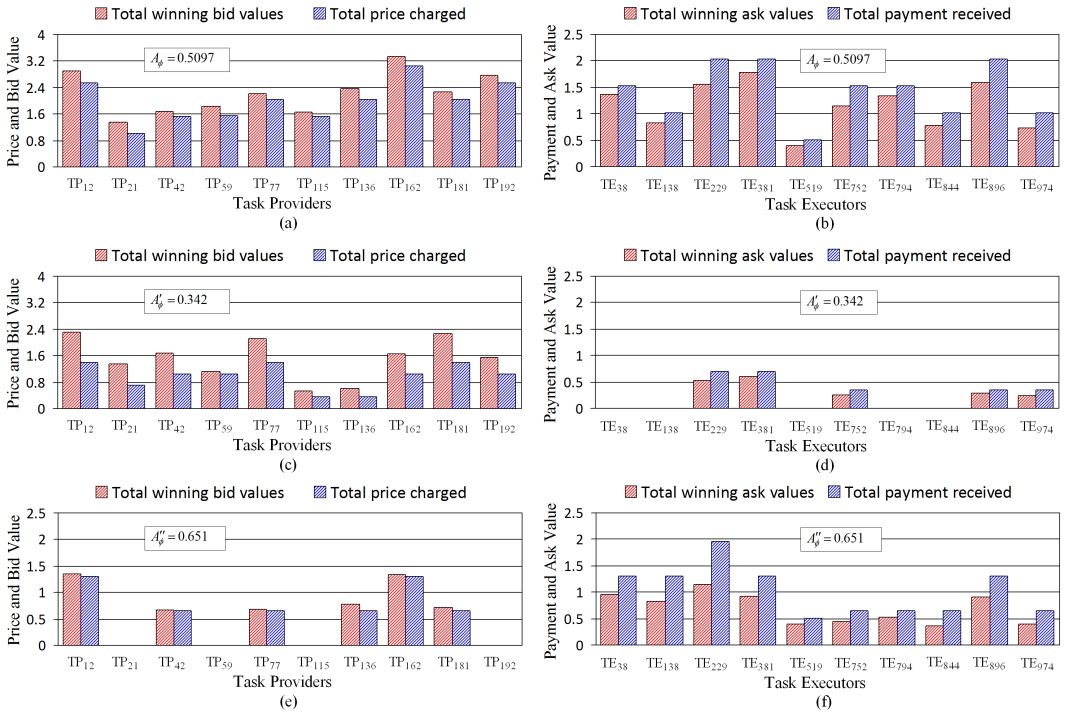


FIGURE 5.7: Compare the (a,c,e) bids and prices of winning TPs for different threshold settings and (b,d,f) payment and ask of winning TEs for different threshold settings.

Budget balance

The result in Fig. 5.8 is used to verify the budget balance property (Lemma 2) of TATA. In order to verify this property, in Fig. 5.8 (a), the total number of task executors are varied

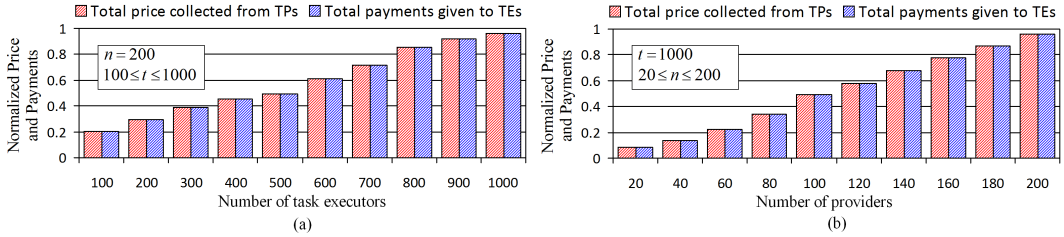


FIGURE 5.8: The normalized value of price collected from TPs and payments given to TEs.

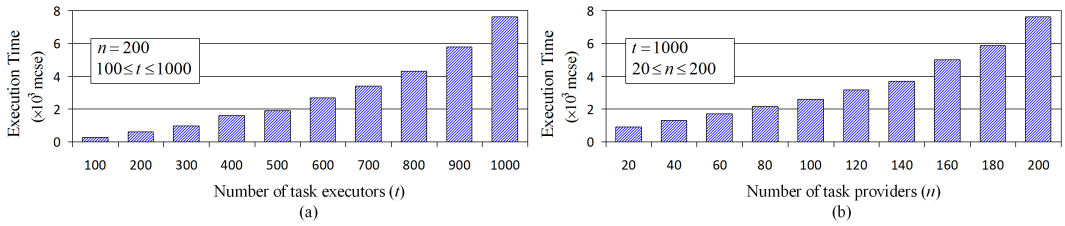


FIGURE 5.9: Execution time of TATA when (a) number of TEs increases from 100 to 1000 and (b) number of TPs increases from 20 to 200.

from 100 to 1000 with an increase of 100 task executors at every step. On the other hand, in Fig. 5.8 (b), the total number of task providers are varied from 20 to 200 with an increase of 20 task providers at every step. For each step, the total price charged to the winning task providers (i.e. $\sum_{TP_i \in \mathscr{P}_w} P_i^{(p)}$) and total payments given to the task executors (i.e $\sum_{TE_k \in \mathcal{E}_w} P_k^{(e)}$) are calculated. Note that, in TATA the total price collected from the winning task providers $TP_i \in \mathscr{P}_w$ is completely distributed to the winning task executors $TE_k \in \mathcal{E}_w$. It can be observed from Fig. 5.8 that the total price charged to the winning task providers is not less than the total payment given to the winning task executors i.e. $\sum_{TP_i \in \mathscr{P}_w} P_i^{(p)} \not< \sum_{TE_k \in \mathcal{E}_w} P_k^{(e)}$. Eventually, the auctioneer doesn't have any deficit. Hence, it is verified that TATA holds budget balance property.

Computational efficiency

In Fig 5.9, the execution time of TATA is examined and the computational efficiency property (Lemma 5) is verified. For different settings of the total number of task executors and the task providers, the execution times are computed. The results are averaged over 100 runs. As shown in Fig. 5.9(a) and (b), the execution time of TATA increases as the total number of task executors or the total number of task providers increase. Fig. 5.9(a) and (b) verify that TATA is subject to polynomial time with respect to the total number of participating task executors and the total number of task providers.

5.5.3 Performance Evaluation

Performance metrics

Following performance metrics are defined to assess the outcome of auction based incentive mechanism.

- *Winning Task Providers' Satisfaction ($TPS^{(w)}$):* It is defined as the ratio of the total number of successful trades to the total needs of the winning task providers.

$$TPS^{(w)} = \frac{|\mathscr{S}^{\mathscr{T}}|}{\sum_{TP_i \in \mathscr{P}_w} \sum_{R_j \in \mathscr{R}} N_i^j} \tag{5.17}$$

- *Winning Task Executors' Satisfaction ($TES^{(w)}$):* It is defined as the ratio of the total number of successful trades to the total number of non zero positive asks of the winning task executors.

$$TES^{(w)} = \frac{|\mathscr{S}^{\mathscr{T}}|}{|\mathscr{N}_w^{ask}|} \tag{5.18}$$

where, $\mathscr{N}_w^{ask} = \{TE_k^j | TE_k^j \in \mathscr{N}, TE_k \in \mathscr{E}_w\}$

- *Total Utility of Winning Executors ($TUE^{(w)}$):* It is the sum of the utilities of all the winning task executors.

$$TUE^{(w)} = \sum_{TE_k \in \mathscr{E}_w} \mathscr{U}_k^{(e)} \tag{5.19}$$

- *Total Utility of Winning Providers ($TUP^{(w)}$):* It is the sum of the utilities of all the winning task providers.

$$TUP^{(w)} = \sum_{TP_i \in \mathscr{P}_w} \mathscr{U}_i^{(p)} \tag{5.20}$$

Comparison with other incentive mechanisms

Now, the performance of TATA is evaluated and compared with two other auction based incentive mechanisms IMRAL [71] and LBS [63] with respect to *Winning Task Providers' Satisfaction ($TPS^{(w)}$)*, *Winning Task Executors' Satisfaction ($TES^{(w)}$)*, *Total Utility of Winning Executors ($TUE^{(w)}$)* and *Total Utility of Winning Providers ($TUP^{(w)}$)*. To perform the experiments, the total number of participating TEs and TPs are varied from 100 to 1000 and 20 to 200, and then performance is evaluated for each setting. Note that, in this experiment, the bids of TPs are normalized between the interval $D_i^j \in (0,1]$, the asks of TEs are uniformly distributed over $A_k^j \in (0,1]$, $N_{max} = 5$ and $|\mathscr{R}| = 100$.

Fig. 5.10 (a) and (b) show normalized $TPS^{(w)}$ of three incentive mechanisms- TATA, IMRAL [71] and LBS [63] for different settings. In the case of Fig. 5.10 (a), n is fixed to 200 and $100 \leq t \leq 1000$. On the other hand, in Fig. 5.10 (b), t is fixed to 1000 and $20 \leq n \leq 200$. As shown in Fig. 5.10 (a), for all the mechanisms, the normalized $TPS^{(w)}$ increases with the increase of the number of task executors (t). TATA achieves relatively high $TPS^{(w)}$, it is because, with the increase of the number of TEs, the number of winning bids increases (i.e. $|\mathscr{S}^{\mathscr{T}}|$ increases) for the fixed set of need vectors of TPs. Now, in Fig. 5.10 (b), with the increase of the number of TPs, the normalized $TPS^{(w)}$ decreases. It is because, with the

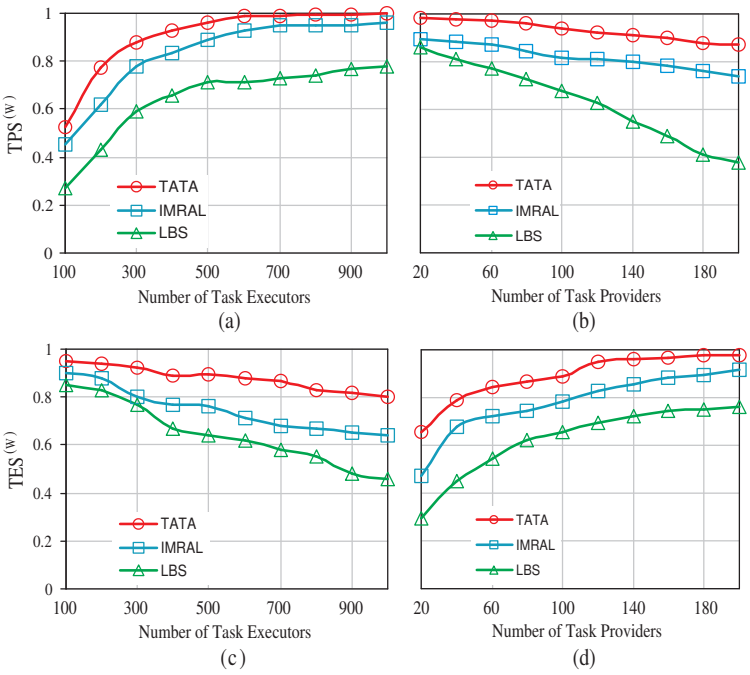


FIGURE 5.10: Task Providers' satisfaction when (a) number of task executors increases and (b) number of providers increases. Task Executors' satisfaction when (c) number of task executors increases and (d) number of providers increases.

increase of n, the number of winning TPs i.e. $|\mathscr{P}_w|$ increases. But, $|\mathcal{E}_w|$ remains the same and it will limit the total number of successful trades $|\mathscr{S}^{\mathscr{T}}|$.

In Fig. 5.10 (c) and (d) the normalized $TES^{(w)}$ of TATA, IMRAL and LBS are presented. Fig. 5.10 (c) shows that as the number of task executors (t) increases, the normalized $TES^{(w)}$ decreases. This is because, with the increase of t the number of winning TEs (i.e. $|\mathcal{E}_w|$) increases. But, $|\mathscr{P}_w|$ doesn't change much and as a result this will limit the total number of successful trades $|\mathscr{S}^{\mathscr{T}}|$. On the other hand, as shown in Fig. 5.10 (d), the normalized $TES^{(w)}$ increases with the increase of the number of task providers n. Here, with the increase of n, $|\mathscr{P}_w|$ and $|\mathscr{S}^{\mathscr{T}}|$ increases. This helps TATA to achieve high $TES^{(w)}$ value when n increases. Clearly, TATA dominates over IMRAL and LBS in terms of normalized $TES^{(w)}$.

Fig. 5.11 (a) and (b) compare the normalized $TUE^{(w)}$ among TATA, IMRAL and LBS for different settings. It is observed that TATA has a sharp surge in $TUE^{(w)}$ in both Fig. 5.11 (a) and (b). The reasons are as follows. In Fig. 5.11 (a), t is varied from 100 to 1000 while n is fixed to 200. In this case, more new asks with low values win in the auction as t increases. Eventually, the $TUE^{(w)}$ value increases. In Fig. 5.11 (b), n is varied from 20 to 200 while t is fixed to 1000. In this case, as n increases, the asks $A_k^j < A_\phi$ that are not yet allocated, will have a chance to be allocated. Eventually, in this case also the $TUE^{(w)}$ value increases. Similarly, in Fig. 5.11 (c) and (d), normalized $TUP^{(w)}$ increases in both the cases. In both the figures as t or n increases, the number of successful trades increases. Eventually, there is an increase in the $TUP^{(w)}$ value. In both cases, there is no mechanism that clearly dominates over others in terms of $TUP^{(w)}$.

We have also explored how executor elimination helps executors achieve more utility.

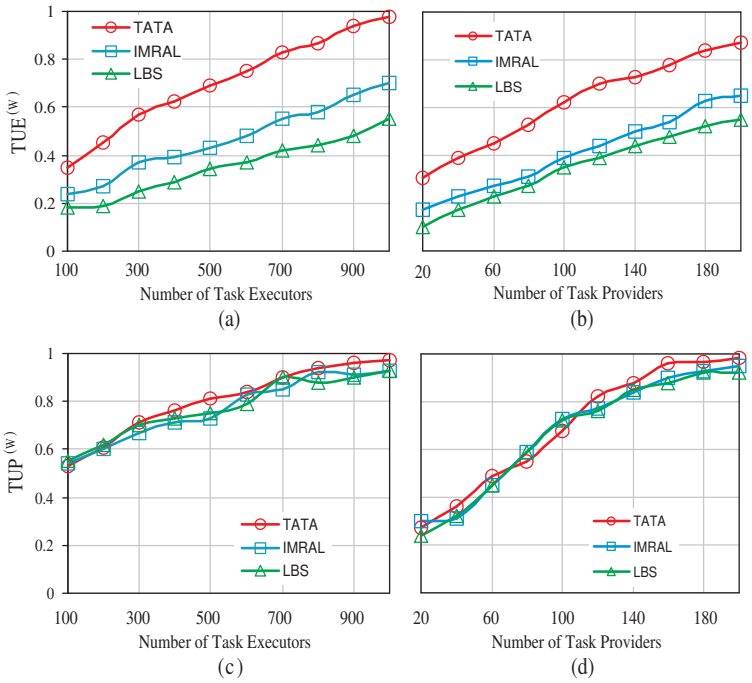


FIGURE 5.11: Total utility of task executors when (a) number of task executors increases and (b) number of providers increases. Total utility of task providers when (c) number of task executors increases and (d) number of task providers increases.

To accomplish this, we determined TATA's total utility when the executor elimination in the candidate trade determination phase was disregarded. In order to find the total utility of TEs without executor elimination in the candidate trade determination phase, we do the following: We choose the task executors at random based on needs mentioned in the need matrix if the task provider wins more task executors than it requires. Fig. 5.12 shows the comparison result of total utility of task executor for TATA and TATA w/o EE (TATA with out executor elimination). Specifically, $TUE^{(w)}$ of TATA and TATA w/o EE are compared in Fig. 5.12(a) by varying the number of TEs when the the number of TPs is fixed at 200. On the other hand, Fig. 5.12(b) compares $TUE^{(w)}$ of TATA and TATA w/o EE by varying number of TPs while keeping the number of TEs constant at 1000. It is observed that task executors achieve greater utility in TATA than in TATA w/o EE in both cases.

Finally, a theoretical comparison among TATA, IMRAL, and LBS is provided in Table 5.7.

Impact of ϕ and N_{max} on auction outcome

In the candidate trade determination phase (Algorithm 2) of TATA, candidate trades are selected based on $\phi - th$ ask value (i.e. A_ϕ) of the ordered list $\mathcal{N}'$ of all TEs' non-zero positive asks. Let us consider, $\mathcal{N}_\phi$ contains $TE_k^j \in \mathcal{N}$ such that $A_k^j < 0$. Also, consider, $\mathcal{M}_\prec$ contains $TP_i^j \in \mathcal{M}$ such that $D_i^j \geq A_\phi$. The cardinalities of $\mathcal{N}_\phi$ and $\mathcal{M}_\prec$ depend on the value of ϕ. Intuitively, as the value of ϕ increases, $|\mathcal{N}_\phi|$ increases, but $|\mathcal{M}_\prec|$ decreases. On the other hand, if the value of ϕ decreases, $|\mathcal{N}_\phi|$ decreases, but $|\mathcal{M}_\prec|$ increases. From Fig. 5.3, it is clear that these two sets affect the outcome of TATA. Fig. 5.13 reveals the impact of ϕ on the

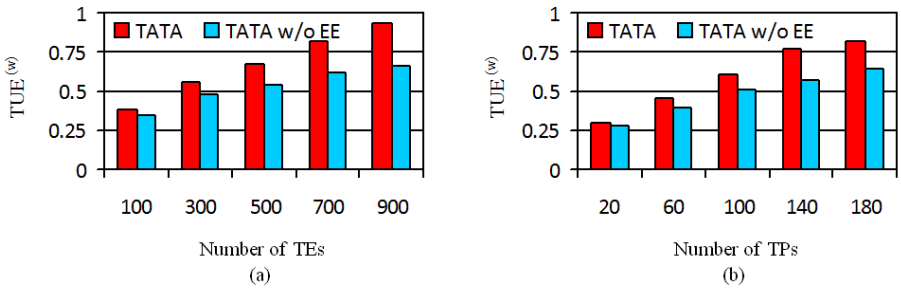


FIGURE 5.12: Comparison of total utility of task executor for TATA and TATA w/o EE (TATA with out executor elimination) when (a) number of task executors increases and (b) number of providers increases.

TABLE 5.7: Comparison among IMRAL, LBS and TATA (for same dataset)

Basis of comparison	IMRAL [71]	LBS [63]	TATA
Truthfulness	Holds truthfulness for task executors.	Doesn't hold truthfulness	Holds truthfulness for both task executors and providers
$TPS^{(w)}$	Achieves high $TPS^{(w)}$ than LBS	$TPS^{(w)}$ of LBS is lower compared to IMRAL and TATA	TATA achieves high $TPS^{(w)}$ than both IMRAL and LBS
$TES^{(w)}$	$TES^{(w)}$ is higher than LBS but lower than TATA	$TES^{(w)}$ is less than both IMRAL and TATA	$TES^{(w)}$ is more than both IMRAL and LBS
Normalized $TUE^{(w)}$	$TUE^{(w)}$ is higher than LBS but lower than TATA	$TUE^{(w)}$ is lower compared to both IMRAL and TATA	$TUE^{(w)}$ is more than both IMRAL and LBS
Normalized $TUP^{(w)}$	$TUP^{(w)}$ is almost same for all the methods	$TUP^{(w)}$ is almost same for all the methods	$TUP^{(w)}$ is almost same for all the methods
Computation time	$O(cn\log n)$	$O(\frac{1}{\varepsilon}n^3 c\log c)$	$O(n^2t+nt^2)$
Individual rationality	Holds individual rationality property	Holds individual rationality property	Holds individual rationality property
Budget balance	Holds budget balance property	Holds budget balance property	Holds budget balance property

n : Total number of task providers; t : Total number of task executors; c : Total number of tasks; ε : a constant term; $TPS^{(w)}$: Winning task providers' satisfaction; $TES^{(w)}$: Winning task executors satisfaction; $TUE^{(w)}$: Total utility of winning executors; $TUP^{(w)}$: Total utility of winning providers.

number of successful trades, total utility of TEs, and total utility of TPs with different N_{max} values.

Fig. 5.13 (a) shows the plot of the number of successful trades ($\mathscr{S}_{\mathscr{T}}$) for different values of ϕ with 100 TPs and 100 TEs. The variation in the number of successful trades depend on the effect of ϕ on the size of $\mathscr{N}_{\phi}$ and $\mathscr{M}_{\prec}$. When the value of ϕ becomes small or large, the size of $\mathscr{N}_{\phi}$ or $\mathscr{M}_{\prec}$ becomes small. Eventually, the number of successful trades ($\mathscr{S}_{\mathscr{T}}$) becomes small. The interesting thing is that as N_{max} increases, the peak value of the number of successful trades also increases. It is very obvious, as a high value of N_{max} leads to less number of candidate task elimination by TATA-STD() and eventually results in high $|\mathscr{S}_{\mathscr{T}}|$ value. Now, Fig. 5.13 (b) and Fig. 5.13 (c) show the impact of ϕ on the total utility of TEs and the total utility of TPs respectively. Both Fig. 5.13 (b) and Fig. 5.13 (c) display a similar trend as in Fig. 5.13 (a). It is because, the total utility of TEs and the total utility of TPs are proportional to $|\mathscr{S}_{\mathscr{T}}|$. Based on the observations in Fig. 5.13 (a)-(c), it is found that the optimal value of ϕ that achieves the best performance is located around $\phi = \lceil (r+1)/2 \rceil$ where $r = |\mathscr{N}'|$. That's why the value of ϕ is selected as $\phi = \lceil (r+1)/2 \rceil$ in the experiment. This value of ϕ is shown by dotted vertical lines in Fig. 5.13 (a)-(c).

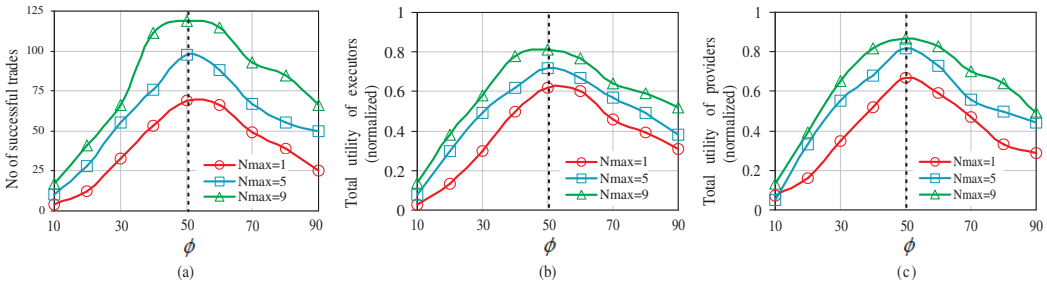


FIGURE 5.13: The impact of ϕ and N_{max} on the outcome of TATA. (a) Number of successful trades vs ϕ (b) Total utility of task executors vs ϕ (c) Total utility of task providers vs ϕ. The dotted vertical lines indicate the optimal value of ϕ at which the best performance of TATA is achieved.

5.6 Summary

In this chapter, a double auction based incentive mechanism called TATA (Truthful Double Auction for Task Allocation) is proposed for participatory sensing based systems. It consists of three major stages, namely candidate trade determination, successful trade determination and price calculation. TATA can efficiently allocate the task executors to the task providers for performing their sensing tasks. TATA holds certain desirable properties like truthfulness, budget balance, computational efficiency, individual rationality, etc for both task executors and task providers. Rigorous proofs of these desirable properties of TATA are provided and also verified with extensive simulation results. The results also indicate that TATA posses high Winning Task Providers' Satisfaction ($TPS^{(w)}$), Winning Task Executors' Satisfaction ($TES^{(w)}$), Total Utility of Winning Executors ($TUE^{(w)}$), and Total Utility of Winning Providers ($TUP^{(w)}$).

Note that in Chapters 4 and 5, we addressed privacy and incentive-related issues and challenges in participatory sensing. Having resolved those concerns, we are now ready to investigate issues (refer to Section 1.2) pertaining to the predictive analysis of spatio-temporal variability in urban dynamics. In the next chapter, we focus to a specific urban dynamic-noise pollution-where we conduct predictive analysis of spatio-temporal variability. We aim to develop techniques capable of addressing diverse scenarios in spatial data analysis, including sparse datasets, high accuracy, better response time, etc for participatory sensing data.

Publication(s) related to this Chapter:

- *A.I. Middya, and S. Roy, 2024. Truthful double auction based incentive mechanism for participatory sensing systems. Peer-to-Peer Networking and Applications. (DOI: https://doi.org/10.1007/s12083-024-01681-3*

Chapter 6

Predictive Modeling and Spatio-temporal Analysis of Environment Data

6.1 Introduction

Spatial distributions of data of natural phenomena can be estimated at high granularity by designing efficient spatial interpolation techniques. These methods can be applied to evaluate a variety of urban dynamics. For example, they can be used to create urban noise pollution monitoring applications that can accurately describe the high granular variability of urban noise pollution levels in any region of interest, allowing for more effective and informed decision-making. Here, our goal is to leverage spatial interpolation techniques to create dynamic (i.e., time and space-varying) noise maps as a service using an IoT-cloud-based framework. Making noise maps is a useful tool for evaluating and visualizing urban noise pollution.

Noise pollution today has become a major problem that may cause serious health threats like hearing loss [214], increasing stress levels [215], behavioral and mental disorders [216], hypertension [217], sleep disorders [218], heart disease [219] and many more [220] [221]. It also affects the behavior and habitat of animals. According to a recent report released by WHO (World Health Organization), there is a piece of strong evidence that noise is one of the top environmental hazards to both physical and mental health and well-being [222]. Another report from EEA (European Environment Agency) has revealed that environmental noise results in more than 16600 premature death in Europe every year, with approximately 13 million adults suffering from sleep disturbance and a further 32 million annoyed by it. Almost 15% population (26 million people) of the U.S.A may have suffered from NIPTS (noise-induced permanent threshold shift) which is a permanent reduction in hearing acuity due to excessive exposure to workplace or leisure noise [223]. Now, according to the directive 2002/49/EC [224], city noise maps are the key resource for analyzing and decision making in the development of action plans to reduce urban noise pollution levels. Here, a city noise map is a geospatial representation of noise pollution level distribution in a city for a particular time interval.

Noise pollution monitoring is generally not an easy task to completely and accurately determine the urban noise pollution in a large region. Traditionally, noise assessment in any region is usually performed using a few noise monitoring stations that are sparsely spread over different locations (e.g. near streets, railway stations, hospitals, airports, residential and commercial areas) of a city. In noise pollution monitoring the noise mapping can be cast as a modelling task (e.g. emission and propagation models [225] [226]). The sparse monitoring stations' data are mainly used to validate the mapping results. Also, there is little existing literature that involves noise mapping techniques for interpolating between noise measurements in urban noise measurement networks [227] [228]. Although the traditional method of noise pollution data collection provides more accurate measurements, it suffers from some limitations. Firstly, using sparse spatio-temporal data, traditional noise mapping models don't help us to create a spatially continuous noise map of high spatial and temporal resolution [82] for a large area like a city. Secondly, it is hard to monitor and assess unusual local or short-term noise pollution. Thirdly, the traditional approach for noise pollution monitoring is costly due to the need for human expertise, expensive equipment, periodic maintenance of equipment, and effort required to monitor noise pollution.

6.1.1 Objective

To overcome the limitations mentioned above, we intend to use a relatively new paradigm of sensing known as *participatory sensing*. Specifically, the main objective of this chapter is to come up with an end-to-end participatory sensing system that can provide a high granular and spatially continuous noise map as a service to the users. This high granular and spatially continuous noise map will make it easier to monitor environmental noise in urban areas. It can also enhance citizens' awareness of environmental noise pollution, and help in developing mitigation plans to tackle the adverse effect of environmental noise pollution.

Low-cost IoT devices like smartphones generate low-quality data. In other words, smartphone measurements of ambient noise levels may not produce true values. In order to produce true noise levels, our another objective is to develop IoT device specific proper calibration mechanisms. Moreover, in participatory sensing data may be generated from different sensing contexts (e.g. the phone is in hand, in pocket/bag or an active phone call is going on, etc.). The ambient noise measurements using this approach may not provide accurate values due to inappropriate sensing context (e.g. when the phone is in pocket/bag). Hence, we aim at exploring proper context-aware data cleaning technique to remove the measurements collected from inappropriate sensing contexts.

Finally, the process of collecting participatory sensing data depends on the mobility pattern of the participants, and certain regions may experience a lack of data leading to a spatially sparse dataset. A sparse dataset makes it difficult to estimate urban noise at high spatial granularity. Note that the spatial missing values in the participatory sensing based data also prevent us to achieve high accuracy. Moreover, a huge noise pollution dataset can make predictive analysis computationally expensive and slow. This may significantly affect the response time if the predictive models are deployed in an end-to-end framework to provide noise maps as a service to the users. Eventually, our additional objectives are to design

efficient techniques that can address scenarios like *sparse datasets*, *high accuracy*, *better response time* in order to analyze, generate, and visualize a dynamic noise map.

6.1.2 Contribution

Our key contributions are summarized as follows:

- An end-to-end IoT-cloud based framework is introduced that enables the users to get spatially continuous noise map as a service based on their query.
- A device specific calibration mechanism is developed. Additionally, a classification algorithm is developed that detects the sensing contexts (whether the phone is in the pocket or in hand) so that noise data collected during inappropriate sensing contexts can be removed.
- We propose three methods (GLIDW, I-GLIDW, GLIDW-OK) for dynamic noise map generation using spatial interpolation techniques based on participatory sensing data. The results show that our proposed methods (I-GLIDW, GLIDW-OK) perform more accurately compared to the baseline methods (IDW, OK) for participatory sensing noise data.

6.2 Existing Works

This section provides a background study explaining existing literature related to participatory sensing based monitoring of urban dynamics including urban noise pollution mapping. A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics is provided in Table 6.1. This table compares the existing works in terms of the contribution, sensors used, context-awareness, calibration, map generation, map type, spatial interpolation, consideration of data sparsity, and end-to-end system behaviour.

Participatory sensing using IoT devices (e.g. smartphones) has recently been used as a sensing method to explore a city's various dynamics (e.g. pollution, human mobility, road condition, etc.) at high spatiotemporal resolution [3, 10, 15, 30]. The widespread adoption of smartphones enables citizens to take part in producing urban noise pollution data. These urban noise pollution data can be then collectively used to offer various services to the citizens. For instance, in [82], Maisonneuve et. al, present an approach called NoiseTube, for the assessment of noise pollution using the data collected from users' smartphone sensors like GPS, microphones, etc. In other words, they use GPS-equipped mobile phones to measure citizens' exposure to noise in their daily life. They combine all the geotagged smartphone measurements and create a PoI (Point of Interest) based map visualization of urban noise pollution data. However, in this work, no data cleaning technique is implemented to eliminate the measurements that are collected from the inappropriate sensing contexts (e.g. when the phone is in a bag/pocket). Eventually, the urban noise map produced in NoiseTube may suffer from erroneous data and may not reflect the actual reality. Also, the noise map created here is not spatially continuous and therefore may not help efficient decision-making.

Becker et. al. [229] developed a smartphone application called WideNoise, that can be used to involve citizens in noise measuring activities. The application WideNoise is developed to record both objective (noise samples) and subjective (opinions, feelings, etc.) data. Using the data provided by smartphone users, the analysis of awareness and learning is performed about current environmental issues. But, the authors mainly focused on presenting a rigorous analysis of participatory sensing based urban noise data, rather than high granular urban noise mapping. In [191], Rana et. al. show the use of different interpolation and regularization methods to reconstruct the environmental noise pollution map from incomplete and random noise samples collected using participatory data collection. However, they did not make any attempt to calculate spatially continuous noise maps by estimating urban noise levels at unsampled locations. In [104], an end-to-end system called *NoiseSense* is proposed for urban noise monitoring and PoI (Point of Interest) based noise mapping. This work mainly focused on the overall monitoring system. The authors didn't use any interpolation technique to calculate noise levels at missing locations. It is interesting to see that none of the existing literature addresses the problem of data sparsity.

The noise pollution data that are collected using IoT based approach can be used by spatial interpolation methods to generate spatially continuous noise pollution maps. More specifically, spatial interpolation is the process of using the values at observed locations to predict the values at other unknown locations in geographic space. The rationale behind spatial interpolation is the assumption that everything is related to everything else, but near things are more related than distant things in space (Tobler's Law of Geography [89]). In literature, there are several methods available for spatial interpolation [86]. Among them, the two popular methods are Inverse Distance Weighting (IDW) [133] and Kriging [88] which are frequently used in the field of environmental science.

6.2.1 Inverse distance weighting method

Inverse distance weighting (IDW) [133] method is one of the simplest and widely used spatial interpolation methods. It is based on an assumption that the value at any unmeasured location can be predicted as a weighted average of measured values surrounding the prediction location. It assigns greater weights to the points that are closer to the prediction location than the points that are far away. The general form to calculate the predicted value u at an unknown point x based on sample $u_i = u(x_i)$ for $i = 1,2,\ldots,N$ using IDW is given as follows [133]:

$$u(x) = \begin{cases} \dfrac{\sum_{i=1}^{N} \dfrac{1}{d(x,x_i)^p} u_i}{\sum_{i=1}^{N} \dfrac{1}{d(x,x_i)^p}}, & \text{if } d(x,x_i) \neq 0 \text{ for all } i \\ u_i, & \text{otherwise} \end{cases} \tag{6.1}$$

where, x denotes prediction point (interpolated point), x_i is a known point (interpolating point), $d(x,x_i)$ is the distance from the interpolating point x_i to the interpolated point x, N is the total number of known points used in interpolation and p is the power parameter.

TABLE 6.1: A detailed comparison of the existing works related to participatory sensing based monitoring of urban dynamics.

City Dynamics	Reference & year	Contribution	Sensors used	Context-aware data cleaning	Calibration	Map generation	Map type	Spatial interpolation	Consideration of data sparsity	End-to-End System
Human mobility	Yabe et. al. [33], 2016	Predicts irregular movement of humans after the mid-level disasters.	GPS	×	×	×	×	×	×	×
	Gu et. al. [12] 2017	Predict venue(s) visited by citizens from the GPS trajectory.	GPS	×	×	×	×	×	×	×
	Tokuda et. al. [35], 2020	Generate spatiotemporal pedestrian density map using street-level images.	GPS, Camera	×	×	SNCM	STA	×	×	×
Road condition	Bhatt et. al. [230], 2017	Develop a system for detection and mapping of potholes.	GPS, Acc, Gyro	×	×	SNCM	STA	×	×	✓
	Hassan et. al. [231], 2019	Low-Cost Road Maintenance and road condition mapping.	GPS, Acc	×	×	SNCM	STA	×	×	×
	Gupta et. al. [13], 2020	estimate the transverse slope profile of road segments.	GPS Acc, Gyro	×	×	×	×	×	×	×
Air pollution	Hasenfratz et. al [27], 2012	Develop a smartphone based system for air pollution monitoring.	GPS, Ozone sensor	×	✓	SNCM	STA	×	×	×
	Dutta et. al [7], 2017	Propose a system for crowdsensing based air pollution monitoring.	GPS, MQ-135, HC-05	×	×	SNCM	STA	×	×	✓
	Middya et. al [18], 2020	Propose a unified framework for urban sensing systems.	GPS, MQ-135, HC-05	×	×	SCM	STA	✓	×	✓
Noise pollution	Maisonneuve et. al [82], 2009	Assessment of noise pollution with mobile phones	GPS, microphone	×	✓	SNCM	STA	×	×	✓
	Becker et. al. [229], 2013	The behaviour of people engaged in noise sensing are analysed.	GPS, microphone	×	×	SNCM	STA	×	×	×
	Rana et. al [191], 2015	Reconstruction of noise map from incomplete and random noise samples.	GPS, microphone, Acc., Ptx.	✓	✓	SNCM	STA	×	×	✓
	Zamora et. al [232], 2017	Evaluate the impact of various algorithms for noise computation.	GPS, microphone	×	✓	SNCM	STA	×	×	×
	Dutta et. al [104], 2017	Real time monitoring of urban noise pollution using MCS.	GPS, microphone	×	✓	SNCM	STA	×	×	✓
	Marques et. al [233], 2019	Real time monitoring of urban noise pollution using MCS.	GPS, microphone	×	×	SNCM	STA	×	×	✓

✓: The requirement has been met, × : The requirement has not been met, Acc : Accelerometer sensor, Gyro: Gyroscope sensor, SCM: Spatially Continuous Mapping, SNCM: Spatially Non-Continuous Mapping, STA: Static Map.

6.2.2 Kriging

Kriging is a frequently applied geostatistical method for spatial interpolation. There is a large body of literature describing the kriging method [88] [86] [234]. Kriging first measures the spatial correlation by measuring the semivariance, based on the assumption that the distance between interpolating points reflects a spatial correlation. We can better capture the variation in the surface with the aid of this spatial correlation. We can then model the similarity of observation points as a function of distance or lag using a semivariogram, which is a plot of semivariance as a function of distance or lag. Specifically, a semivariogram allows measuring the degree of similarity among the observation points separated by a given distance. The

semivariogram can be computed from sample points by using Eq. 6.2 as given below [86].

$$\widehat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} \{z(x_i) - z(x_i + h)\}^2 \tag{6.2}$$

Here, $\widehat{\gamma}(h)$ is the estimated semivariance at a distance h, and $z(x_i)$ and $z(x_i + h)$ denote values at x_i and $x_i + h$ that are separated spatially by distance h. The number of pairs of sample points that are spatially separated by the distance h is denoted by the term $N(h)$. Now, the next step in kriging is to fit a model to the empirical semivariogram such that semivariances can be obtained at all possible separation distances (i.e. lags). There are popular models (like Spherical, Exponential, Gaussian, Linear, etc) that are given in kriging literature [235]. After the model is fitted to the empirical semivariogram, it can be used to calculate the weights λ_i that minimize the variance in the prediction. The following formula can now be used to make an estimation. [236]:

$$\hat{z}(x_0) - \mu = \sum_{i=1}^{k} \lambda_i [z(x_i) - \mu(x_0)] \tag{6.3}$$

where, μ is called stationary mean which is calculated as the average of the whole dataset [237], λ_i is kriging weight, k and $\mu(x_0)$ are the total numbers of observation points and the mean of the observation points within the search neighborhood. Many kriging variations can be found in kriging literature [235]; some traditional techniques include Ordinary Kriging (OK), Simple Kriging (SK), and Universal Kriging (UK). In the case of ordinary kriging, Eq. 6.3 is modified by replacing μ with a local mean $\mu(x_0)$ which is the mean of the observation points within the search neighborhood [238] [239]. On the other hand, in the case of simple kriging, Eq. 6.3 is modified by replacing $\mu(x_0)$ with the stationary mean μ. In other words, stationary mean μ is assumed to be constant over the whole domain. Universal kriging (UK) is an extension of Ordinary Kriging that takes into account the spatial variability in z values as well as the spatial correlation between sample points.

In the context of noise map generation, we have proposed three spatial interpolation techniques that are inspired from IDW and OK to address diverse scenarios like *sparse dataset*, *high accuracy*, *better response time*, etc.

6.3 Problem statement

Given a set of users $U = \{User_1, User_2, \ldots, User_n\}$ and a set of noise samples $S = \{s_1, s_2, \ldots, s_m\}$ collected from those users' smartphones, where, any noise sample s_i should be in the form as given in Section 6.4.1. Suppose, $S_{pq} \subseteq S$ is a the set of samples within a given time interval $T_{pq} = [t_p, t_q]$, where t_p and t_q are start and end of the time interval. Let us consider, $\mathsf{G} = \{g_1, g_2, \ldots, g_k\}$ is a set of grid cells. Our objective is to come up with efficient technique (or techniques) that utilize the samples in S_{pq} to find a set of grid cells' urban noise values $N_{\mathsf{G}}^{(pred)} = \{N_1^{(pred)}, N_2^{(pred)}, \ldots, N_k^{(pred)}\}$ by addressing diverse scenarios like sparse dataset, high accuracy, better response time, etc., where $N_i^{(pred)}$ is the predicted noise value of grid cell $g_i \in \mathsf{G}$. Additionally, our aim is to create a noise map for the given interval T_{pq} using

TABLE 6.2: Notation

Symbols	Description
$d(x,x_i)$	Euclidean distance between points x_i and x
p	Power parameter of IDW
$\widehat{\gamma}(h)$	Estimated semivariance at a separation distance h
$N(h)$	Number of pairs of sample points separated by distance h
Z_i	Observed value at point i
L_{eq}	Equivalent continuous sound level
sm	Smartphone model name
si	Smartphone state information
$\xi^{sm_i}(L_{eq})$	Calibration offset of noise reading L_{eq} in smartphone model sm_i
$L_{eq}^{'}$	Noise level after adding calibration offset ξ to L_{eq}
acc_x	x-axis data of accelerometer reading
acc_y	y-axis data of accelerometer reading
acc_z	z-axis data of accelerometer reading
n	The dimension of each grid cell
$\mathbf{S}$	Sample data
G	Set of all grid cells having size $n \times n$
g_i	i^{th} grid cell
R	Radius that is considered for computing weight
m_i	Midpoint of grid cell g_i
$N_{\mathsf{G}}^{(pred)}$	Predicted noise values for all grid cells in G
N_{Th}	Number of nearest neighbour required to calculate a grid cell's noise
NN_i	A set of sample points that are within radius R from the midpoint m_i
R'	Radius for iterative computation in I-GLIDW
G_{Th}	A predefined threshold for I-GLIDW
V^m	Set of variogram models

those predicted noise values and serve this to the users for visualizing and assessing noise pollution based on their query.

6.4 Proposed Approach

6.4.1 Participatory sensing based framework

Fig. 6.1 provides an overview of the framework of our participatory based noise map generation technique, which consists of three parts: sensing and data collection, pre-processing, and service. Noise levels are measured on the smartphones and then transmitted to the server. The server combines spatio-temporal participatory sensing noise measurements collected from different participants. Note that, combining ambient noise data from different users is required because the noise readings are not available for all spatial and temporal resolution for a particular user.

Sensing and data collection

The sensing and data collection required participants to first install our application NoiseSense [104] for Android. As shown in the leftmost portion of Fig. 6.1, using that Android application, we measure spatio-temporal noise data (see Section 6.4.2) and trajectories

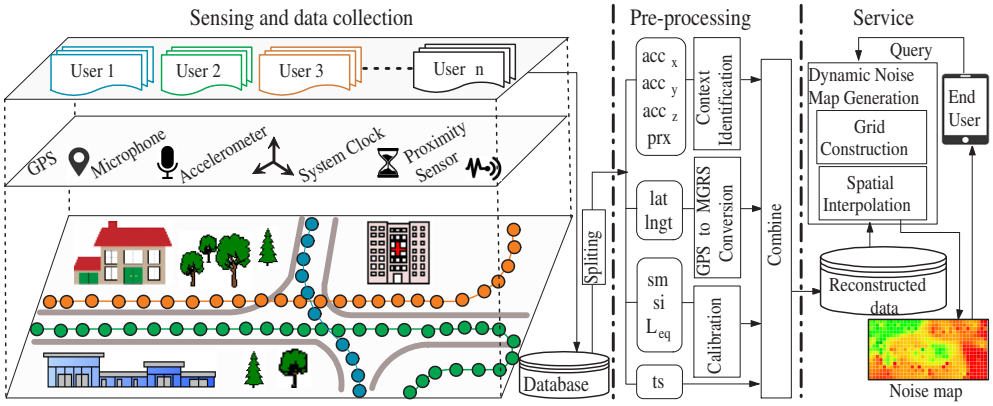


FIGURE 6.1: Participatory sensing based framework

(sequence of GPS coordinates) for participants. The measured noise level and GPS location (latitude and longitude) are further tagged with some other sensor values like 3-axis accelerometer, proximity sensor values for context-aware (such as whether the phone is in hand or in pocket/bag) data cleaning (see Section 6.4.3). Using the NoiseSense application, data samples get recorded at an interval of every 5 seconds in the format $< user_id, sm, si, lat, lngt, ts, L_{eq}, acc_x, acc_y, acc_z, prx >$ and are uploaded to a database. Here, $user_id$ is the unique user identifier; sm denotes smartphone model name; si is the state information (whether an active phone call in progress or idle) of the phone; lat and $lngt$ are latitude and longitude of the sampling location respectively; ts is the time stamp value; L_{eq} is the recorded noise level; acc_x, acc_y and acc_z are three axis accelerometer values; and prx is the proximity value. An important thing is to be noted that users can share all the above mentioned sensors data anonymously in order to preserve their privacy.

Pre-processing

In the pre-processing stage, the dataset is first split into four parts. The 1st part contains 3-axis accelerometer and proximity sensor values for context classification (see Section 6.4.3) to identify noise sensing contexts i.e whether the phone was in hand or pocket during data collection. This is necessary to decide whether the noise samples are accurate or not. The 2nd part contains GPS coordinates (latitude and longitude). We convert the GPS coordinates to MGRS (Military Grid Reference System) grid index (see Section 6.4.4). The 3rd part contains the smartphone model name (sm), recorded noise levels (L_{eq}) and state information (si). We have discussed in the later part (see Section 6.5.2) of this chapter, the calibration offset may vary from phone to phone and also under different sound levels. We only calculate calibration offset for those sm and L_{eq} pair for which $si = idle$. In other words, we are calculating calibration offset for those noise samples that are recorded while there was no phone call in progress. The 4th part contains only the timestamps and we keep this unchanged. After all pre-processing, we reconstruct the original dataset in such a way that we keep only those noise samples with modified noise levels (i.e. $L_{eq} + offset$), MGRS grid index and timestamps for which context is *"in hand"* and $si = idle$.

Noise map as a Service

Based on the reconstructed dataset we can come up with noise map of the Region Of Interest (ROI) using our proposed spatial interpolation algorithms. Before creating a noise map we divide the ROI into grid cells. Then based on users' query (for example, a user may be interested to view the evening session's noise map of a particular region) any of our proposed algorithms (GLIDW, I-GLIDW, GLIDW-OK) can be used to produce a noise map. We discuss later in the result section that the predictive accuracy and response time of a method depend on the data sparsity level of a dataset. Depending on how sparse the dataset is we choose an appropriate method for noise map generation (see Section 6.5.4). We use GLIDW if the dataset is dense. On the other hand, we prefer to use I-GLIDW and GLIDW-OK when the data sparsity levels of the dataset are medium and high respectively.

6.4.2 Urban noise level measurement

It is quite common to measure urban noise levels (or sound levels) as equivalent continuous sound level (L_{eq}). Usually, sound pressure levels fluctuate over time. L_{eq}, which is measured in decibel (dB), is used to represent sound pressure levels that fluctuate over a time interval by taking into account the total sound energy over the time interval. In other words, L_{eq} would result in the same effects as the varying sound levels over a given time interval. The equivalent continuous sound level (L_{eq}) is calculated using the following equation [138]:

$$L_{eq} = 10\log_{10}\left[\frac{1}{t_e - t_s}\int_{t_s}^{t_e}\left(\frac{p_t}{p_o}\right)^2 dt\right] \tag{6.4}$$

Here, t_s and t_e are the start and end of the time interval respectively, p_t is the acquired sound pressure of the sound signal and p_o is reference sound pressure level (typically 20 μPa). This noise value (L_{eq}) may not be the true value, we need to add an offset ξ which is calculated by calibrating the smartphone against a standard device (e.g. Sound Level Meter). Hence, the modified noise value is determined as follows:

$$L_{eq}^{'} = L_{eq} + \xi \tag{6.5}$$

Later, in Section 6.5.2, we explain how to calculate the value of ξ.

6.4.3 Context aware data cleaning

Note that, not all noise samples give accurate noise measurements. The noise sensing contexts (for example, whether the phone is in the user's hand or pocket) determine whether the noise samples are accurate. We discard those noise samples that are recorded in sensing contexts that provide inaccurate noise measurements. Some examples of such sensing contexts that provide inaccurate noise measurements are: (i) an active phone call in progress, (ii) phone is kept in pocket, etc.

It may happen that the noise levels are recorded while there is an active phone call in progress. Hence, we need to correctly identify this scenario and eliminate the noise values that are recorded during that time. In order to do that, we also record the state information of the phone (idle or active call in progress) along with the ambient noise value. We only keep those data records for which the phone is in the idle state (i.e. no active phone call in progress) and remove others.

The accuracy of the ambient noise data that are collected using the smartphones also depends on the sensing contexts like whether the smartphone is carried in hand or pocket. Usually, noise samples give accurate noise measurements when the smartphone is carried by hand. On the other hand, noise samples give inaccurate noise measurements when the smartphone is kept in the pocket. Hence, we need to correctly identify these two scenarios to eliminate the noise values that are recorded while the phone was in the pocket and keep those noise values that are recorded while the phone was in hand. We use a classification algorithm to identify these two contexts. We examine the feasibility of using the sensor data from accelerometer, gyroscope, light sensor, and the proximity sensor as features for classification. We appoint 5 subjects to record the readings using smartphone sensors. Each of the subjects is given two identical smartphones each keeping in a different sensing context (i.e. each subject is carrying one smartphone in hand and the other in pocket). Each subject is asked to collect 30 minutes of data. After analyzing the sensor data we decide to exclude the gyroscope and light sensor as their values are not significantly different in the two sensing contexts. We decide to use the accelerometer and the proximity sensor values for the purpose of classification. We represent normalized mean total acceleration and proximity sensor values for different subjects in different context in Fig. 6.2. Here, the mean total acceleration is calculated using the following equation (Eq 6.6). Here, a_{xi}, a_{yi} and a_{zi} are x, y and z axis data of the i^{th} accelerometer reading respectively.

$$\overline{m} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(a_{xi})^2 + (a_{yi})^2 + (a_{zi})^2} \tag{6.6}$$

We use Naive Bayes algorithm for context classification. We choose Naive Bayes as it can not only predict with fair accuracy (81%) but also easy to interpret and lightweight compared to other effective machine learning algorithms like support vector classification. Since it is not the main focus of our work, we do not pay much attention on improving the accuracy.

6.4.4 GPS to MGRS Convertion

For analyzing the participatory sensing based noise data, the Military Grid Reference System (MGRS) [240] is used. MGRS provides a way to represent the surface of the Earth into squares of small grid cells. In this work, we convert the GPS coordinates of noise samples into MGRS grid cells of $1m^2$. The major motives for approximating GPS with the small square areas are as follows: Firstly, it is not feasible to infer the noise value for each and every possible GPS coordinate due to the large number of such coordinates. Secondly, the standards for noise pollution monitoring advised to measure the pollution levels in such square areas [241]. We can use the MGRS grid reference system without making any changes. But,

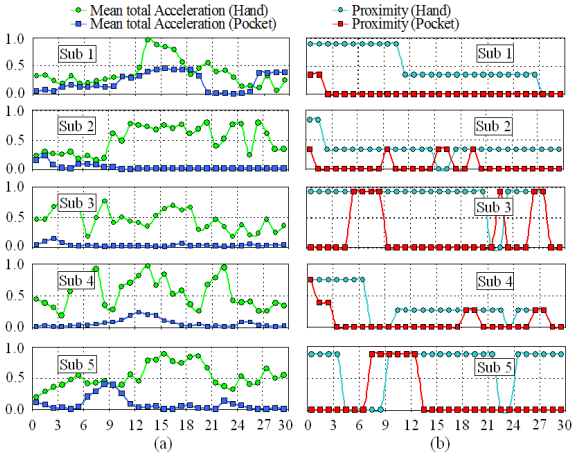


FIGURE 6.2: Normalized mean total acceleration and proximity sensor behavior for hand and pocket contexts for five different subjects

in the case of other reference systems like Universal Transverse Mercator (UTM), Universal Polar Stereographic (UPS) [242] etc some modifications might be required. Now, one thing to be noted here is, the dimension of the grid cell for the proposed methods (namely GLIDW, I-GLIDW and GLIDW-OK) may not be the same as the MGRS grid dimension. For instance, using parameter selection (see Section 6.5.3) it is found that if prediction grid cell's dimension $n = 100$ meter then the prediction accuracy is maximum. In other words, each of the prediction grid cells is of size 10^4 square meters.

6.4.5 GLIDW

The first approach that we propose is an MGRS based local spatial interpolation technique, called Grid based Local Inverse Distance Weighting (GLIDW). GLIDW predicts grid noise locally using a method that is inspired from the concept of Inverse Distance Weighted (IDW) interpolation. Recall from Section 6.4.4 that we develop grid based noise prediction, because predicting noise for every GPS coordinate is impractical.

Algorithm 1 describes the procedure of GLIDW. The inputs to this algorithm are sample noise data $\mathbf{S}$, radius R, a threshold N_{Th}, the dimension of each grid cell n, and a time interval $T_{pq} = [t_p, t_q]$. Initially, we divide the study area into grid cells having dimension $n \times n$. We create a set G that contains all such grid cells. Next, a subset $S_{pq} \subseteq \mathbf{S}$ of samples is created for which the sampling times are within the time interval $T_{pq} = [t_p, t_q]$, where t_p and t_q are the start and end of the time interval. From line 6 - line 10, we calculate distances $d(m_i, p_j)$ of the sampling location p_j of sample $s_j \in S_{pq}$ from the midpoint m_i of grid cell g_i and come up with a set D_i^n of all those distances. In line 11, we construct a set NN_i of sample points that are within radius R from the midpoint m_i. Next, in line 12, we check whether $|NN_i|$ is greater than or equal to some predefined threshold N_{Th}. If $|NN_i| \geq N_{Th}$, we consider only N_{Th} nearest neighbors for predicting grid cell's noise value. Later, in the result section (see Section 6.5.3) we will see the optimal value of N_{Th} such that we can calculate noise efficiently. From line 15 - line 22, we calculate the weights of N_{Th} closest sample points. Next, we predict noise value of grid cell g_i in line 23. But, if the condition mentioned in line 12 is not satisfied, then

it indicates that there are an insufficient number of sample points to predict the grid cell's noise value correctly, and eventually we put NULL as the noise value in line 25. At the end, the algorithm returns G along with the predicted noise values $N_{\mathsf{G}}^{(pred)}$.

Algorithm 6 GLIDW($\mathbf{S}$, R, N_{Th}, n, T_{pq})

Input: Sample data $\mathbf{S}$, Radius R, Number of neighbours N_{Th}, Dimension of each grid cell n, Time interval T_{pq}.
Output: Set of grid cells G, Grid cells' noise values $N_{\mathsf{G}}^{(pred)}$

1: Create a set of grid cells G having dimension $n \times n$;
2: $N_{\mathsf{G}}^{(pred)} \leftarrow \emptyset$; $S_{pq} \leftarrow$ samples of $\mathbf{S}$ having time interval $T_{pq} = [t_p, t_q]$
3: **for** each grid cell $g_i \in \mathsf{G}$ **do**
4: $m_i \leftarrow$ mid point of grid cell g_i;
5: $D_i^n \leftarrow \emptyset$;
6: **for** each sample $s_j \in S_{pq}, m_i \neq p_j$ **do**
7: //p_j is sampling location of sample s_j;
8: $d(m_i, p_j) \leftarrow$ distance between m_i and sample point p_j;
9: $D_i^n \leftarrow D_i^n \cup \{d(m_i, p_j)\}$;
10: **end for**
11: $NN_i = \{p_i^j \mid d(m_i, p_j) \leqslant R\ ,\ d(m_i, p_j) \in D_i^n\}$;
12: **if** $|NN_i| \geq N_{Th}$ **then**
13: Sort the sample points $p_i^j \in NN_i$ in order to create an ordered list $L_i = \langle p_i^{j_1}, p_i^{j_2}, \ldots, p_i^{j_{|NN_i|}} \rangle$ such that $d(m_i, p_{j_1}) \leq d(m_i, p_{j_2}) \leq \ldots \leq d(m_i, p_{j_{|NN_i|}})$;
14: Create a set $L_i^{'} = \{p_i^{j_k} \mid p_i^{j_k} \in L_i\ ,\ k \leq N_{Th}\}$;
15: **for** each neighbour $p_i^{j_k} \in L_i^{'}$ **do**
16: **if** $d(m_i, p_i^{j_k}) < 1$ **then**
17: $w_i^{j_k} = 1$;
18: **else**
19: $w_i^{j_k} = \left[\frac{1}{d(m_i, p_i^{j_k})} - \frac{1}{R}\right]$;
20: **end if**
21: $noise[p_i^{j_k}] \leftarrow$ noise value of sample $p_i^{j_k}$;
22: **end for**
23: $N_i^{(pred)} = \dfrac{\sum\limits_{p_i^{j_k} \in L_i^{'}} w_i^{j_k} \times noise[p_i^{j_k}]}{\sum\limits_{p_i^{j_k} \in L_i^{'}} w_i^{j_k}}$;
24: **else**
25: $N_i^{(pred)} = NULL$;
26: **end if**
27: $N_{\mathsf{G}}^{(pred)} = N_{\mathsf{G}}^{(pred)} \cup N_i^{(pred)}$;
28: **end for**
29: **return** $(\mathsf{G}, N_{\mathsf{G}}^{(pred)})$;

Algorithm GLIDW runs in time $O(|\mathsf{G}| n_s log(n_s))$ where, n_s and $|\mathsf{G}|$ are the total number of sample points and the total number of grid cells respectively. The outer for loop (line 3-line 28) iterates over $O(|\mathsf{G}|)$ times. For each iteration of the outer loop, the for loop of line 6-line 10 iterates over $O(n_s)$ times. Moreover, for each iteration of the outer loop, if the condition at line 12 is satisfied, then NN_i can contain n_s sample points in the worst case, and eventually sorting at line 13 can be performed in $O(n_s log(n_s))$ time in this case. The next for loop (line 15-line 22) iterates over $O(N_{Th})$ time, where, N_{Th} is a user defined threshold (in our case we set $N_{Th} = 10$). So overall time complexity is $O(|\mathsf{G}|(n_s + N_{Th} n_s log(n_s))) \approx O(|\mathsf{G}| n_s log(n_s))$. On the other hand, the space complexity of GLIDW is $O(|\mathsf{G}| + n_s)$. Set G contains all the grid cells which require $O(|\mathsf{G}|)$ spaces. To store the distances between a particular midpoint and the sample points we require $O(n_s)$ space (line 9). In the worst case,

NN_i takes $O(n_s)$ space (line 11). The set $L_i^{'}$ in line 14 takes $O(N_{Th})$ space. The set $N_{\mathsf{G}}^{(pred)}$ takes $O(|\mathsf{G}|)$ space as it needs to store the predicted noise values of $|\mathsf{G}|$ grid cells. The remaining variables take $O(1)$ spaces. So, overall space complexity is $O(|\mathsf{G}| + n_s + N_{Th}) \approx O(|\mathsf{G}| + n_s)$.

The main advantages of GLIDW are that it is simple to implement and it has a low computational time (i.e. low execution time). Note that, low execution time leads to better *response time* (execution time + communication time) if the method is deployed in the end-to-end framework (see Section 6.4.1). Despite these advantages, GLIDW suffers from some limitations. Firstly, it is sensitive to the missing values. In other words, it is unable to predict grid cell's noise if the grid cell does not contain sufficient samples. Hence, in the case of a *sparse dataset* (i.e. dataset having missing values), if the data sparsity level (Definition 4) is high, then it usually leads to many grid cells to remain unpredicted. Secondly, it is a local method for predicting noise and there is no direct way to predict noise value of a grid cell that has an insufficient number of samples. This may severely affect the performance of GLIDW and eventually, *high accuracy* may not always be achieved. Next, we propose two different methods (I-GLIDW and GLIDW-OK) in order to overcome the above-mentioned limitations regarding the sparse dataset and accuracy. Each of these two methods has its own advantages and disadvantages.

Definition 4 *(Data sparsity level) The data sparsity level δ of a dataset can be defined as follows:*

$$\delta = \left[1 - \frac{|\mathsf{G}^{(ns)}|}{|\mathsf{G}|}\right] \times 100\% \tag{6.7}$$

where, $\mathsf{G}^{(ns)}$ is the set of grid cells containing noise samples and G is the set of all grid cells. For example, if data sparsity level of a dataset $\delta = x\%$, then it means $x\%$ of the grid cells having missing values.

6.4.6 I-GLIDW

I-GLIDW (Iterative GLIDW) is an iterative form of GLIDW that is proposed to overcome the limitations of GLIDW. In this method of interpolation, the local effect of grid cells are propagated to neighbouring grid cells in an iterative manner as described in Algorithm 2.

The inputs to I-GLIDW are sample noise data, radius for GLIDW, radius for I-GLIDW, threshold G_{Th}, and the dimension of each grid cell. The first step of this method is to compute the grid cells' noise that has a sufficient number of samples using GLIDW. In the first iteration, for each grid cell $g_i \in \mathsf{G}$, if the noise value of that cell is not predicted yet (line 10), then we calculate the fraction η of neighbouring grid cells having already predicted noise values within radius $R^{'}$ (line 12). Next, we check whether η is greater than or equal to some predefined threshold G_{Th}. If so, we compute weights w_i^j for each neighbour $g_j \in GN_i$, where GN_i is a set of all nearest neighbours of grid cell g_i having already predicted noise values. Based on the weights w_i^j and corresponding noise value $N_j^{(pred)}$ of each grid cell $g_j \in GN_i$, we compute the predicted noise value $N_i^{(pred)}$ of g_i (line 18). But, if η is less than G_{Th}, we

Algorithm 7 I-GLIDW($\mathbf{S}$, R, N_{Th}, n, R', G_{Th}, T_{pq})

Input: Sample data $\mathbf{S}$, Radius R for GLIDW, Number of neighbours N_{Th} for GLIDW, Dimension of each grid cell $n \times n$, Radius R' for iterative computation, Threshold G_{Th}, Time interval T_{pq}.

Output: Set of grid cells G, Grid cells' noise value $N_{\mathsf{G}}^{(pred)}$

1: $(\mathsf{G}, N_{\mathsf{G}}^{(pred)}) \leftarrow GLIDW(\mathbf{S}, R, N_{Th}, n, T_{pq})$;
2: $GN_{R'}^{(total)} \leftarrow$ Total number of neighbours within R';
3: $C = |\mathsf{G}|$;
4: **while** $C \neq 0$ **do**
5: **if** $|\mathsf{G}| - C \neq 0$ **then**
6: $C = |\mathsf{G}|$;
7: **end if**
8: **for** each grid cell $g_i \in \mathsf{G}$ **do**
9: $C \leftarrow C - 1$;
10: **if** $N_i^{(pred)} == NULL$ **then**
11: Construct a set $GN_i = \{g_j \mid N_j^{(pred)} \neq NULL\ ,\ i \neq j\ ,\ d(m_i, m_j) \leqslant R'\}$;
12: $\eta = \frac{|GN_i|}{GN_{R'}^{(total)}}$;
13: **if** $\eta \geq G_{Th}$ **then**
14: $N_{\mathsf{G}}^{(pred)} = N_{\mathsf{G}}^{(pred)} \setminus \{N_i^{(pred)}\}$;
15: **for** each neighbour $g_j \in GN_i$ **do**
16: $w_i^j = \left[\frac{1}{d(m_i, m_j)} - \frac{1}{R'}\right]$;
17: **end for**
18: $N_i^{(pred)} = \frac{\sum_{g_j \in GN_i} w_i^j \times N_j^{(pred)}}{\sum_{g_j \in GN_i} w_i^j}$;
19: $N_{\mathsf{G}}^{(pred)} = N_{\mathsf{G}}^{(pred)} \cup \{N_i^{(pred)}\}$;
20: **else**
21: $N_i^{(pred)} = NULL$;
22: $C \leftarrow C + 1$;
23: **end if**
24: **end if**
25: **end for**
26: **end while**
27: **return** $(\mathsf{G}, N_i^{(pred)})$;

assign predicted noise value $N_i^{(pred)}$ as NULL (line 21). If, at the end of an iteration all the grid noise values are not computed (line 4), then we go into the next iteration along with the noise values computed in the previous iteration. This process is continued until all the grid cells' noise values are computed. The main objective of this approach is to propagate grid cell's noise to its neighbours.

Algorithm I-GLIDW runs in time $O(|\mathsf{G}|(n_s log(n_s) + I.GN_{R'}^{(total)}))$. Let us consider the while loop iterates over I times. The for loop of line 8 iterates over $O(|\mathsf{G}|)$ times and the for loop at line 15 iterates over $O(GN_{R'}^{(total)})$ times in worst case. Hence, overall time complexity = Complexity due to GLIDW (line 1) + $O(I|\mathsf{G}|GN_{R'}^{(total)}) \approx O(|\mathsf{G}|n_s log(n_s)) + O(I|\mathsf{G}|GN_{R'}^{(total)}) \approx O(|\mathsf{G}|(n_s log(n_s) + I.GN_{R'}^{(total)}))$. On the other hand, there can be maximum of $GN_{R'}^{(total)}$ neighbours in GN_i. Hence, it requires $O(GN_{R'}^{(total)})$ space in worst case. Remaining variables take $O(1)$ spaces. Therefore, the overall space complexity of this algorithm is = space required for GLIDW + $O(GN_{R'}^{(total)}) \approx O(|\mathsf{G}| + n_s + GN_{R'}^{(total)})$

Fig. 6.3 shows the process of I-GLIDW. In Fig. 6.3(a) we predict the noise of those grid cells that have a sufficient number of noise samples. Fig. 6.3(b) shows a particular grid cell

with noise samples. We compute the grid cell's noise if it has a sufficient number of noise samples using GLIDW. Here, $R = \frac{1}{2} \times Length\ of\ diagonal\ = \frac{1}{2} \times \sqrt{2} \times n = \frac{n}{\sqrt{2}}$ is the radius for GLIDW, and we consider only those noise samples that are within the radius. Fig. 6.3(c) and (d) show a single iteration of I-GLIDW to compute noise values from already predicted grid cells' noise. As shown in Fig. 6.3(d), the total number of neighbours (i.e. neighbouring grid cells' midpoints) within radius $R^{'}$ is 8 (i.e. $GN_{R^{'}}^{(total)} = 8$), as the figure indicates the value of $R^{'} = \sqrt{2} \times n$, where n is the grid cell's dimension. We can generalize this as follows. Suppose there exist N number of grid cells in the row and M number of grid cells in the column within any user defined $R^{'}$, then the value of $GN_{R^{'}}^{(total)}$ will be $N \times M - 1$. It should be noted that the threshold value is assumed to be $G_{Th} = \frac{1}{N \times M - 1}$. This assumption allows us to calculate all the missing grid cells' values in I-GLIDW even in the worst case (i.e when the data sparsity is maximum). A large number of iterations may be required to predict all the grid cells' noise for a sparse dataset. Note that although I-GLIDW can handle the problem related to the sparse dataset, a very high data sparsity level can affect its predictive accuracy (see Section 6.5.4).

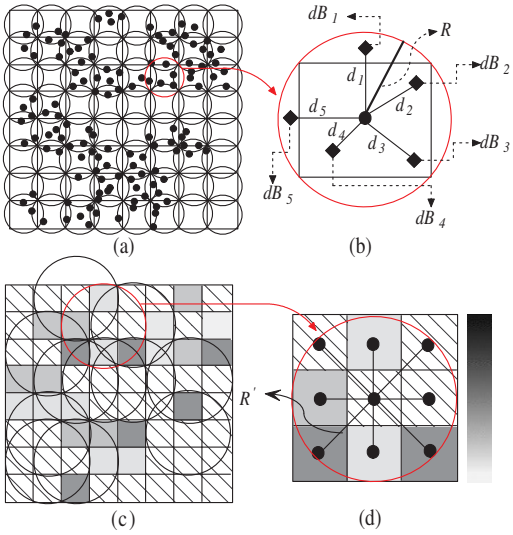


FIGURE 6.3: Process of I-GLIDW (a) Prediction of grid cells' noise having a sufficient number of noise samples. Small dots represent sample points. The samples within each circle are responsible to find the noise value of the grid cell within that circle (b) A particular grid cell is zoomed in. dB_i are the known noise values at sample points (square dots) having distances d_i for $i = \{1,2,3,4,5\}$ from the midpoint (circular dot) of the grid cell. (c) First iteration of I-GLIDW (d) Prediction of an unpredicted grid cell's noise using already predicted neighbors (shaded grid cells) in an iteration

6.4.7 GLIDW-OK

This method is based on both GLIDW and Ordinary Kriging (OK) methodologies. The main idea behind this method is to analyze the study area in both local and global views of the grid. Initially, we analyze grid cells locally by inferring the noise based on the sample data local to the grid cells (in our case, we are considering sample points within a predefined radius from the midpoints of the grid cells). On the other hand, when we are analyzing the grid globally, it means multiple grid cells (already predicted) are taking part in inferring the noise value of

an unpredicted grid cell. For local analysis, we apply GLIDW and eventually compute those grid cells' noise value which has a sufficient amount of sample data. The result of GLIDW is then used as input for the Ordinary Kriging (OK) method. For global analysis, the ordinary kriging method is used to predict grid cells' noise values that are not predicted yet.

Algorithm 3 describes the procedure of GLIDW-OK. Initially, we call GLIDW (line 1) to compute the grid cells' noise values that have a sufficient number of sample data points. Next, we are creating an experimental semivariogram using already predicted noise values. The experimental semivariogram is calculated using Eq. 6.8.

$$\widehat{\gamma}(h) = \frac{1}{2N(h)} \sum_{N(h)} (Z_i - Z_j)^2 \tag{6.8}$$

Where, h is specified lag distance, z_i and z_j are two points that are separated spatially by h. The term $N(h)$ is the total number of points that are separated by the distance h. The semivariogram is the sum of squared differences between values separated by a distance h. After that, we are choosing a model that is the best fit for the experimental semivariogram. Next, we use the Ordinary Kriging method to predict the remaining grid cells' urban noise value by using already predicted noise and selected semivariogram model.

Standard ordinary kriging at line 4 takes $O(|\mathsf{G}|n_s^3)$ time and $O(|\mathsf{G}|n_s^2)$ space [243]. So, overall time required for GLIDW-OK is = time required for GLIDW (line 1) + $O(|\mathsf{G}|n_s^3) \approx O(|\mathsf{G}|n_s^3)$. On the other hand, the overall space required for GLIDW-OK is = space required for GLIDW (line 1) + $O(|\mathsf{G}|n_s^2) \approx O(|\mathsf{G}|n_s^2)$.

Algorithm 8 GLIDW-OK(**S**, R, N_{Th}, n , V^m, T_{pq})

Input: Sample data **S**, Radius R for GLIDW, Number of neighbours N_{Th} for GLIDW, Dimension of each grid cell $n \times n$, V^m is the set of variogram models, Time interval T_{pq}.

Output: Set of grid cells G, Grid cells' noise value $N_{\mathsf{G}}^{(pred')}$.

1: $(\mathsf{G}, N_{\mathsf{G}}^{(pred)}) \leftarrow GLIDW($ **S**, R, N_{Th}, n, $T_{pq})$
2: Generate Semivariogram of data set using equation 7.
3: $model \leftarrow Modelling(V^m)$.
4: $(\mathsf{G}, N_{\mathsf{G}}^{(pred')}) = Ordinary_Kriging(\mathsf{G}, N_{\mathsf{G}}^{(pred)}, model)$
5: **return** $(\mathsf{G}, N_{\mathsf{G}}^{(pred')})$;

6.4.8 Performance metrics

We use 10-fold cross validation [244] method in order to validate the simulation accuracy for comparing the performance of our proposed spatial interpolation methods with some baselines (e.g. IDW, OK). Some common statistical measures like mean absolute error (MAE), relative mean absolute error (RMAE), mean relative error (MRE), root mean squared errors (RMSE) are used to evaluate spatial prediction accuracy and model simulation effect. MAE, RMAE, MRE, RMSE are defined as follows:

$$MAE = \frac{1}{k} \sum_{i=1}^{k} |z(x_i) - \widehat{z}(x_i)| \tag{6.9}$$

$$RMAE = \frac{1}{k}\sum_{i=1}^{k}\frac{|z(x_i) - \widehat{z}(x_i)|}{z(x_i)} \tag{6.10}$$

$$MRE = \frac{1}{k}\sum_{i=1}^{k}|z(x_i) - \widehat{z}(x_i)/z(x_i)| \tag{6.11}$$

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\{z(x_i) - \widehat{z}(x_i)\}^2} \tag{6.12}$$

Where, k represents the total number of samples in the validation set, $z(x_i)$ and $\widehat{z}(x_i)$ represent the measured and predicted noise values respectively. MAE, RMAE, MRE, and RMSE are used to assess the prediction accuracy of interpolation methods. Smaller values of MAE, RMAE, MRE, and RMSE represent greater accuracy in the prediction result. We also measure the relative improvement (RI) of our proposed spatial noise prediction models with respect to the baselines (e.g. IDW, OK). For a given model M_i relative improvement (RI) of RMSE with respect to another model M_j can be calculated using Eq 6.13.

$$RI^{(RMSE)}_{M_i/M_j} = \frac{RMSE^{M_j} - RMSE^{M_i}}{RMSE^{M_j}} \tag{6.13}$$

where, $RMSE^{M_i}$ and $RMSE^{M_j}$ are RMSE values of model M_i and M_j respectively. If $RMSE^{M_i}$ is smaller than $RMSE^{M_j}$, the RI will be positive, which indicates M_i is a better model than M_j.

6.5 Experiment and results

In this section, we discuss our study area and the data collection, calibration, parameter and model selection and at the end, we present the results and comparison. In this study, all the experiments are performed on a 64-bit Windows PC with 3.30 GHz Intel(R) Xeon(R) CPU E3-1226 v3 processor and 64 GigaByte of RAM.

6.5.1 Study Area and Data Collection

The study was conducted in the Jadavpur area which is extended between $22°48'91''N - 22°50'70''N$ latitude and $88°36'76''E - 88°40'07''E$ longitude covering an area of 7 km^2 which is a part of city Kolkata. The region is densely populated, with different land use/land cover. Fig. 6.4 shows the study area. A total of 26 participants having diverse demographic characteristics (e.g. occupation, age, gender, etc.) are dedicatedly involved in the data collection process for a period of 2 years (June 2017-June 2019). Here, all the 26 participants ($\approx$ 35% female and $\approx$ 65% male) are asked to collect the noise pollution data within the above-mentioned study area with GPS-enabled smartphones. All the participants live in the city of Kolkata, India. The demographic data of these participants are presented in Fig. 6.5. Over half of the participants ($\approx$ 65%) are undergraduate (UG) and postgraduate (PG) students,

approximately 13% of them are employees, and the rest of the others ($\approx$ 25%) are research scholars. Moreover, approximately 13% of the participants have age less than 20 years, 70% of the participants have age between 20 to 30 years, and 17% of the participants have age greater than 30 years.

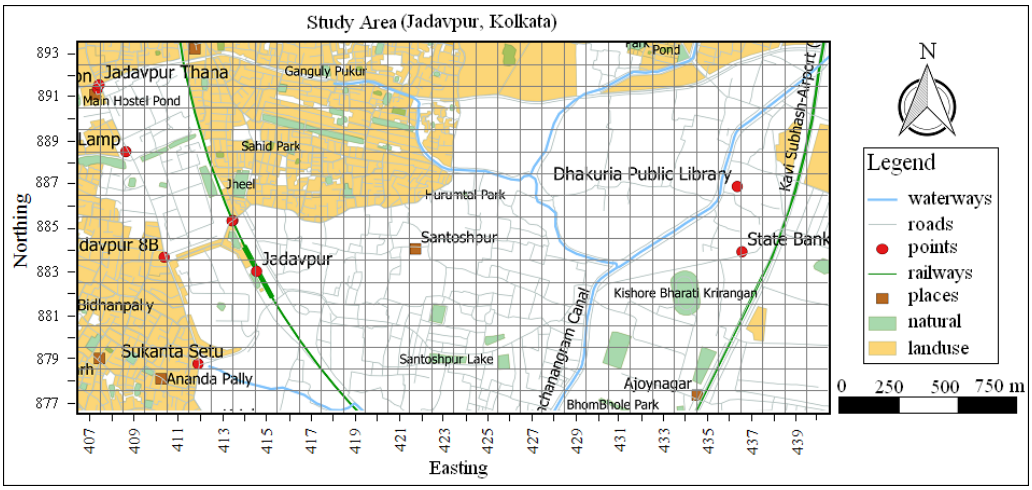


FIGURE 6.4: Map of our study area. This map is created using QGIS. The study area is part of city Kolkata, India. Here, Easting and Northing are East-West and North-South positions respectively of MGRS coordinate value.

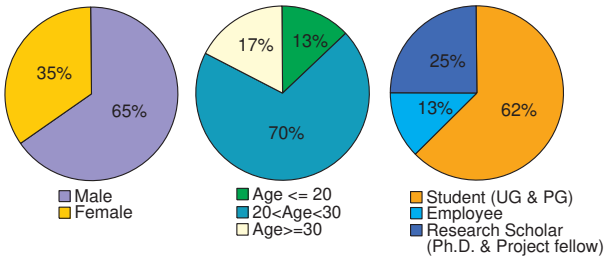


FIGURE 6.5: Demographic data of participants.

The data collection required participants to install our application NoiseSense [104] for Android. In the NoiseSense application, data gets recorded at an interval of 5 seconds in a database in the format $< user_id, sm, si, lat, lngt, ts, L_{eq}, acc_x, acc_y, acc_z, prx >$. The user interface of the application NoiseSense for logging ambient noise data is shown in Fig. 6.7 (a). Any smartphone which is calibrated with the standard sound level meter (SLM) can be used for data collection at any point of time. Although, in the experiment, we have used the same set of smartphones for the entire data collection process. Here, the noise pollution data samples are mainly collected at three different sessions (Morning Session (MS), Afternoon Session (AS), and Evening Session (ES)) of different days. We consider MS, AS, and ES as the time intervals [9:00AM-11:00AM], [3:00PM-5:00PM], and [6:00PM-8:00PM] respectively. Here, our main motive behind selecting those time intervals for data collection is to generate noise pollution maps for the rush hours in the city. After context-aware data cleaning (see Section 6.4.3), there are a total of 93,786 data samples present in the participatory sensing based noise pollution dataset. Fig. 6.6 depicts the spatial distribution of sampling

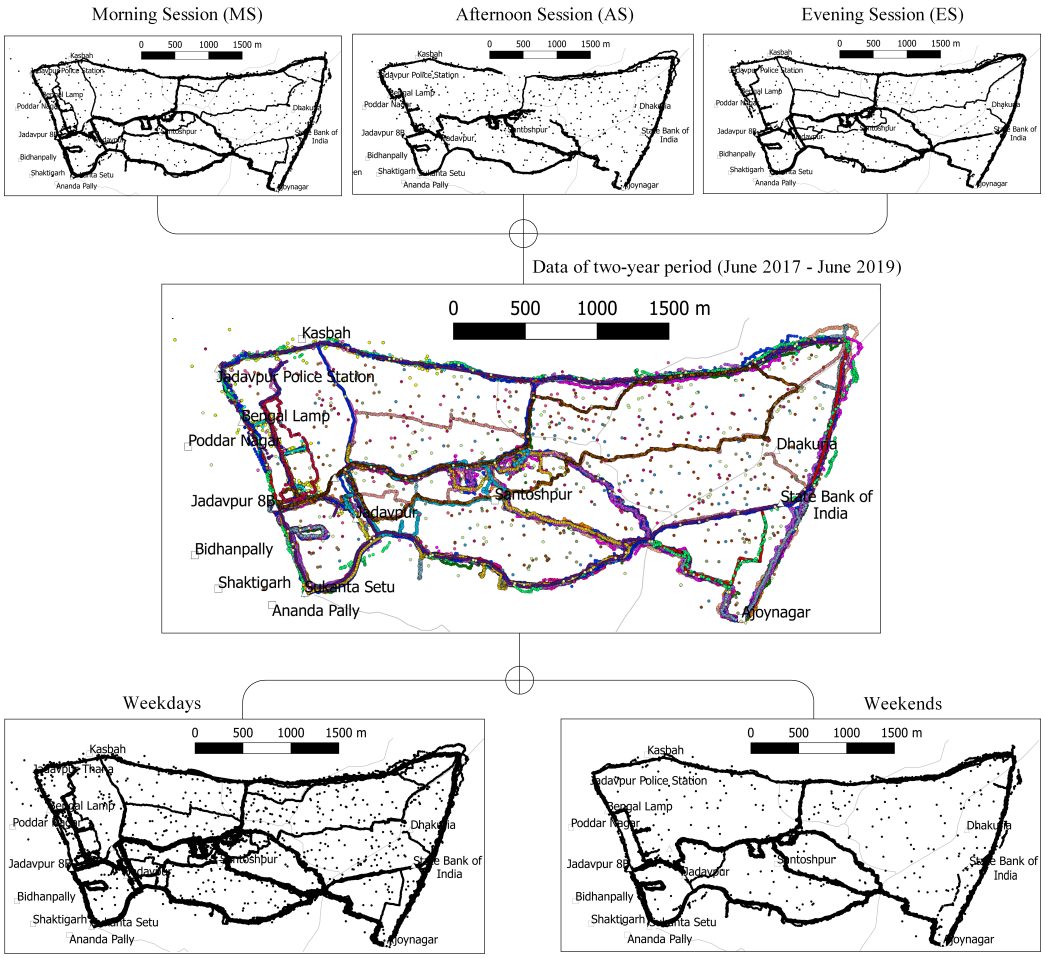


FIGURE 6.6: Spatial distribution of the sampling locations in the study area for the morning session (MS), afternoon session (AS), evening session (ES), data collected over the two-year period, weekdays, and weekends. Different colors in the map indicate sampling locations corresponding to different participants.

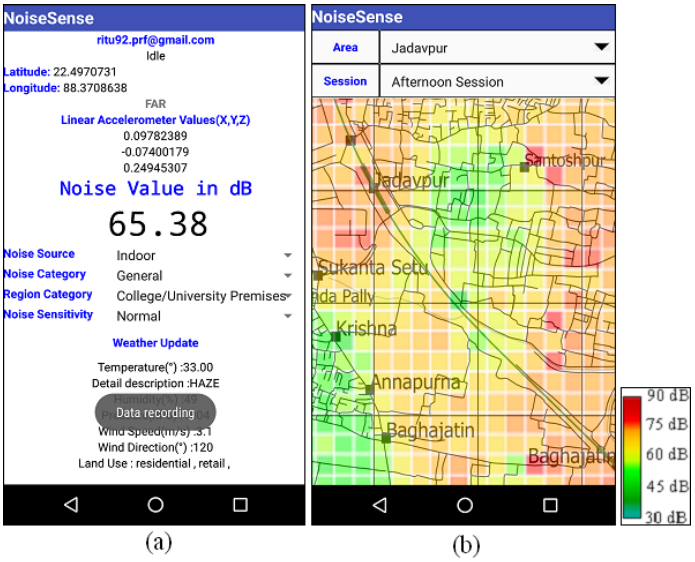


FIGURE 6.7: (a) Screenshot of data logging activity using our Android application (NoiseSense). (b) Noise pollution map as a service.

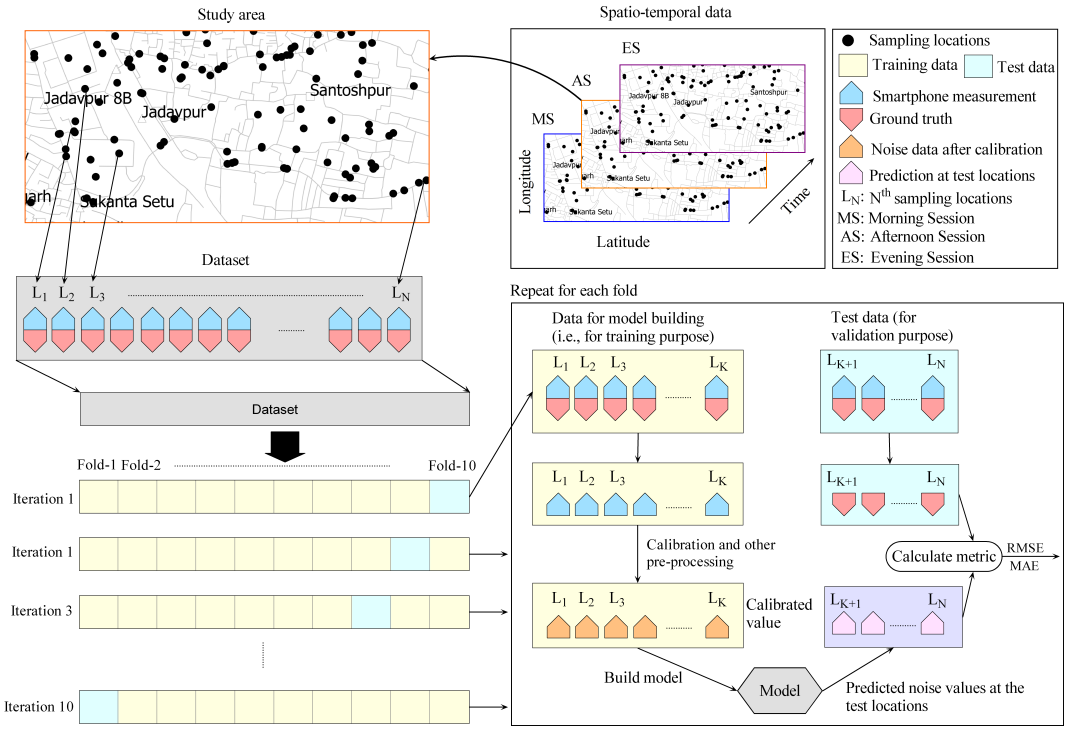


FIGURE 6.8: The details of the soatio-temporal data and the approach for evaluation of the spatial noise mapping techniques.

locations in the study area. Note that the data sparsity levels (δ) for MS, AS, and ES sessions are 5.1%, 6.2%, and 4.9% respectively. Whereas, combining all sessions' data reduces the sparsity levels to 2.3%. For ground truth data collection we use standard noise measuring devices like Amprobe SM10 [245] and Mextech SL-4012 sound level meter [246]. In this work, the participants carry both standard device (for ground truth data) and smartphones (for participatory sensing data) during data collection, they collect ground truth data in the same locations where data are collected using smartphones.

The data that are collected using ground truth devices are used to validate the proposed predictive models. The details of the spatio-temporal aspects of the data along with our approach for the evaluation of spatial noise mapping are illustrated using Fig. 6.8. Note that we have used 10-fold cross validation (CV) for the evaluation. As shown in Fig. 6.8, for each iteration of the CV, the data from the spatial locations for model building (i.e., training purpose) and validation (i.e., testing purpose) are disjoint sets. Hence, the ground truth data in the test set (to which the predicted values are compared) are in different spatial locations from those data that are used for the model building. The noise values are predicted on the test locations after the model is built, and then the predicted values are compared to the ground truths at those locations (i.e., test locations) to measure the metric.

6.5.2 Calibration

As we have discussed in Section 6.4.2, we need to estimate the calibration offset ξ in order to correctly measure L_{eq}. Calibration is necessary, it is because, without good calibration, the smartphone sensors may produce data that are inaccurate. Now, to calculate calibration

offset ξ, we measure and compare the readings from SLM (Sound Level Meter) and different smartphones under different sound levels. For example, the scatter plot for SLM readings and their corresponding readings of smartphones are shown in Fig. 6.9 for three smartphone models (Google Pixel, Moto Z2 Play and Panasonic Eluga). Moreover, we use linear regression (least squares fitting technique) to show that there exists a linear relationship between smartphone noise level measurements and true noise levels (i.e. SLM readings). For different smartphone models, we find different linear models and calculate the metrics of goodness of fit to show the quality of linear fittings. These linear models and their goodness of fit for some smartphone models are shown in Table 6.3. Suppose, for any smartphone model sm_i, the linear model is given by Eq. 6.14, where x is the smartphone noise reading. Now, for any noise reading L_{eq} in a smartphone model sm_i, the corresponding calibration offset is calculated by Eq. 6.15. We observe that for different smartphone models, the calibration offset not only varies from phone to phone but also may vary under different sound levels.

$$f^{sm_i}(x) = p_1 x + p_2 \tag{6.14}$$

$$\xi^{sm_i}(L_{eq}) = f^{sm_i}(L_{eq}) - L_{eq} \tag{6.15}$$

TABLE 6.3: Linear Models and Quality of Linear Regression

Smartphone Device	Linear Model $[f(x) = p_1 x + p_2]$		Quality of Linear Regression [Goodness of fit]			
	p_1	p_2	SSE	R^2	Adjusted R^2	RMSE
Moto Z2 Play	1.073	20.52	1004	0.9653	0.9651	2.845
Panasonic Eluga	1.225	-1.89	1059	0.9397	0.9391	3.27
Google Pixel	1.023	26.55	756.2	0.9713	0.9711	2.47
Lenovo K3 Note	0.969	11.61	760	0.9595	0.9592	2.571
Lava Pixel v3	0.9412	13.88	101.8	0.9843	0.9841	1.302
Redmi 3S Prime	1.013	18.99	338.4	0.9693	0.9688	2.395
Galaxy J7 Nxt	0.9448	16.46	321.7	0.9755	0.9752	2.83
Galaxy J7 Prime	1.0396	15.74	216.1	0.9715	0.9713	2.55

6.5.3 Parameters and model selection

Two parameters N_{Th} and n of GLIDW need to be known in order to predict noise within a grid locally. We change the values of the threshold N_{Th} as well as the dimension of the grid cell n to find an optimal combination of them, so that this optimizes our prediction accuracy. Fig. 6.10 shows the RMSE values for different settings of N_{Th} and n. It can be seen that, if n is in range $50 \sim 80$ meters and N_{Th} is around 10, then the lowest RMSE is achieved, but the problem is, a small cell dimension like $50 \sim 80$ meters, will cause few noise samples within a grid cell which leads to many grid cells to remain unpredicted. It should be noted that,

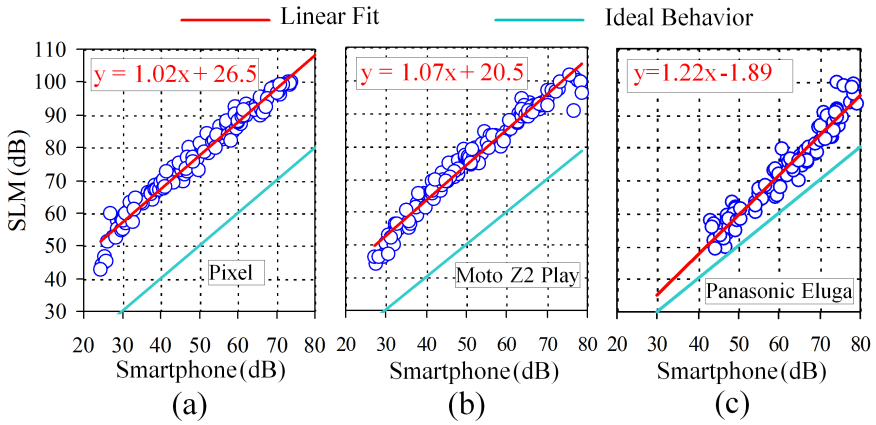


FIGURE 6.9: Scatter plots of SLM readings and cottesponding smartphone readings for three different smartphomes (a) Google Pixel (b) Moto Z2 Play (c) Panasonic Eluga. A regression line (red line) shows linear relationship between smartphone noise level measurements and true noise levels. The difference between regression line (red line) and ideal line (cyan) gives calibration offset ξ.

if n is in the range $80 \sim 100$ meters and N_{Th} is around 10, then also we have got a decent result. In our work, we have taken $n = 100$ (i.e. 100×100 grid cells) and $N_{Th} = 10$. Note that the value of radius R in GLIDW depends on the cell dimension n. As shown in Fig. 6.3(b), the value of R is half the length of the diagonal of a grid cell. Hence, in this case, $R = \frac{1}{2} \times Length\ of\ diagonal\ = \frac{1}{2} \times \sqrt{2} \times n = \frac{n}{\sqrt{2}} = \frac{100}{\sqrt{2}}$. It is also important to note that the parameter selection here is performed based on the whole dataset.

For GLIDW-OK, Fig. 6.11 shows the fitting of different models to the semivariogram generated from the sample data. We select the spherical model which is the best fit compared to other models like exponential model, linear model, etc. The parameters of the model are: nugget variance (C_o)= 12.354196 , sill (C)= 77.422 and range (a)= 8.633. Hence, the fitted model is as follows:

$$\gamma(h) = 12.354196 + 77.422 \times \left(\frac{3h}{17.266} - \frac{1}{2} \times \left(\frac{h}{8.633} \right)^3 \right) \tag{6.16}$$

In the case of the baseline IDW, the best value of the power parameter p is found out by testing IDW with different values of p. Note that $p = 2$ is selected as it provides the best performance. On the other hand, spherical variogram is used as a variogram model in case of the baseline OK as it is found to be the best fit model as compared to other models like exponential model, linear model, etc.

6.5.4 Results

Each of the three methods (GLIDW, I-GLIDW, and GLIDW-OK) that we have proposed produces a fairly different noise pollution map on participatory sensing data of our study area. As shown in Fig. 6.12, GLIDW produces a noise map of our study area which is based on afternoon session data (3:00 PM-5:00 PM). The figure reflects that a lot of grid cells remain unpredicted by GLIDW. Recall from Section 6.4.5, this is one of the limitations of GLIDW. Fig. 6.13, shows the noise maps produced at different iterations (total 15 iterations, I_1-I_{15}) by I-GLIDW. Two limitations of I-GLIDW can be identified clearly from this figure

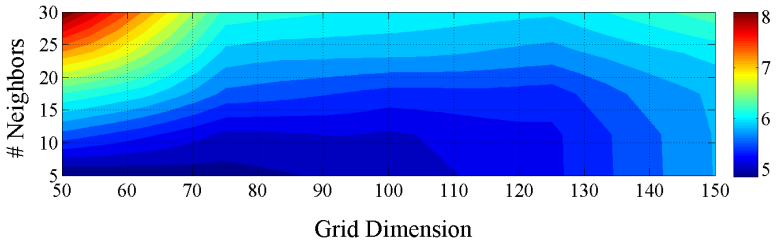


FIGURE 6.10: Prediction accuracy (RMSE values) for different combinations of neighboring threshold N_{Th} and cell dimension n.

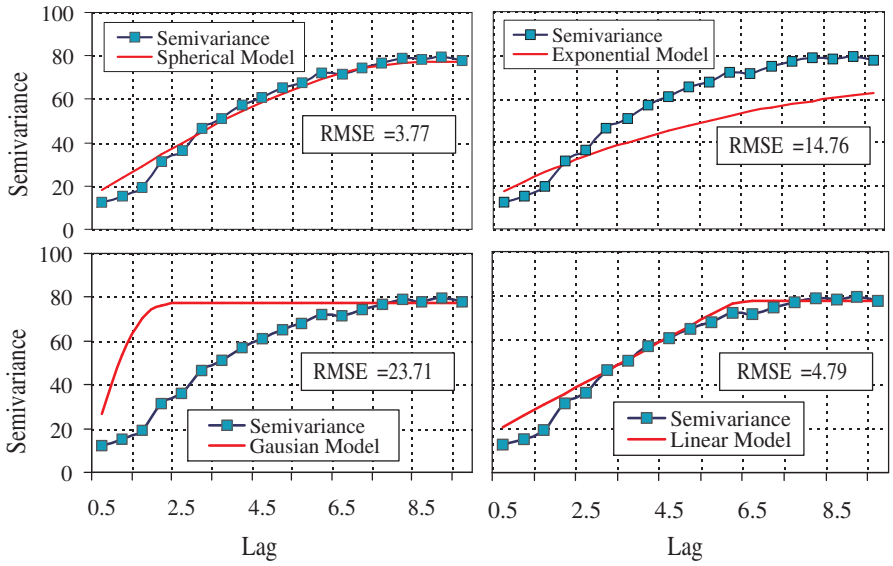


FIGURE 6.11: Fitting of four commonly used semivariogram models (Spherical, Exponential, Gaussian and Linear) to the experimental semivariogram

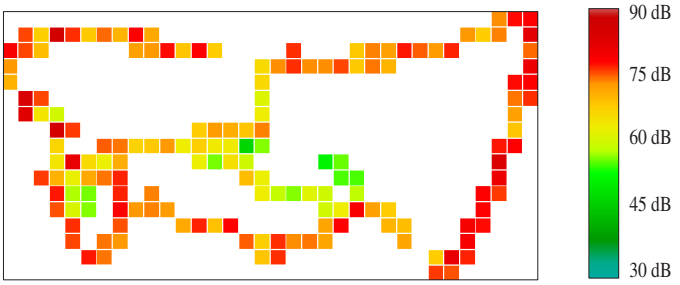


FIGURE 6.12: Noise map produced by method GLIDW. Grid cells having an insufficient number of sample points remain unpredicted (white-colored regions).

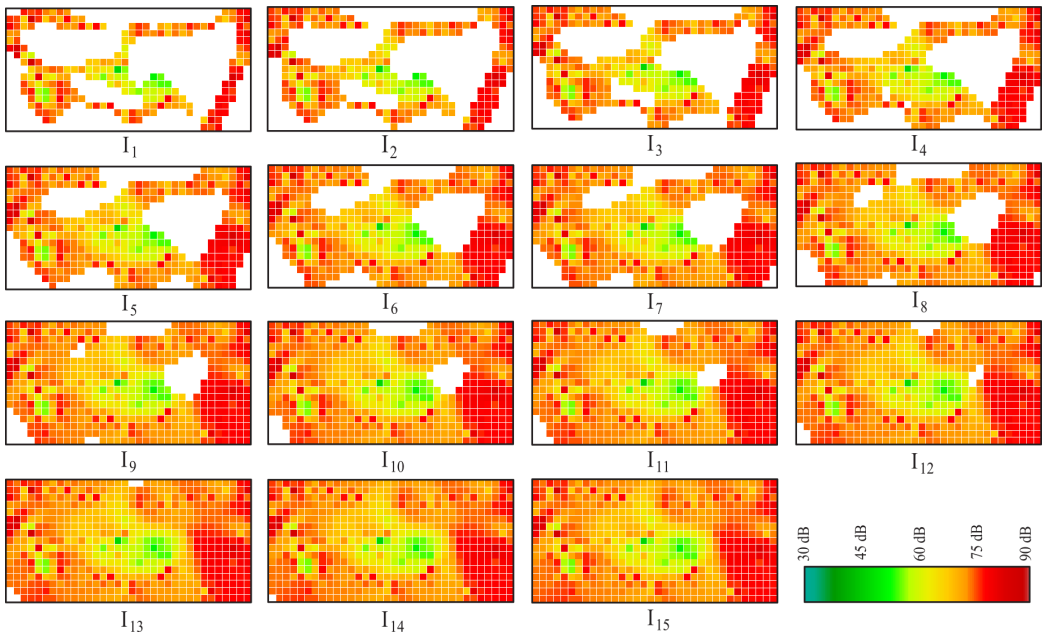


FIGURE 6.13: Noise maps produced in different iterations ($I_1 - I_{15}$) by method I-GLIDW

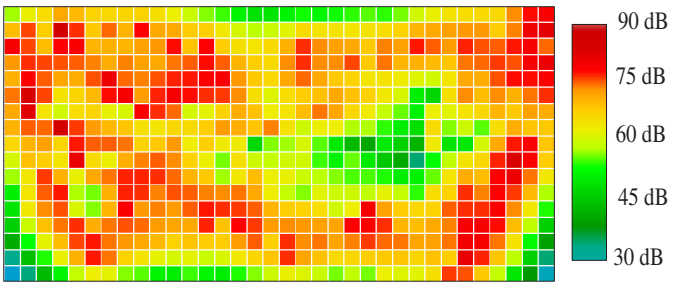


FIGURE 6.14: Noise map produced by method GLIDW-OK

(Fig. 6.13). We can see at the bottom left portion of the noise map of iteration I_1, there is an area with high noise values (deep red) surrounded by unpredicted grid cells having missing values. As iterations go on, noise values of those unpredicted grid cells become predicted by I-GLIDW. If we carefully see different iterations, I-GLIDW propagates the initial noise values (i.e. the deep red portion) to the unpredicted grid cells. This is the first limitation of I-GLIDW. Moreover, we can see, to generate the final noise map, I-GLIDW takes too many iterations (in this case 15, it can be even more for a different distribution of sample points), which increases its computational cost. Now, Fig. 6.14 shows the noise map produced by GLIDW-OK on the same data set. Later, we evaluate these methods in terms of different performance metrics.

Performance Comparison with baselines

We compare our proposed spatial interpolation methods with two baselines. The first one is Inverse Distance Weighting (IDW) [133] and the second one is Ordinary Kriging (OK) [235]. The cross-validation results of noise levels estimation using GLIDW, I-GLIDW, GLIDW-OK, IDW, and OK are shown in Table 6.4. The results show that there is not a single method

TABLE 6.4: Performance comparison with baselines

Session	Methods	Quality of Models			
		MAE	RMAE	MRE	RMSE
MS (9:00AM-11:00AM)	GLIDW	8.1094	0.1099	73.1977	9.6156
	I-GLIDW	6.1134	0.1059	72.4287	7.3737
	GLIDW-OK	5.8477	0.1028	72.4512	7.1441
	IDW	6.4452	0.1063	72.4171	7.8882
	OK	6.0316	0.1053	72.8635	7.2423
AS (3:00PM-5:00PM)	GLIDW	6.9134	0.0992	70.5698	8.4805
	I-GLIDW	5.9648	0.0921	66.9346	7.4226
	GLIDW-OK	6.3137	0.0935	66.9213	7.8568
	IDW	6.3707	0.0956	66.9404	8.4793
	OK	6.3353	0.0943	66.9923	7.9312
ES (6:00PM-8:00PM)	GLIDW	9.5457	0.143	75.4855	10.6806
	I-GLIDW	5.1884	0.0792	72.8963	6.9227
	GLIDW-OK	6.3966	0.1019	72.9158	7.2958
	IDW	7.1925	0.1063	72.8644	8.9272
	OK	6.5141	0.1043	72.4623	7.9513

MS: Morning Session; AS: Afternoon Session; ES: Evening Session; GLIDW: Grid Based Local Inverse Distance Weighting; I-GLIDW: Iterative Inverse Distance Weighting; GLIDW-OK: Grid Based Local Inverse Distance Weighting- Ordinary Kriging; IDW: Inverse Distance Weighting; OK: Ordinary Kriging

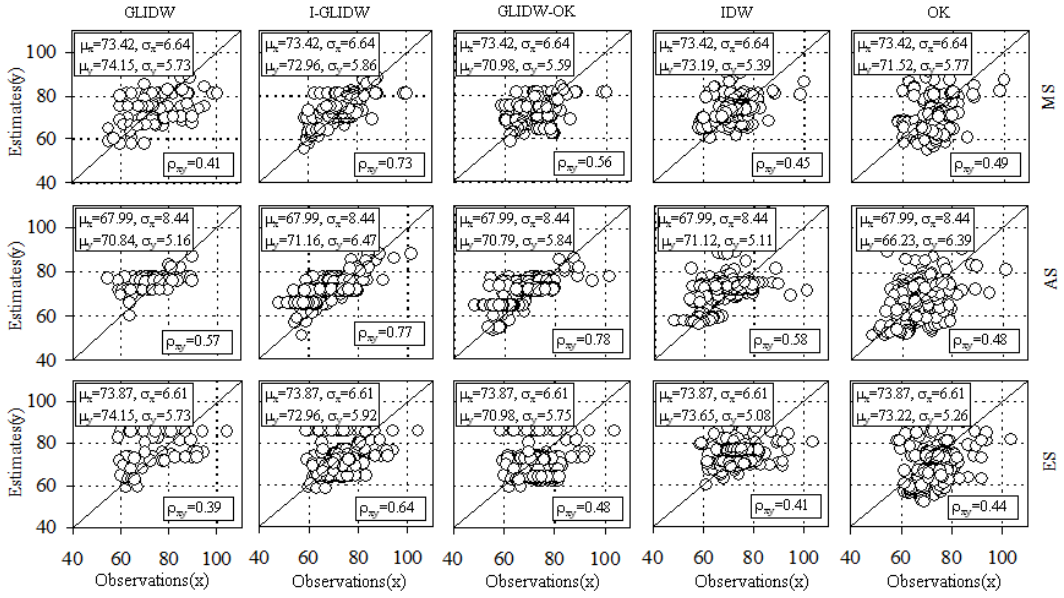


FIGURE 6.15: Observation vs Prediction plots of different methods (GLIDW, I-GLIDW, GLIDW-OK, IDW, OK) in different sessions (MS-Morning Session, AF-Afternoon Session, ES-Evening Session) of a day

that clearly dominates over other methods in terms of predictive accuracy (MAE, RMAE, RME, RMSE) in all three sessions (e.g. Morning Session (MS), Afternoon Session (AS), Evening Session (ES)) of a day. This is because datasets of different sessions may have different data sparsity levels and the impact of data sparsity on the predictive accuracy of different methods may not be the same (see Section 6.5.4). The highest error is obtained with GLIDW, irrespective of the sessions. We have also shown the scatter plots (see Figure 6.15) of the original noise levels (observations) versus the predicted noise levels (estimates) from GLIDW, I-GLIDW, GLIDW-OK, IDW, and OK in different sessions of a day. In all the methods, the estimated mean noise (μ_y) is almost similar to the observed mean noise (μ_x)(i.e. actual mean). We can also see, estimated standard deviation (σ_y) is smaller than the observed standard deviation (σ_x) (smoothness). Additionally, the correlation coefficients (ρ_{xy}) between observations (x) and estimates (y) are fairly good for both I-GLIDW and GLIDW-OK in all three sessions. The method IDW smoothed out the prediction of noise levels at the most. On the other hand, the smoothing effect of prediction is the least for I-GLIDW. It is also important to note that, for each of the MS, AS, and ES rows of Fig. 6.15, the distribution of observations for GLIDW is different from the rest of the other methods. It is because, as discussed in Section 6.4.5, one of the major limitations of GLIDW is that, it is unable to predict grid cells' noise if the grid cells do not contain a sufficient number of noise samples. In other words, if for any grid cell g_i, the condition $NN_i < N_{Th}$ holds true in Algorithm 1 then GLIDW returns NULL, where NN_i is the number of nearest neighbour samples within radius R. Hence, in the case of GLIDW, some of the predicted values could be NULL. Whereas other methods (I-GLIDW, GLIDW-OK, IDW, OK) always come up with a non-NULL predicted value. Now, Fig. 6.16 (a), (b), and (c) compare the execution time (T_E) and communication time (T_c) of the proposed methods with the baselines by varying the number of urban noise samples ($n_s \approx 4000$, $n_s \approx 8000$, $n_s \approx 12000$). Here, the execution time T_E of OK is found to be maximum compared to the execution time of other approaches. However, the communication time (T_c) doesn't vary significantly across the methods for the same number of samples. We also show the relative improvement between each pair of methods in the form of three heatmaps (see Fig. 6.17) one for each session (MS, AS, ES). We can clearly see from Fig. 6.17 that, the relative improvements are fairly good for I-GLIDW and GLIDW-OK.

The features of the spatial interpolation techniques that we have proposed may vary from one technique to another. Hence, we provide a theoretical comparison among GLIDW, I-GLIDW, GLIDW-OK, IDW and OK in Table 6.5.

Discussions and observations

The predictive accuracy of different methods depends on how sparse the dataset is [236]. To study the effectiveness of our proposed methods, the performance (in terms of RMSE and response time) is computed under different data sparsity levels (Definition 4). More specifically, it is necessary to explore how the predictive *accuracy* and *response time* of the methods depend on the data sparsity levels of the *sparse datasets*. In order to perform the experiment, sparse datasets with different data sparsity levels (δ) must be created. For this

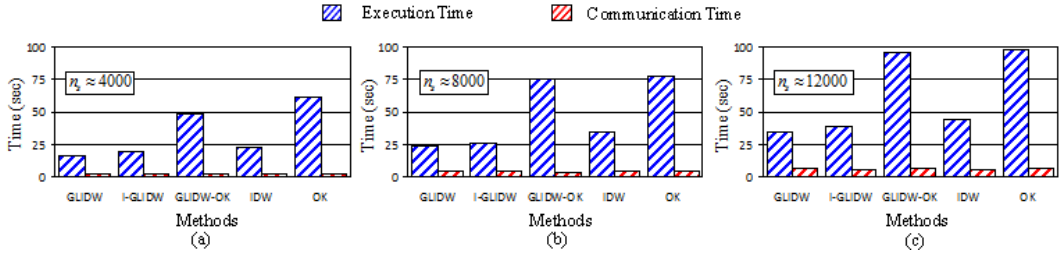


FIGURE 6.16: Execution time (T_E) and communication time (T_c) for different methods by varying the number of data samples n_s. (a) T_E and T_c with $n_s \approx 4000$ (b) T_E and T_c with $n_s \approx 8000$ (c) T_E and T_c with $n_s \approx 12000$

TABLE 6.5: Features and Theoretical Comparison among IDW, OK, GLIDW, I-GLIDW and GLIDW-OK

Basis of comparison	IDW [133]	OK [86] [234]	GLIDW	I-GLIDW	GLIDW-OK
Type of Interpolation	Deterministic	Geostatistical	Deterministic method	Deterministic method	Combined method (Deterministic + Geo-statistical)
Exact/Inexact[a]	Exact method	Can be exact/inexact	Exact method	Exact method	Exact method
Local/Global[b]	Local method	Local method	Local method	Local (can be forced to be global)	Local method
Convex/non-convex[c]	Convex method	Can be non-convex	Convex method	Convex method	Can be non-convex
Limitation	If data samples are sparse/uneven, then it may not works well	The assessment of error largely depends on semivariogram, data sample distribution etc.	It is very sensitive to missing values and unable to predict all grid cells' noise	Number of iterations can be very high	Sometimes the predicted values are unrealistic, when the data set is very sparse
Computation time	$O(\beta n_s^2)$	$O(\beta n_s^3)$	$O(\lvert\mathsf{G}\rvert n_s log(n_s))$	$O(\lvert\mathsf{G}\rvert(n_s log(n_s) + I.GN_{R'}^{(total)}))$	$O(\lvert\mathsf{G}\rvert n_s^3)$
Space requirement	$O(\beta n_s)$	$O(\beta n_s^2)$	$O(\lvert\mathsf{G}\rvert + n_s)$	$O(\lvert\mathsf{G}\rvert + n_s + GN_{R'}^{(total)})$	$O(\lvert\mathsf{G}\rvert n_s^2)$
Suitability	It works well when data points are sufficiently dense	If enough data samples are available to compute semivariograms, then it provides good results.	If sufficient sample points are available in each grid cell, then it performs well	If spatial variation in noise levels is not abrupt, then it works well.	It can handle sparse as well as irregularly spaced data samples very well

n_s : Total number of sample points; β : Total number of unknown locations for estimating noise; $|\mathsf{G}|$: Total number of grid cells; I : Number of iterations in $I-GLIDW$; $|GN_i|$: Number of neighbors; $GN_{R'}^{(total)}$: Total number of neighbours within radius R'.

[a] An Exact method predicts a value that is the same as the observed/known value at a sampled point. An inexact method predicts a value that differs from its observed/known at a sampled point.

[b] A local method uses the samples within a search window to estimates an unknown value. A global method uses every known point in the region of interest to estimate an unknown value.

[c] A convex method generate predicted values that lie somewhere in between the minimum and maximum of the observed/known sample values. A non-convex method can generate predicted values that lie outside the range of minimum and maximum of the observed/known sample values.

TABLE 6.6: Performance of methods under different data sparsity levels

Methods	$\delta = 0\%$		$\delta = 10\%$		$\delta = 20\%$		$\delta = 30\%$		$\delta = 40\%$		$\delta = 50\%$		$\delta = 60\%$		$\delta = 70\%$		$\delta = 80\%$	
	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE
GLIDW	25.40	6.15	23.30	14.68	19.62	15.07	17.66	15.28	15.30	15.44	13.28	15.62	10.34	15.86	8.76	16.12	6.61	16.23
I-GLIDW	27.24	6.17	24.21	9.45	23.80	10.01	21.56	11.13	17.75	11.29	15.14	12.76	12.91	13.32	12.11	14.17	10.44	15.69
GLIDW-OK	81.34	6.15	72.88	10.65	63.04	10.86	51.39	11.34	39.15	11.61	33.71	11.69	29.78	11.76	25.17	11.98	21.82	12.62
IDW	35.32	6.87	32.42	14.62	27.44	15.01	23.62	15.23	20.47	15.53	16.37	15.66	13.59	15.83	12.19	16.22	10.64	16.49
OK	77.75	6.69	73.26	12.08	68.28	13.12	62.75	13.77	51.35	14.03	43.72	14.62	36.88	15.11	29.44	15.97	23.39	16.33

δ: Data sparsity level; T_R: Response time (in seconds), which is the sum of method execution time T_E and communication time T_C; RMSE: Root Mean Squared Error

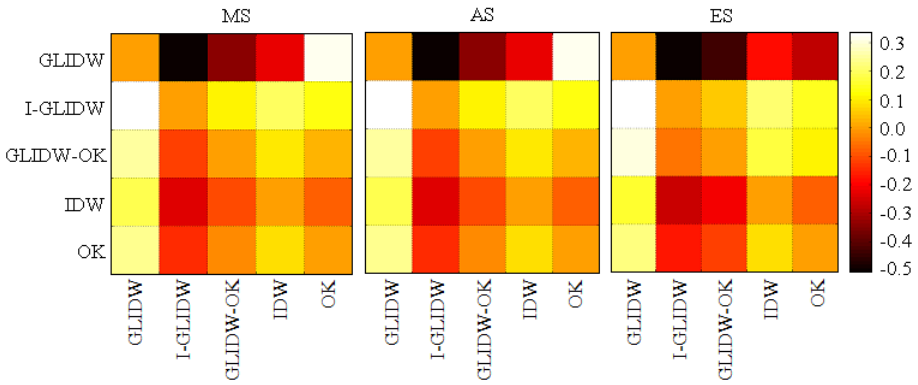


FIGURE 6.17: Relative Improvements between all possible pairs of methods in different sessions of a day

purpose, first a sub-region is selected where each of the grid cells of the sub-region must contain some noise samples. It should be noted that a sub-region of the above-mentioned type is necessary to be selected for the data sparsity experiment instead of considering the whole study area. It is because the study area already contains many grid cells having no noise samples, hence the whole study area cannot be directly used to achieve the data sparsity levels less than what it originally has. Now, the datasets with different δ values (i.e. data sparsity levels) can be created by randomly eliminating the girds with the noise samples from that sub-region. For instance, a dataset with $\delta = x\%$ can be created by randomly eliminating $x\%$ of the girds (along with the noise samples) from that sub-region. For the experiment, the datasets are generated corresponding to 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, and 80% data sparsity levels. Table 6.6 shows predictive accuracy (RMSE) and response time (T_R) of our proposed methods under different data sparsity levels. From that table, we can see, if $\delta = 0\%$ (i.e when all grid cells contain noise samples), GLIDW dominates over other methods in terms of both accuracy and response time. So, if a dataset is very dense, we prefer to use GLIDW to produce a noise map. When δ is in range $(10\%, 40\%)$ I-GLIDW produces a high accuracy noise map with a good response time. So, if data sparsity medium, we prefer to use I-GLIDW. If data sparsity is high $(\delta > 40\%)$, then GLIDW-OK dominates over other methods in terms of predictive accuracy. Though, in this case, GLIDW-OK takes more time to produce the noise map, we prefer to use this method as it provides much better accuracy than other methods. In summary, from the performance evaluation of the methods with respect to the three scenarios (sparse dataset, high accuracy, response time), it is confirmed that the proposed methods dominate the baselines (IDW, OK). Now, it should be important to mention how we measure RMSE for GLIDW where δ is not zero in the above data sparsity experiment. In the case of $\delta > 0$, there will be some grid cells having no noise samples. In GLIDW, if there are no noise samples in a grid cell, then GLIDW unable to predict the noise value for that grid cell and returns NULL. But, in order to calculate overall RMSE, we need to have the predictions for all the grid cells. In GLIDW, to overcome this problem we replace NULL values with the mean of the non-NULL predictions.

6.6 Summary

In this chapter, an IoT-cloud based system is proposed that can provide a high granular and spatially continuous noise pollution map as a service to the users. More specifically, we focus on the IoT based approach that uses participatory sensing to generate dynamic noise maps with the help of our proposed spatial interpolation techniques (GLIDW, I-GLIDW, GLIDW-OK). The presented algorithms can handle a variety of scenarios, including sparse datasets, high accuracy, faster response times, and so on. The IoT-cloud based end-to-end system reduces the overall cost of noise pollution monitoring and can aid citizens, policy makers and city authorities with high granular noise maps. The high granular noise maps not only help citizens to assess their individual exposure to ambient noise in their daily environment but also allow them to keep track of noise pollution levels in their surroundings and eventually help to improve awareness. In this work, proper calibration and context-aware data cleaning techniques are designed and implemented as pre-processing techniques for the proposed participatory sensing based noise map generation methods. Naive Bays classifier is employed for context classification. The performances of the proposed techniques are evaluated using participatory sensing data collected by participants over a period of 2 years (June 2017-June 2019) in the Jadavpur area of city Kolkata. The participants of diverse demographic characteristics (e.g. occupation, age, gender, place of residence, etc.) take part in the data collection process. We also compare the performance of these spatial interpolation methods for noise map generation with two baseline methods namely inverse distance weighting (IDW) and ordinary kriging (OK). The experimental results show that the proposed methods outperform the baseline methods. The results also indicate that GLIDW performs well when the dataset is very dense. On the other hand, I-GLIDW and GLIDW-OK dominate if data sparsity levels are medium and high respectively. It is also observed that the Relative Improvements (RI) of I-GLIDW and GLIDW-OK are always positive as compared to the baseline models. It is also interesting to note that if we have spatially dense datasets, then GLIDW dominates over other methods, and for such a scenario, I-GLIDW or GLIDW-OK is not needed. In this case, GLIDW can be used as an independent and separate method, and it is not needed for GLIDW to be acted as a preprocessing algorithm. However, if the dataset is sparse, only GLIDW is not sufficient. In such cases, the algorithms like I-GLIDW and GLIDW-OK are required and GLIDW is used as a preprocessing algorithm.

This chapter mainly explores the best performing models for a particular study area. However, while a range of powerful models can be offered for distinct problem-specific concerns for diverse urban dynamics, it has been revealed that no single optimal model can outperform in all study areas. It is because there may exists spatio-temporal variations in the target variables' data. In this context, the next chapter intends to conduct a case study to evaluate spatial and temporal heterogeneity in model performance. Specifically, we consider the deep learning-based models for predicting COVID-19 health dynamics.

Publication(s) related to this Chapter:

- *A.I. Middya, and S. Roy, 2021. Spatial interpolation techniques on participatory sensing data. ACM Transactions on Spatial Algorithms and Systems, 7(3), pp.1-32. (DOI: https://doi.org/10.1145/3457609)*

Chapter 7

Mining Spatio-temporal Variation in the Performance of the Models

7.1 Introduction

Spatio-temporal data analysis is becoming increasingly important due to the increasing availability of large scale data in various fields of research. Deep learning models have recently been used in spatio-temporal predictive analysis in a variety of applications, including pollution monitoring (e.g., particulate matter prediction), climate monitoring (e.g., precipitation forecasting), public health monitoring (e.g., COVID-19 health outcome prediction), and so on. However, due to the spato-temporal variations in the data of target variables, the models' performance varies greatly both spatially and temporally. In the aforementioned context, this chapter intends to study spatial and temporal heterogeneity in the effectiveness of deep learning-based approaches for estimating COVID-19 health outcomes in India.

At the beginning of 2020, the world health organization announced the COVID-19 outbreak a public health emergency of global concern. So far, COVID-19 has been associated with over 256 million confirmed cases and about 5.2 million fatalities globally. Many countries seek to protect their citizens' health by imposing mitigation strategies including social distancing, quarantines, travel restrictions, hard and soft locks, as well as postponing and cancellations of activities. In this respect, numerous deep learning-based models for estimating COVID-19 health consequences have been developed in the existing literature to help with healthcare resource management and mitigation planning. Specifically, precise prediction could be useful in determining goals, guiding principles, and strategies for mitigating potential COVID-19 consequences.

Previous deep learning-based investigations [98, 247–250] from around the world have revealed a variety of high-performing techniques for anticipating COVID-19 health effects. It is worth noting that the deep learning techniques can learn and extract features from raw and imperfect data on their own. Specifically, these techniques could be helpful in a time series forecasting scenario for automatically learning the temporal dependency from the data. Moreover, advanced deep learning techniques, e.g., LSTM (long short-term memory) [180, 181, 251] and its variants, are highly capable in detecting patterns from input data that span lengthy periods of time. These techniques can help to address the problem by removing the requirement for extensive feature engineering activities, data scaling techniques,

and differencing to make the data stationary. The classical timeseries forecasting methods cannot perform well if there exist complex associations between the variables, noisy data, missing data, and irregular temporal patterns in data. In other words, to perform successfully, classical timeseries forecasting methods often require clean and comprehensive data sets. On the other hand, the deep learning based techniques are resistant to noise in the input data, and can even learn and forecast in the existence of missing data.

7.1.1 Objective

Despite the fact that a number of powerful deep learning-based techniques are capable of addressing a range of problem-specific challenges in various domains, it has been noticed that there is no one optimal deep learning-based forecasting model that can outperform everywhere. This is because there is a wide variation exist in the COVID-19 health outcome time-series. It is worth noting that the variation in COVID-19 health outcome time-series depends on various risk factors that vary greatly across the geographical space. Moreover, several literature [252,253] highlights that there exists a geographically varying nature of the association between COVID-19 health outcomes and the complex risk factors. Hence, it is expected that there should have some variation in the models' performance. In the above-mentioned context, the objective of this study is to explore the spatio-temporal variation in the effectiveness of the deep learning based predictive methods for predicting covid-related health outcomes.

There are several existing deep learning based forecasting methods for COVID-19 health outcome forecasting. For instance, in [98], Mohimont et al. employed temporal CNN to forecast various health outcomes, e.g., number of confirmed cases, deaths, etc. Chimmula et al. [247] predicted the spread of COVID-19 in Canada by utilizing long short-term memory (LSTM) networks. Additionally, several earlier studies [248–250] investigate different LSTM variations for covid-19 prognosis, including Bi-LSTM, Stacked LSTM, etc. To our knowledge, none of the work that currently exists has examined spatio-temporal variation in the performance of deep learning-based techniques. This study not only attempts to explore such spatio-temporal variation in the models' performance but also aims to identify spatial and temporal distribution of the methods for predicting covid-related health outcomes in India.

7.1.2 Contribution

The major contributions of this work can be summarized in the following points.

- Explore spatio-temporal variability pattern in the performance of deep learning based forecasting models (namely CNN, RNN, Vanilla LSTM, LSTM Autoencoder, and Bidirectional LSTM) for predicting COVID-19 health outcomes in India.
- Thoroughly evaluate the performance of the methods based on various performance indicators for COVID-19 mortality time-series data of 36 states and union territories of India for various forecast horizons.

- Finding spatial distribution of best-performing models for predicting Covid-19 mortality data for different temporal resolutions (e.g., one week, two weeks, etc).

7.2 Existing Works

This section discusses existing works that are primarily concerned with spatiotemporal data modeling of COVID-19 health outcomes. Specifically, in the following paragraphs, we present related works, their limitations/gaps, and how our work attempts to bridge the gaps.

Several previous works capture patterns of COVID-19 progression using mathematical, machine learning, and deep learning techniques [254]. Kavouras et al. [255] studied the efficacy of employing deep learning techniques to properly model COVID-19 transmission. However, they focuses on low-granular (country-level) data modelling and analysis over EU (European Union) regions. Furthermore, they created models that can only predict COVID-19 health outcomes on a daily basis (i.e., short forecast horizon). Nikparvar et al. [256] conducted spatio-temporal forecasting of COVID-19 health outcomes in US counties. However, they solely utilized the vanilla LSTM and did not compare the performance of their suggested model to more advanced deep learning models, such as other LSTM versions. The authors of [257] performed real-time spatiotemporal prediction and monitoring of COVID-19 in Brazil by utilizing both geographical and time-dependent characteristics. Their approach ultimately results in low predictive accuracy since it solely concentrates on shallow machine learning-based techniques (such as random forests, support vector machines, etc.). Bhimala et al. [258] used the meteorological parameter integrated deep learning technique to estimate COVID-19 incidents for various states of India. However, they only consider a vanilla LSTM model for relatively short forecast horizon. Conventional epidemiological methods (e.g., the Susceptible-exposed-infectious-removed (SEIR) method and variations) [259–261] have also been used to quantify viral dissemination and predict the influence of policy actions on infection rates. However, these methods suffers from various shortcomings. For instance, these methods are reliant on a large number of hypothesised input factors. Since SEIR methods are so sensitive to variations in these input factors, their predicted accuracy might suffer significantly.

In contrast to earlier research, this study focused on data modelling to investigate if advanced deep learning models can better capture the spatiotemporal variability of COVID-19 health outcomes. Specifically, a variety of advanced deep learning models namely CNN, RNN, Vanilla LSTM, LSTM Autoencoder, and Bidirectional LSTM are built to accurately perform state-level high granular COVID-19 variability analysis in India. This research work also focuses on developing models with a relatively longer forecast horizon (maximum forecast horizon of 21 days, i.e., 3 weeks). Moreover, compared to the existing works, the advanced deep learning models are rigorously evaluated on multiple forecast horizons to verify the models' scalability.

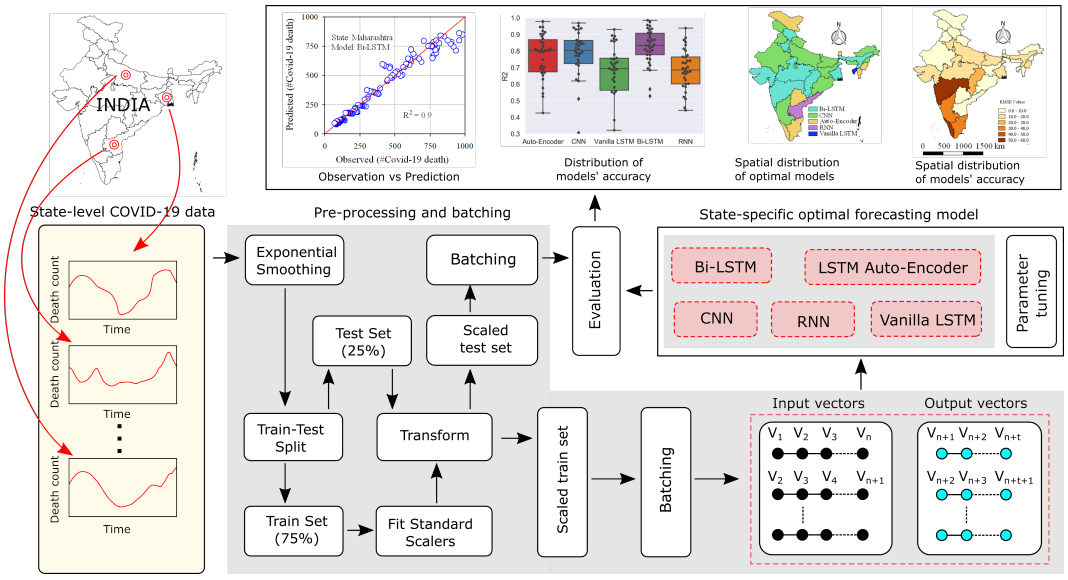


FIGURE 7.1: Overall framework for exploring spatio-temporal variability in the performance of deep learning based approaches for estimating state-level COVID-19 mortality in India.

7.3 Problem statement

Suppose $S = \{S_1, S_2, S_3, \ldots, S_n\}$ is a set of states (=28) and union territories (=8) of India. Given a total of $n = 28 + 8 = 36$ different COVID-19 death time-series datasets corresponding to the states and union territories. Let, $\mathbf{D}_i = \{V_{i,1}, V_{i,2}, V_{i,3}, \ldots, V_{i,m}\}$ denotes the time series dataset of daily COVID-19 death count for $S_i \in S$. Here, $V_{i,j}$ is the number of COVID-19 deaths on j-th day in $S_{i \in [1,2,\ldots,36]}$. Suppose, $M = \{M_1, M_2, \ldots, M_k\}$ is a set of popular deep learning based time-series forecasting models that could be used for predicting future values of COVID-19 health outcomes. The goal of this research work is to determine the optimal set of forecasting models $M^{(o)} \subseteq M$ and a mapping between S and $M^{(o)}$, i.e., $\sigma : \{p : S_p \in S\} \rightarrow \{q : M_q \in M^{(o)}\}$. More specifically, the objective is to find a state-wise spatial mapping of optimal deep learning based methods for estimating the number of covid-related deaths in India. Moreover, this study attempts to explore spatial variability in the performance of the optimal forecasting models.

7.4 Proposed approach

Fig. 7.1 provides a detailed framework for exploring spatio-temporal variability in the performance of deep learning based approaches for estimating state-level COVID-19 mortality in India. This section demonstrates how state-level COVID-19 mortality time-series data is acquired, preprocessed, batched, and subsequently utilized for model training and performance evaluation.

Dataset description

On the basis of daily time-series data from India, location-specific optimal forecasting models are found. The COVID-19 death time series data are collected for a total of 36 states and

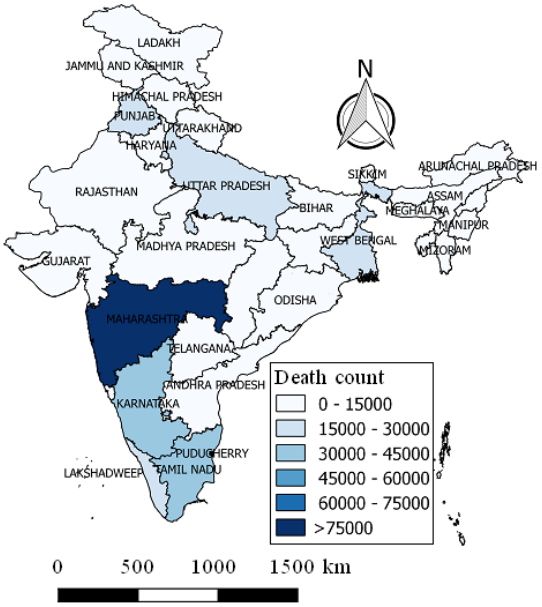


FIGURE 7.2: Spatial distribution of state-level COVID-19 death counts.

union territories of India. More specifically, 36 time-series datasets (one for each of the states and the union territories) are constructed. The raw COVID-19 data are collected up to November 01, 2021 from two sources namely (i) MOHFW website [1] (ii) COVID19INDIA website[2]. COVID19INDIA is a crowd-sourcing project that documents COVID-19 data from India's states and union territories. Each of the 36 datasets contains about 576 days' data samples. Geographical distributions of state-level death counts are presented in Fig. 8.1. More than 46% of the states report more than 10000 COVID-19 related death cases. The state of Maharashtra is the most severely affected by COVID-19, with more than 140000 deaths.

Data pre-processing and batching

This section provides the details of the pre-processing of the raw time-series data and batching of the pre-processed data. The COVID-19 mortality time-series for each state and UTs is smoothed with an exponential function to lessen the jaggedness of the curve. Eq. 7.1 provides the mathematical form of exponential the smoothing function.

$$Z_k = \delta X_k + (1-\delta) Z_{k-1}, \quad k > 0 \tag{7.1}$$

where δ represents the smoothing factor and $\delta \in (0,1)$. The raw COVID-19 mortality time series data is denoted by $\{X_k\}$ starts at $k = 0$ and the smoothed output is represented as $\{Z_k\}$. Note that Eq. 7.1 considers that $Z_0 = X_0$.

The data are divided into train and test sets once it has been smoothed. The time-series data cannot be shuffled since the order is determined by 'timestamp'. As a result, the test set

[1] https://www.mohfw. gov.in/
[2] https://www.covid19india. org/

is made up of the last 150 days of data, which constitutes around 25% of the whole dataset. After that, we fit a standard scaler to the training data only, then use that scaler to standardise both the training and test data. Standardization can be expressed mathematically in Eq. 7.2.

$$v = \frac{x - \bar{x}}{\sigma} \tag{7.2}$$

where, x denotes data point, $\bar{x}$ and σ represent mean and standard deviation respectively.

Next, the time-series sequence data are converted into the supervised form which we call the batching process. Let us consider, $Seq = [v_1, v_2, v_3, \ldots, v_r]$ is a sequence of data values found after standardizing the mortality time-series. The sequence of data can be split into several input-output vectors, where n consecutive values are considered as an input vector and next t consecutive values are considered as an output vector (i.e., target vector). If the input and corresponding output vectors are denoted by $\mathbf{x}$ and $\mathbf{t}$ respectively, the scaled data after batching can be represented as $L = \{(\mathbf{x}_1, \mathbf{t}_1), (\mathbf{x}_2, \mathbf{t}_2), \ldots, (\mathbf{x}_z, \mathbf{t}_z)\}$, where, $\mathbf{x}_i = [v_i, v_{i+1}, v_{i+2}, \ldots, v_n]$, and $\mathbf{t}_i = [v_{n+1}, v_{n+2}, v_{n+3}, \ldots, v_{n+t}]$. Note that, $z = r - (n + t - 1)$. The details of the batching process are provided in Algorithm 1.

Algorithm 9 Batching (Seq, n, t)

Input: A sequence Seq of data values, input vector length n, output vector length t
Output: Set of input-output pairs L

$L = \emptyset$
$S_len \leftarrow$ Number of elements in Seq
for $i = 1$ to $S_len - (n + t - 1)$ **do**
    $x_end = i + n$ // End index of current input
    $t_end = x_end + t + 1$ // End index of current output
    $\mathbf{x}_i = Seq[i : x_end]$ // i-th input vector
    $\mathbf{t}_i = Seq[x_end + 1 : t_end]$ // i-th output vector
    $L = L \cup (\mathbf{x}_i, \mathbf{t}_i)$
end for
return L

Deep learning models

This section discusses various deep learning models that are employed in predicting state-level COVID-19 health outcomes in India. Spatial variation in the models' performance could then be used in finding the geographical distribution of best-performing models.

Convolutional neural network (CNN)

The CNN [262, 263] is usually used to extract features from two-dimensional data and is commonly utilized for image processing. However, because of its ability to learn both local and global characteristics from sequence data, CNNs are also used in several time-series applications. In this context, a one-dimensional CNN (1D-CNN) is employed for forecasting COVID-19 health outcomes from COVID-19 mortality time-series data. CNN's major feature learning layers are the convolutional and pooling layers. The following is the procedure

(i.e., Eq. 7.3) for performing a 1D convolutional operation

$$(x*s)[i] = \sum_{j=0}^{m-1} x[i-j] \cdot s[m-1-j] \tag{7.3}$$

where, x and s are the input vector and the kernel respectively; m is the kernel size. Across the feature maps produced by the convolution layer, the pooling is utilized to decrease the size of the feature map.

Recurrent neural network (RNN)

For sequence prediction tasks, RNNs [264] are the most often utilized neural network architecture. RNN consists several RNN units. Elman developed the ERNN (Elman RNN) cell, which is a type of base recurrent unit [265]. An ERNN cell could be mathematically expressed using Eq. 7.4 and Eq. 7.5.

$$J_t = \sigma\left(K_i \cdot x_t + Q_i \cdot J_{t-1} + b_i\right) \tag{7.4}$$

$$z_t = \tanh\left(Q_o \cdot J_t + b_o\right) \tag{7.5}$$

where the input and output of the cell at t-th time step are x_t and z_t respectively, J_t indicates hidden state. For the hidden state, Q_i and K_i represent the weight matrices, and b_i represents the bias vector. Similarly, Q_o and b_o are the cell output's weight matrix and bias vector, respectively. Here, $\sigma(\cdot)$ represents the sigmoid function. Note that the present hidden state is determined by the earlier time step's hidden state as well as the present input. For the lengthy sequences, the RNNs suffer from the exploding gradient and vanishing gradient issues. This means that RNN will not be capable to carry long-range dependencies into the future.

Vanilla LSTM

Many variants of the basic RNN have been produced over time to overcome its flaws [264]. LSTM model, which is able to learn long-range dependencies, is possibly the most popular branch of RNN. An LSTM cell has two states: the internal cell state and the hidden state, which are different from the standard RNN cell. Eq. 7.6 - Eq. 7.12 describes an LSTM cell mathematically

$$i_t = \sigma\left(K_i \cdot x_t + Q_i \cdot J_{t-1} + b_i\right) \tag{7.6}$$

$$o_t = \sigma\left(K_o \cdot x_t + Q_o \cdot J_{t-1} + b_o\right) \tag{7.7}$$

$$f_t = \sigma\left(K_f \cdot x_t + Q_f \cdot J_{t-1} + b_f\right) \tag{7.8}$$

$$\tilde{C}_t = \tanh\left(K_c \cdot x_t + Q_c \cdot J_{t-1} + b_c\right) \tag{7.9}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{7.10}$$

$$J_t = o_t \odot \tanh(C_t) \tag{7.11}$$

$$z_t = J_t \tag{7.12}$$

where cell state and hidden states at t-th time step are represented by C_t and J_t, $\tilde{C}_t$ denotes candidate cell state. The input and output are x_t and z_t respectively. Q_c, Q_f, Q_o, and Q_i represent the weight matrices of the cell state, forget gate, output gate, and input gate respectively. Similarly, the weight matrices related to the present input are represented by K_c, K_f, K_o, and K_i. On the other hand, b_c, b_f, b_o, and b_i are bias vectors. The forget, output, and the input gate vectors are denoted by i_t, o_t, and f_t respectively. The function $\sigma(\cdot)$ denotes the sigmoid activation. For time series forecasting, LSTM is one of the most commonly utilized models.

LSTM Autoencoder

An autoencoder (AE) [266] is a from of neural network architecture utilized to reduce dimensions and feature extraction. Specifically, an AE compresses the data to a smaller-dimensional code, which they then use to rebuild the output. In general, it is made up of two key parts such as an encoder and a decoder. Compressed representations are learnt by the model during the encoding stage. On the other hand, the reconstructions of the outputs from the compressed representation are performed in the decoding stage. The encoding and decoding could be mathematically expressed using Eq. 7.13 and Eq. 7.14.

$$q(x) = g(W_1 x + b_1) \tag{7.13}$$

$$y = f(W_2 q(x) + b_2) \tag{7.14}$$

where, $g(\cdot)$ and $f(\cdot)$ are encoder and decoder functions respectively; x represents the input to the encoder; y represents the output from the decoder; $q(x)$ is the encoder's output which is also used as the input to the decoder; b_1 and b_2 represents the bias vectors. The AE is trained by minimizing the reconstruction error.

The LSTM autoencoder [248] is an AE in which the LSTM network serves as both the encoder and the decoder components. The capability of LSTM to identify patterns in time series data over extended periods makes them ideal for creating a forecasting model.

Bidirectional LSTM

Bidirectional LSTMs (Bi-LSTMs) [248] are a type of LSTM that considers both past and future states in order to increase predictive performance. The Bi-LSTM is made up of two separate LSTM networks, one processing the time-series data from right and the other processing it from left. It can learn sequential input through both backward and forward directions using the LSTM architecture. The forecast result for the upcoming time step is generated by

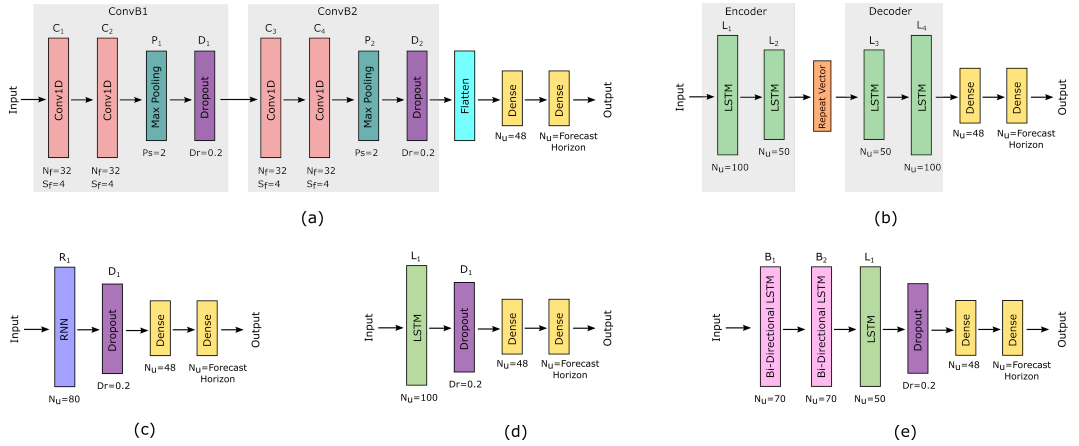


FIGURE 7.3: Model architecture for state-level COVID-19 death count forecasting models (a) CNN (b) LSTM Autoencoder (c) RNN (d) Vanilla LSTM (e) Bi-Directional LSTM.

combining the outcomes of both the backward and forward LSTM networks. In this work, Bi-LSTM is found to be an effective technique for the future prediction of COVID-19 health outcomes.

Implementation of deep learning models

This section discusses the details of problem-specific model architectures, training, hyper-parameter tuning, etc.

Fig. 7.3 presents the architecture of the models used in this study to predict COVID-19 health outcomes at the state level in India. Fig. 7.3(a) presents the CNN-based model architecture for prediction, which consists of two convolutional blocks (ConvB1 and ConvB2). Each of the convolutional blocks has two Conv1D layers followed by a max pooling and a dropout layer. For each Conv1D layer, number of filters (N_f) = 32 and kernel size (S_f) = 4. A pool size (P_s) of 2 and dropout rate (D_r) of 0.2 are used for max pooling and the dropout layers respectively. The output of ConvB2 is flattened by the flatten layer which is followed by two dense layers. In the first dense layer, the number of units (N_u) is 48. On the other hand, in the second dense layer, N_u is the value of the forecast horizon (i.e., the number of output steps). Fig. 7.3(b) presents the architecture of the LSTM auto-encoder. The encoder part contains two LSTM layers (L_1 and L_2) with 100 and 50 units respectively. Similarly, the decoder part has two LSTM layers (L_1 and L_2) with 50 and 100 units. There is a repeat vector layer between the encoder and decoder parts. Finally, similar to CNN, there are two dense layers with the same N_u values. The architecture of the RNN based predictive model is presented in Fig. 7.3(c). It made up of an RNN layer followed by a dropout layer and two Dense layers. Here, N_u in the RNN layer and the dropout rate in the dropout layer are N_u = 80 and D_r = 0.2 respectively. Fig. 7.3(d) presents the architecture of the vanilla LSTM. It has a single LSTM layer with 100 units followed by a dropout layer (D_r = 0.2) and two dense layers. Finally, the bi-directional LSTM model's architecture is shown in Fig. 7.3(e). There are two bi-directional LSTM layers each with N_u of 70. These layers are followed by an LSTM layer (with N_u = 50), a dropout layer (D_r = 0.2), and two dense layers.

As previously stated, approximately 25% of the most recent COVID-19 mortality data is included in the test set, while the remaining 75% is used for training purposes. When creating learning models, hyper-parameter optimization is a critical step. The manual process of trying random combinations takes a long time because identifying the set of optimal parameters is a time-consuming procedure. To combat this, a parameter tuning technique known as grid-search is applied to various sets of parameter values to identify optimal parameter settings. In this work, model-specific parameter settings are identified by considering a search space specific to the model. The parameters considered during model tuning are: learning rate, batch size, kernel size, number of epochs, dropout rate, and number of units. The parameter settings are found based on a validation set taken from the training data.

MSE (mean squared error), which is expressed in Eq. 7.15, is the loss function used for the model training.

$$MSE = \frac{\sum_{k=1}^{N}(A_i - P_i)^2}{N} \tag{7.15}$$

where, A_i and P_i denote the actual and estimated death counts and N denotes the total number of instances. The MSE loss is suggested over the MAE loss because the MAE derivative remains constant whereas the MSE derivative reduces as the estimation approaches to the target. MSE's learning rate would slow down, whereas MAE's would stay the same.

When training the predictive models, the Adam optimizer is considered. The Adam optimizer could be expressed using the following set of equations (Eq. 7.16 - Eq. 7.19).

$$g_{t+1} = g_t - \hat{q}_t \left(\frac{\gamma}{\sqrt{\hat{u}_t + \varepsilon}} \right) \tag{7.16}$$

$$\hat{q}_t = \frac{q_t}{1 - \xi_1^t}, \quad \hat{u}_t = \frac{u_t}{1 - \xi_2^t} \tag{7.17}$$

$$q_t = \xi_1 q_{t-1} + (1 - \xi_1) \left[\frac{\delta F}{\delta g_t} \right] \tag{7.18}$$

$$u_t = \xi_2 u_{t-1} + (1 - \xi_2) \left[\frac{\delta F}{\delta g_t} \right]^2 \tag{7.19}$$

where, loss function is denoted by L; learning rate is denoted by γ; the momentum is represented by q; sum of the squares of the previous gradient is denoted by u; ξ_1 and ξ_2 represent the decay coefficients; the bias-corrected form of u_t and q_t are $\hat{u}_t$ and $\hat{q}_t$ respectively; g_t represents the weight at t-th step.

7.5 Evaluation

7.5.1 Experimental setup

The numerical experiments are conducted under a Windows environment on a PC with 3.30 GHz Intel(R) Xeon(R) CPU E3-1226 v3 processor and 128 GB of RAM. Python[3] is used to implement all of the deep learning-based COVID-19 health outcome prediction models and various analyses. The predictive models are developed using the following major Python libraries: (i) numpy (version 1.19.5) (ii) pandas (version 1.1.5) (iii) keras (version 2.7.0) (iv) sklearn (1.0.1). Moreover, for the visualization of spatial data, QGIS (Quantum GIS) is utilized.

7.5.2 Evaluation metrics

Various evaluation metrics (namely mean absolute error, root mean squared error, and coefficient of determination) are used to assess the effectiveness of the methods for predicting COVID-19 mortality counts. The mathematical form of root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) are expressed in Eq. 7.20, Eq. 7.21, and Eq. 7.22 respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T}(O_i - E_i)^2}{T}} \tag{7.20}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|O_i - E_i| \tag{7.21}$$

$$R^2 = 1 - \frac{RSS}{TSS} \tag{7.22}$$

where, O_i and E_i are i-th observed and the predicted value, T is the total number of predictions, sum of squares of residuals is represented by RSS, and total sum of squares is represented by TSS.

7.5.3 Spatial variability in the models' performance

For different states and union territories, deep learning based forecasting techniques are built using state-level COVID-19 mortality data. The geographical distribution of MAE and RMSE values for different states and union territories are provided in Fig. 7.4 and Fig. 7.5 respectively. The accuracies are computed for a 3-week forecast horizon. It is observed that for the north-western (primarily the states of Rajasthan and Punjab) and south-eastern (primarily the states of Andhra Pradesh, Telangana, and Odisha) parts of India, the CNN-based

[3]https://docs.python.org/3.7/

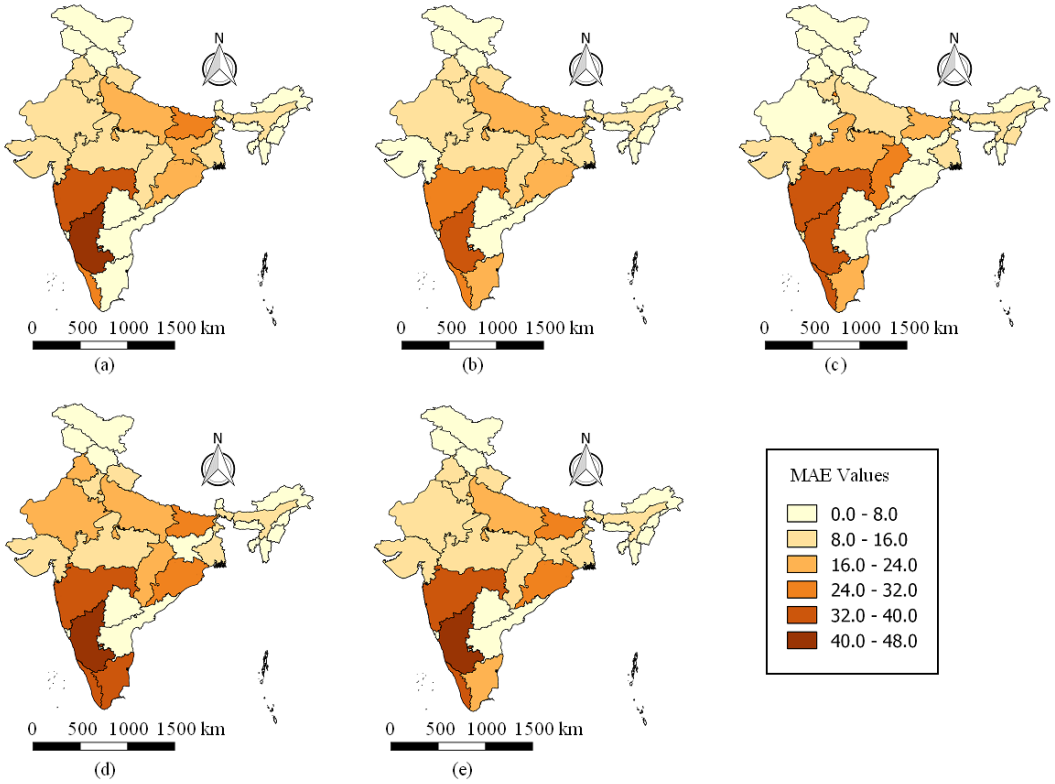


FIGURE 7.4: Spatial distribution of MAE values for different models. (a) LSTM Auto-Encoder (b) Bi-LSTM (c) CNN (d) vanilla LSTM (e) RNN

predictive models perform better in terms of MAE and RMSE values. Almost all of the models achieve better accuracy in the northern and eastern parts. In the southern parts (primarily the states of Kerala and Tamil Nadu) the LSTM auto-encoder performs relatively better compared to other methods with respect to both the MAE and RMSE values. In comparison to the other models, the Bi-LSTM model performs better in the western regions (primarily the states of Gujarat, Maharashtra, and Karnataka).

The distribution of the models' COVID-19 health outcome predictive accuracies are also observed to vary significantly over the forecast horizon. Fig. 7.6, Fig. 7.7, and Fig. 7.8 present the distribution of MAE, RMSE, and R^2 values over three forecast horizons (1 week, 2 weeks, and 3 weeks). The accuracy distribution is visualised using box plots that overlap with the corresponding swarm plots. The box plots show various distribution summaries namely minimum value, first quartile, median value, third quartile, and the maximum value. Note that a swarm plot corresponding to a box plot indicates where the data point falls in the distribution summaries. For the box plots of MAE (Fig. 7.6) and RMSE (Fig. 7.7) values, it is clear that all the methods perform well for a shorter horizon. For a shorter forecast horizon, the interquartile range (IQR) for different models is smaller. It implies that the predictive accuracy across most of the states does not differ significantly for shorter forecast horizons. The increase in IQR due to the increase of forecast horizon is less in the case of Bi-LSTM model. Moreover, for both the MAE and RMSE, the visual analysis reveals that the models like Bi-LSTM and CNN achieve better median accuracies across the forecast horizons. Fig. 7.8 shows relatively high R^2 values are by Bi-LSTM and CNN models with less IQR.

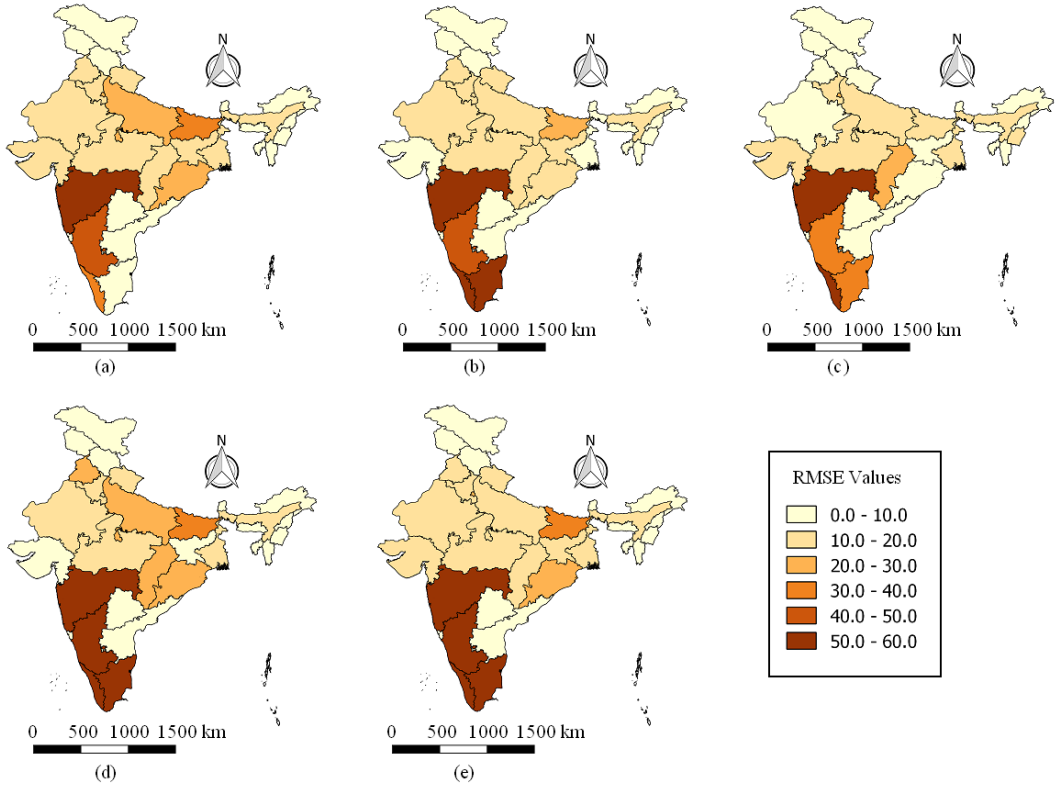


FIGURE 7.5: Spatial distribution of RMSE values for different models. (a) LSTM Auto-Encoder (b) Bi-LSTM (c) CNN (d) vanilla LSTM (e) RNN

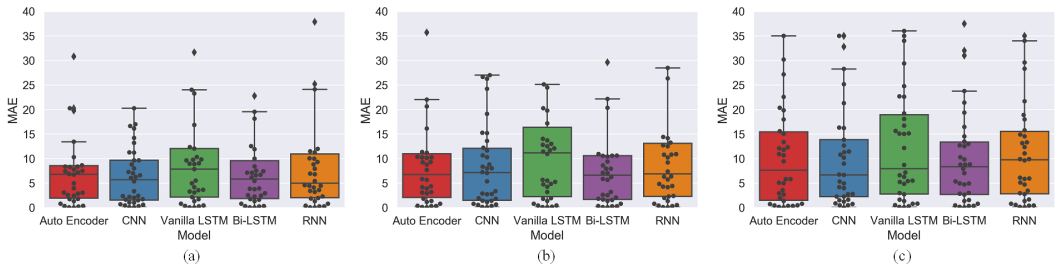


FIGURE 7.6: Box and swarm plots of MAE values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks

The observation versus prediction plots of COVID-19 death count for some of the highly covid-affected states is provided in Fig. 7.9. For a particular state, the plots are provided only for the best-performing model. The results show that the estimated COVID-19 death counts are closely spread along the $x = y$ line relative to the observed values in the majority of cases. The models are unable to predict higher values for some states and union territories (e.g., Maharashtra, Delhi, etc.).

7.5.4 Spatial distribution of optimal models

This section presents the spatially varying distribution of best-performing models for predicting COVID-19 death counts in India. There is no single best method that can outperform across different states in India due to the wide variation in COVID-19 mortality time-series. Fig. 7.10 shows the distribution of the state-specific optimal models in India for different

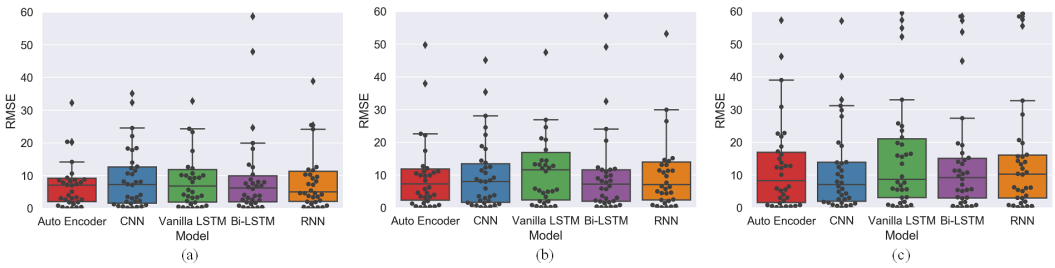


FIGURE 7.7: Box and swarm plots of RMSE values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks

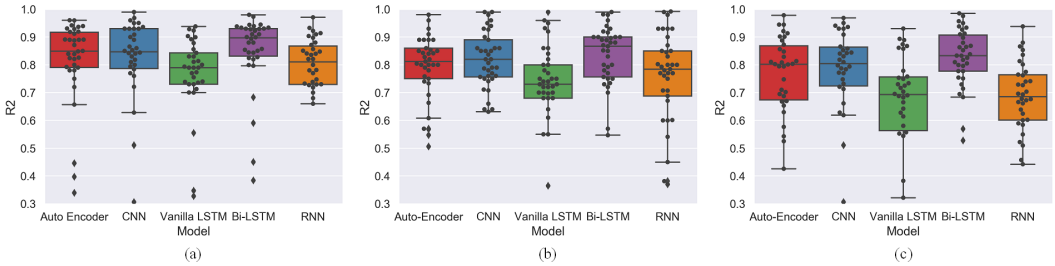


FIGURE 7.8: Box and swarm plots of R^2 values for various predictive models over different forecast horizons (a) one week (b) two weeks (c) three weeks

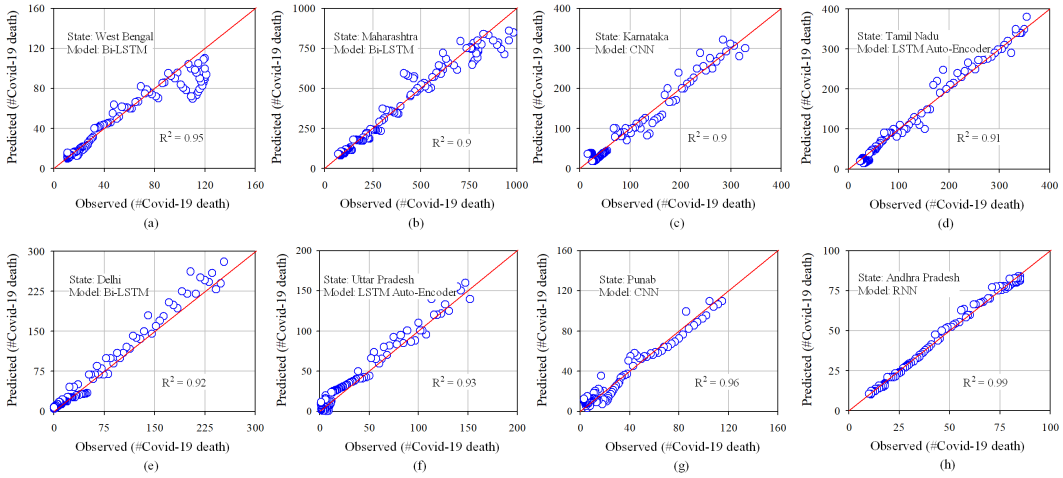


FIGURE 7.9: Observation versus predicted plots of COVID-19 death count for some of highly covid-affected states (a) West Bengal (b) Maharashtra (c) Karnataka (d) Tamil Nadu (e) Delhi (f) Uttar Pradesh (g) Punjab (h) Andhra Pradesh

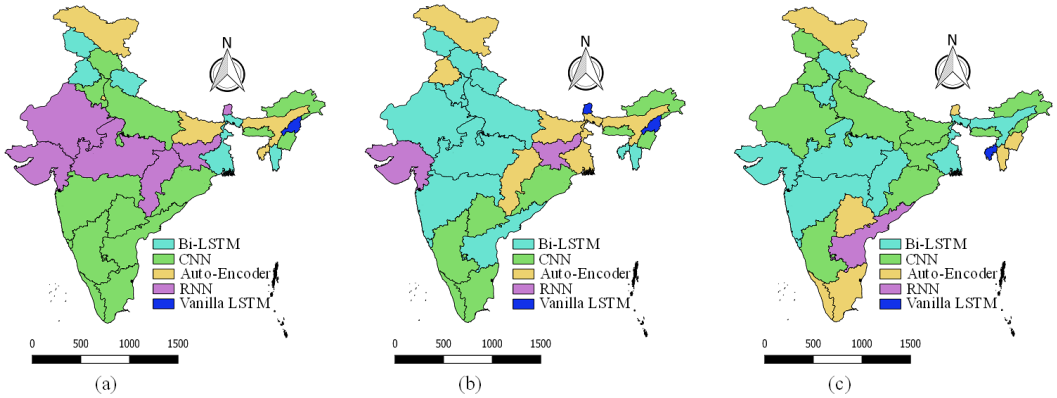


FIGURE 7.10: Geographical distribution of the best performing models for different forecast horizons (a) one week (b) two weeks (c) three weeks

forecast horizons (1 week, 2 weeks, and 3 weeks). It is interesting to observe how the distribution of best-performing models changes over different forecast horizons. Fig. 7.10(a) presents the models' distribution when the forecast horizon is 1 week. Note that in the case of a shorter horizon, both the basic models such as CNN and RNN dominates for most of the states. For instance, CNN mainly dominate in the southern, northern and eastern parts of India. On the other hand, RNN mainly dominates in the central and western parts of India. However, in the case of a longer forecast horizon, some advanced architecture like Bi-LSTM, LSTM auto-encoder outperform instead of RNN. Bi-LSTM and CNN achieve significantly better results across the states in all three forecast horizons. Bi-LSTM provides optimal performance in approximately 14%, 39%, and 34% of the states for a forecast horizon of 1 week, 2 weeks, and 3 weeks respectively. On the other hand, CNN achieves optimal performance in approximately 42%, 28%, and 39% of the states for a forecast horizon of 1 week, 2 weeks, and 3 weeks respectively.

7.5.5 Discussion

The existing studies [267] reveal that the COVID-19 health outcomes depend on complex factors including socio-economic, environmental pollution, meteorological parameters, local strategies for mitigation, and public health measures. The impact of these factors varies greatly across the geographical space in the countries like India. Consequently, it is not unexpected that the nature of COVID-19 mortality time-series varies across different states. Due to the geographically varying nature of the association between COVID-19 health outcomes and the complex risk factors, the predictive models' performance differs significantly across the geographic space.

The results of various deep learning models highlight that they could be adopted by most of the states to forecast COVID-19 health outcomes in advance. Specifically, deep learning methods like Bi-LSTM and CNN produce overall better predictive accuracies compared to the others. However, the findings show that selecting an appropriate forecasting model is influenced not only by the geographic location but also by the forecast horizon for which the prediction is made. For example, in terms of the shorter horizon (e.g., 1 week), CNN

dominates in India's southern, northern, and eastern regions, whereas RNN dominates in the country's central and western regions. In 65% of the states and union territories, the CNN and RNN models perform better. However, the basic RNN does not perform well over longer time horizons, such as 2 or 3 weeks, because it is unable to carry long-term dependencies into the future. For a longer forecast horizon (see Fig. 7.10 (b, c)), Bi-LSTM provides better accuracies compared to RNN. It could also be verified by the box and swarm plots of Fig. 7.6, Fig. 7.7, and Fig. 7.8. The IQR along with the median MAE and RMSE values in Fig. 7.6(b,c) and Fig. 7.7(b,c) are smaller in the case of Bi-LSTM compared to RNN. Furthermore, when compared to the basic RNN, Bi-LSTM has a significantly higher median R^2 value.

7.6 Summary

This chapter focuses on finding the spatial distribution of best performing deep learning based techniques to predict Covid-19 health outcomes over India. Specifically, the use of widely applied deep learning techniques for such forecasting at the state level, as well as the geographic variability of their predictive performance, are investigated. To find the state-specific optimal deep learning model and the spatial variability of their performances, a total of 36 time-series datasets (one for each of India's states and union territories) are used. Various deep learning based time-series forecasting models including CNN, RNN, Vanilla LSTM, LSTM Autoencoder, and Bidirectional LSTM are investigated in this work. The best-performing models' geographical distribution varies significantly across forecast horizons. Models like CNN and RNN dominate for shorter horizons, but models like Bi-LSTM and CNN dominate for longer horizons in most states.

From the findings of this chapter it can be concluded that there is no single best method that can outperform across different states in India due to the wide variation in COVID-19 mortality time-series data. In this context, the variability in the distribution of COVID-19-related health outcomes might be related to many underlying variables, including demographic, socioeconomic, or environmental pollution related factors. Hence in the next chapter, a statistical method called geographically weighted regression (GWR) is employed to explore the geographical relationships between COVID-19 health dynamics and different driving factors.

Publication(s) related to this Chapter:

- *A.I. Middya, and S. Roy, 2022. Spatio-temporal variation of Covid-19 health outcomes in India using deep learning based models. Technological Forecasting and Social Change, 183, p.121911.*
  (DOI: https://doi.org/10.1016/j.techfore.2022.121911)

Chapter 8

Geographically varying Spatio-temoral relationships of COVID-19 health outcomes

8.1 Introduction

This chapter focuses on exploring the heterogeneity in the relationships of health dynamics of citizens with various driving factors. Specifically, in this chapter, the health dynamics related to the COVID-19 is explored which is a global issue. The novel coronavirus disease (COVID-19) has spread rapidly to all parts of the world, causing almost 2.5 million deaths as of mid-February 2021 [268]. Because of its unpredictable nature and lack of appropriate medications, COVID-19 is now a global health concern. There is unprecedented urgency to investigate the major factors that are related to COVID-19 death. In this context, recent studies are focusing on exploring person-specific risk factors for COVID-19-related health outcomes [269–271]. Also, there are research works that examine the association of COVID-19-related health outcomes with different socio-economic, environmental, and region-specific factors [272–274]. These factors play a very important role in determining the patterns of COVID-19 mortality.

Both global and local models can be utilized to explore the above-mentioned associations. A global model comes up with a geographically constant relationship across the entire geographic space. On the other hand, a local model can capture the local relationships that can vary across the geographic space. Most of the studies that focus on exploring the relationship of COVID-19 cases with different possible risk factors are based on global models (e.g. Ordinary Least Square) [92, 275]. But, the global models assume that the associations between the independent variables and the dependent variable are stationary (i.e. homogeneous) throughout the study area. Besides, these models also assume that there is no spatial autocorrelation in the dataset. Eventually, they yield estimates of the parameters that reflect average behaviour [276]. But, in reality, the relationships between the dependent and the independent variables may not be homogeneous and can be geographically varying [277]. Therefore, such models usually suffer from low accuracy especially in those locations where weak association exists between dependent and independent variables. Now, various local techniques can be utilized in order to overcome the above-mentioned shortcomings of

the global models. Some widely encountered local spatial statistics include geographically weighted regression (GWR) [90, 91], local Moran's I [278], spatial regressions, etc.

8.1.1 Objective

In India, no comprehensive study is performed at the local level to investigate geographical relationships between COVID-19 health dynamics and associated potential factors. To bridge the gap, a local method (GWR) is employed to explore the geographical distribution and associated potential socio-economic, demographic, and environmental factors for COVID-19 deaths. Note that, the GWR model helps us to identify whether there is geographical heterogeneity present in the relationships. Moreover, a comparison between local (OLS) and global (GWR) models are also performed. This chapter offers further knowledge and insight into geographical targeting of intervention and control strategies against the COVID-19 epidemic. In summary, the key objectives of this chapter is to explore geographically varying relationships / associations between any phenomena of interest and various risk factors.

8.1.2 Contribution

The major contributions of this chapter are summarized as follows:

- The geographically varying relationships of COVID-19 health outcome with the driving factors has been investigated by employing local (GWR) model.
- The potential socio-economic, demographic, and environmental driving factors for COVID-19 health dynamics has been explored.
- The results of the local (GWR) model with the global (OLS) model are compared to validate its suitability.

8.2 Existing Works

Many studies have been performed on geographical analysis of COVID-19 disease [279–282]. For instance, in [279], the authors present a number of analytical mappings to track coronavirus outcomes and related events around the world. In [282], multi-source data were used to develop indicators for evaluating the COVID-19 outbreak. A rigorous experimental study on 736 fine-grain grids was performed utilizing the GeoDetector and decision tree approaches. In literature, geographical analysis was conducted for mapping data to the various administrative levels (districts, provinces, etc.) in countries like America [283], Iran [284], Brazil [285], Spain [286], Italy [287], South Korea [288], etc. In [287], the authors aim to model and forecast the risk of COVID-19 infection, demonstrating the effects of spatial diffusion. Specifically, they used spatiotemporal modeling to examine the pattern of COVID-19 outbreaks over time and place. Furthermore, a number of recent investigations have been conducted to examine person-specific risk factors for health outcomes associated with COVID-19 [269–271]. Furthermore, some literature examine how sociodemographic

characteristics affect the total number of deaths caused by the coronavirus [273,289]. For example, the geographic association between the sociodemographic variables and COVID-19 incidence and death across Europe was evaluated in [273]. This study has demonstrated the heterogeneity of the detected COVID-19 incident cases and mortality across Europe.

8.3 Problem Statement

The chapter is investigating the factors affecting the geographical variation of COVID-19 mortality rates along with the varying associations across districts in India. Let us consider we are given some district-level potential socio-economic, demographic, and environmental pollution factors as the independent variables. Also consider we have COVID-19 mortality rate per district as the dependent variable. Our objective is to is to model a Geographically Weighted Regression (GWR) to explore how the relationships between these factors and COVID-19 mortality rates vary geographically across India.

8.4 Proposed Approach

8.4.1 Data description

The geographical variabilities of COVID-19 deaths are modeled based on the district-level data across India. Note that, the COVID-19 mortality data are acquired for more than 400 districts in India. The geographical distributions of COVID-19 deaths are shown in Fig 8.1. The largest number of COVID-19 deaths are observed in the districts of the state Maharashtra. A total of 9 among 28 states contains at least one district that reports more than 1000 COVID-19 deaths. Table 8.1 summarizes all the raw datasets, their descriptions, the sources including the links from where these data can be found, and potential factors (independent variables) that are extracted from the raw datasets.

Datasets

Three raw datasets are mainly utilized to investigate geographically varying relationships of COVID-19 deaths with different environmental, demographic, and socio-economic factors. The first dataset includes district wise COVID-19 death counts in India. The cumulative number of COVID-19-related deaths for each district is collected up to February 24, 2021, from the COVID19INDIA website (https://www.covid19india.org/). COVID19INDIA is a crowdsourced initiative to document the COVID-19 data from the states and union territories of India. In this study, the district-level COVID-19 death count is considered as the dependent variable. The second dataset pertaining to environmental pollution includes the daily concentration of different air pollutants (e.g. $PM_{2.5}$, SO_2, NO_2, etc.). The concentration of air pollutants (from January 2016 to January 2020) for a total of 130 monitoring stations are obtained from the Central Pollution Control Board (CPCB [290]), INDIA. The third dataset

contains socio-economic and demographic data that may have an association with COVID-19 mortality. The district-level socio-economic and demographic data are obtained from the last census in India that was conducted in 2011.

Additionally, the district-level data of each district needs to be linked with the GPS co-ordinate of the centroid of that district. The dataset containing GPS coordinates of the districts of India are collected from Kaggle (https://www.kaggle.com/sirpunch/indian-census-data-with-geospatial-indexing).

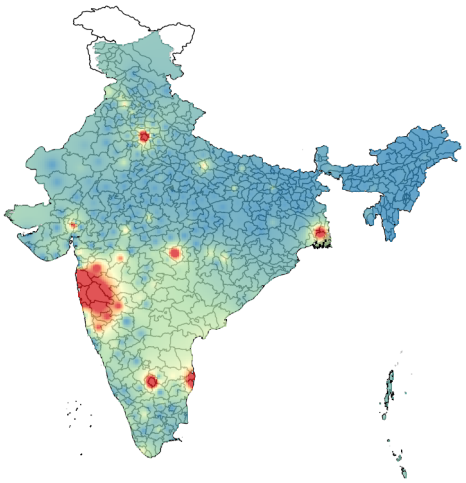


FIGURE 8.1: Geographical distribution of COVID-19 deaths across India. The spatially continuous distribution map is generated in QGIS (https://qgis.org/en/site/) by using Inverse Distance Weighting (IDW) interpolation.

TABLE 8.1: A summary of datasets

Dataset	Dataset description	Source	Variable name	Variable Explanation
COVID-19 data	COVID-19 data from the states and union territories of India up to February 24, 2021.	(i) COVID19INDIA website (https://www.covid19india.org/) (ii) Ministry of Health and Family Welfare, Government of India (https://www.mohfw.gov.in/)	*COVID19_Death*	District-level COVID-19 death count up to February 24, 2021
Census data	Contains district wise socioeconomic and demographic data of India	India census, 2011 (https://censusindia.gov.in/)	*Tot_Population*	District-level total population
			HH_Above_8_P	District-level count of total number of households with at least 9 persons
			Growth_Rate	District-level rate at which population increases.
			Sex_Ratio	District-level count of the number of females per 1000 males
			Age_Abv_50	District-level count of total number of persons with age 50 years or more
			HH_With_TCMC	District-level count of total number of households having TV, Computer (or laptop), Mobile phone, and Car.
			Higher_Edu	District-level count of total number of persons having higher education
			P_Urb_Pop	District level percentage of urban population
Environmental air pollution data	Concentration of air pollutants (from January, 2016 to January, 2020) for a total of 130 monitoring stations	(i) Central Pollution Control Board (CPCB), Government of India (https://cpcb.nic.in/)	$PM_{2.5}$	District-level exposure to $PM2.5$, averaged across the period 2016 – 2020
			NO_2	District-level exposure to NO_2, averaged across the period 2016 – 2020
			SO_2	District-level exposure to SO_2, averaged across the period 2016 – 2020

TABLE 8.2: VIFs with all the 11 explanatory variables

Variable	VIF
Tot_Population	7.92
Growth_Rate	1.24
Sex_Ratio	1.30
HH_With_TCMC	3.91
HH_Abv_8_P	12.4
Higher_Edu	3.12
$PM_{2.5}$	2.65
Age_Abv_50	3.47
P_Urb_Pop	1.93
SO_2	1.20
NO_2	1.52

TABLE 8.3: VIFs after removing the variable *HH_Abv_8_P*

Variable	VIF
Tot_Population	3.93
Growth_Rate	1.21
Sex_Ratio	1.32
HH_With_TCMC	2.96
Higher_Edu	2.94
$PM_{2.5}$	2.35
Age_Abv_50	2.87
P_Urb_Pop	1.80
SO_2	1.19
NO_2	1.54

Data preparation

From the raw datasets, a total of eleven potential demographic, socioeconomic, and environmental pollution related factors (see Table 8.1) are selected to explain the district-level geographical variation of COVID-19 mortality. The district-level demographic and socioeconomic factors that are selected in this study are: population; households with at least 9 persons; growth rate; sex ratio; persons with age 50 years or more; households having TV, computer (or laptop), mobile phones and car; number of persons having higher education; the percentage of the urban population. On the other hand, the environmental pollution related variables that are selected are as follows: $PM_{2.5}$ exposure; NO_2 exposure; SO_2 exposure.

The district-level long-term exposure to three air pollutants namely $PM2.5$, NO_2, and SO_2 are calculated from the raw data of 130 pollution monitoring stations. The mean concentration of each of the above-mentioned air pollutants of all the 130 monitoring stations is computed for the period 2016-2020. For each pollutant, the computed values are spatially aggregated by averaging the values of all monitoring stations of a district. If a district doesn't contain any monitoring stations, then it's exposure to that pollutant is computed using Nearest Neighbour interpolation (NNI).

A multicollinearity verification is performed via the Variance Inflation Factor (VIF) to remove unnecessary redundancy among the explanatory variables. VIF can be expressed as follows (Eq. 8.1):

$$VIF^k = \frac{1}{1 - R_k^2} \tag{8.1}$$

$$R_k^2 = 1 - \frac{SSE_k}{SST_k} \tag{8.2}$$

where, R_k^2 denotes the coefficient of determination that is computed by regressing the k^{th} variable on remaining explanatory variables. The mathematical expression for R_k^2 is given in Eq. 8.2. Here, SSE_k and SST_k denote the sum of squares of total variation and sum of squares

of errors respectively. Firstly, regression analysis is conducted among all the 11 explanatory variables to compute the VIFs that are shown in Table 8.2. It is observed that the variable *HH_Abv_8_P* has high Variance Inflation Factor (VIF=12.4). Now, if VIFs are larger than 10, it indicates that there is multicollinearity [291]. Eventually, the variable *HH_Abv_8_P* is removed from the set of explanatory variables. After that, the regression is again performed on the remaining 10 variables, with the VIFs given in Table 8.3. Now, it is observed that no VIF exceeds 10 eventually this set of 10 variables can be used for model building.

8.4.2 Modeling spatial relationship

In this chapter, the OLS (Ordinary Least Square) and GWR (Geographically Weighted Regression) models are utilized to determine the geographical relationship of COVID-19 mortality with potential risk factors.

The OLS method generally attempts to understand the global relationships between the dependent and independent variables. In this case, the regression and its parameters are unchanged over the geographic space. Mathematically, Eq. 8.3 represents a global regression model as follows:

$$\mathcal{Y}_i = \eta_0 + \sum_{k=1}^{n} \eta_k \mathcal{X}_{ik} + \delta_i \tag{8.3}$$

where, $\mathcal{Y}_i$ denotes the dependent or response variable; $\mathcal{X}_{ik}$ is the i^{th} observation of k^{th} independent variable; η_k the global regression coefficient for k^{th} independent variable; η_0 represent the intercept parameter; and δ_0 denotes the error term.

GWR technique extends the global regression (Eq. 8.3) by enabling local parameter estimation [91]. It allows regression coefficients to be a function of geographical location. In other words, the regression coefficients are quantified independently in different geographical locations. A GWR model (Eq. 8.4) can be represented as follows:

$$\mathcal{Y}_i = \xi_{i0} + \sum_{k=1}^{n} \xi_k(\mu_i, \nu_i)\mathcal{X}_{ki} + \delta_i \tag{8.4}$$

where, $\mathcal{Y}_i$, $\mathcal{X}_{ki}$, and δ_i denote the dependent (or response) variable, k^{th} independent (or predictor) variable, and error at location i respectively; (μ_i, ν_i) denotes coordinates of location i; $\xi_k(\mu_i, \nu_i)$ represent local coefficient for k^{th} predictor at location i. Note that, GWR model allows regression parameters to vary continuously across the geographic space. For each location i, a set of regression parameters is estimated. The estimation of parameters can be performed as follows:

$$\widehat{\xi}(\mu, \nu) = (\mathscr{X}^T \mathscr{W}(\mu, \nu)\mathscr{X})^{-1} \mathscr{X}^T \mathscr{W}(\mu, \nu)\mathcal{Y} \tag{8.5}$$

where, $\mathscr{X}$ denotes a matrix containing the values of independent variables and a column of all 1s; $\mathcal{Y}$ represents a vector of values of the dependent variable; $\widehat{\xi}(\mu, \nu)$ is a vector of local regression parameters; $\mathscr{W}(\mu, \nu)$ is a diagonal matrix whose diagonal elements represent the

geographical weighting of the observations for regression location. The weights in $\mathscr{W}(\mu,\nu)$ assigns greater weights to the observations that are closer to the regression point than the observations that are farther away. In this work, the weights are computed using a Gaussian kernel function which is defined as follows:

$$\begin{cases} w_{ij} = exp\Big[-\frac{1}{2}\Big(\frac{D_i^j}{B}\Big)^2\Big], & \text{if } D_i^j \leq B \\ w_{ij} = 0, & \text{otherwise} \end{cases} \tag{8.6}$$

where, B represents the bandwidth and D_i^j denotes the distance between the regression point i and the location of observation j. Note that, the bandwidth can be defined either by a fixed number of closest neighbors (known as adaptive bandwidth) or by a fixed distance (known as fixed bandwidth). Golden Section search [292] is utilized to find the optimum size of the bandwidth for GWR.

Performance metrics

The performance of the models are assessed by three metrics namely R^2, adjusted R^2, and AICc. Here, AICc is a corrected version of the Akaike Information Criterion (AIC). AICc can be defined as follows [91]:

$$AICc = N\ ln(2\pi)\ +\ 2N\ ln(\hat{\sigma})\ +\ N \times \left(\frac{N + tr(S)}{N - 2 - tr(S)}\right) \tag{8.7}$$

where, N denotes the sample size, S is the hat matrix, $tr(S)$ denotes the trace of S, and $\hat{\sigma}$ represents the estimated standard deviation of the error term. AICc denotes model's accuracy and lower AICc indicates better model quality. It is usually used to find the best-fit model. The value of R^2 represents the ability of a model to explain the variance in the dependent variable and therefore a larger R^2 signifies the better performance of the model. It is computed from the estimated and the actual values of the dependent variable. Moreover, Moran's I index is computed to investigate the spatial autocorrelation of the model residuals. Mathematically, it is defined as follows:

$$I = \frac{N\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\Big(\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\Big)\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{8.8}$$

where, N denotes total number of observations, y_i and y_j are variable values at location i and j respectively,$\bar{y}$ represents the mean value, and w_{ij} denotes a weight between location i and j. The value of Moran's I index can vary between -1 (perfect dispersion) to +1 (a perfect positive autocorrelation)). Note that, a zero value indicates perfect spatial randomness.

Model Building

Here, a step-wise GWR model selection using AICc is presented that can be utilized to investigate geographically varying associations of COVID-19 mortality with various driving

factors. The following are the steps to build an appropriate GWR model [293].

- *Step 1:* Suppose there are n explanatory variables (in our case $n = 10$). For each of the explanatory variables, fit a separate GWR model by regressing that variable against the $COVID19_Death$ variable. Compute AICc for each of the $n = 10$ models. Find the model that generates the lowest AICc and permanently include the corresponding explanatory variable in subsequent model building.
- *Step 2:* Subsequently select a variable from the remaining $(n-1)$ variables, build a model with the permanently included variables along with the newly selected variable. Find the explanatory variable that produces the lowest AICc and permanently include it in subsequent model building. Set $n = n-1$.
- *Step 3:* Repeat Step 2 until it is observed that there is no reduction in AICc.

The above-mentioned steps are carried out using MGWR 2.2 software [294]. When calibrating the GWR, an adaptive bisquare spatial kernel is applied. Moreover, in order to select an optimal bandwidth, the Golden Section search [292] is employed. Fig. 8.2 shows the changes in AICc during the step-by-step selection of explanatory variables for model building. It is observed that after the inclusion of a total of five variables, the AICc values start increasing when further new variables are included. Note that, both a global (OLS) model and a local (GWR) model are calibrated with these five explanatory variables.

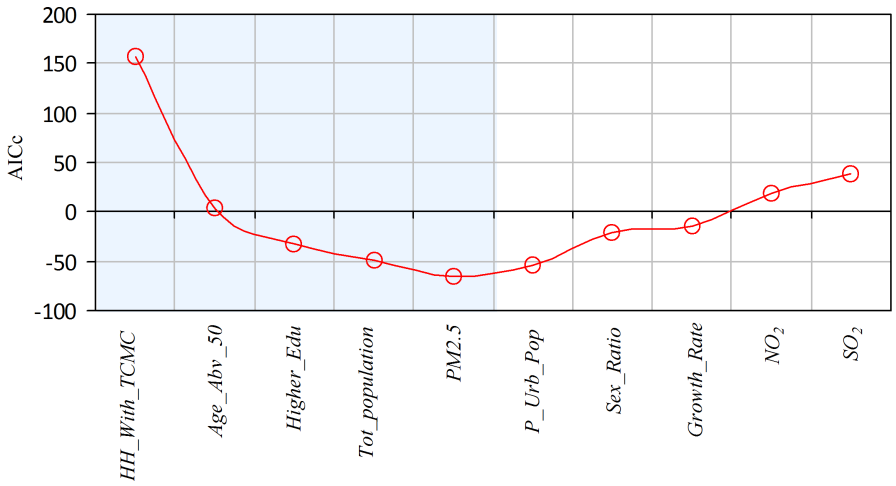


FIGURE 8.2: Stepwise variable selection for geographically weighted regression (GWR).

8.5 Results

In this section, firstly the performance of the global model (OLS) and local model (GWR) are discussed. Next, the geographically varying relationships of COVID-19 mortality with different factors are presented.

8.5.1 Performance of OLS and GWR Model

A detailed summary of the OLS model is presented in Table 8.4. The variables *Tot_population*, *HH_With_TCMC*, *Age_ABV_50*, and $PM_{2.5}$ returns significant t values of 2.91, 12.114, -1.225 and -2.485 respectively. Moreover, the Moran's I of the residuals of the global OLS model are also analysed. It is found that there is significant spatial autocorrelation (Moran's I = 0.348 and $p < 0.01$). The assumptions of OLS estimation are violated as there exist dependent residuals. Eventually, the GWR model is utilized to show the geographical variations of the relationships with different factors. A detailed summary of the GWR model for the local parameter estimates is presented in Table 8.5.

TABLE 8.4: Summary of the global model (OLS) for various socioeconomic, demographic, and environmental pollution related factors.

Variable	Coef. Est	Est Err	t statistic	p-value
Intercept	0.000	0.026	0.000	1.000
Tot_population	0.322	0.11	2.91	0.005*
HH_With_TCMC	0.675	0.045	12.114	0.000*
Age_Abv_50	0.125	0.102	-1.225	0.050*
Higher_Edu	0.136	0.073	1.851	0.064
$PM_{2.5}$	-0.079	0.032	-2.485	0.013*

* Significant at 0.05

TABLE 8.5: Summary of the local model (GWR) for various socioeconomic, demographic, and environmental pollution related factors.

Variable	Mean	STD	Min	Median	Max
Intercept	0.413	0.292	-0.789	0.043	1.134
Tot_population	0.152	0.604	-2.525	0.126	1.385
HH_With_TCMC	0.520	0.382	-0.258	0.408	2.550
Age_Abv_50	0.317	0.648	-1.016	0.189	2.861
Higher_Edu	-0.038	0.381	-1.566	-0.025	1.488
$PM_{2.5}$	0.101	0.352	-1.721	0.015	0.842

The performance of OLS and GWR model in terms of R^2, $Adj\ R^2$, and AICc are also provided in Table 8.6. Moreover, Fig. 8.3 (a) and Fig. 8.3 (b) show the scatter plots between predicted and observed COVID-19 death count using the global OLS and GWR models. These figures indicate that the GWR model resulted in a better fit as compared to the global OLS model. This is because, in the case of GWR, the predicted values are closely distributed along the 1:1 line relative to the observed values. The global model explains only 71.9% of the variance of district-level COVID-19 deaths which is increased to 97% if the model is calibrated as GWR by taking into account the local impact of the explanatory variables. Comparing the models in terms of AICc, show that the model fit is greatly enhanced by reducing the value of AICc from 655.835 (OLS model) to -66.42 (GWR model). Moreover,

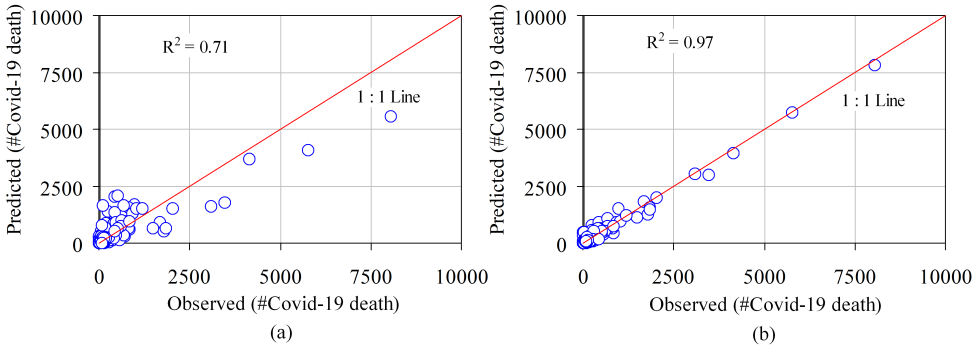


FIGURE 8.3: Scatter plot of the observed and the predicted COVID-19 death count using (a) global OLS model and (b) GWR model.

the verification of Moran's I of the residuals of the GWR model indicates that the residuals are randomly distributed (Moran's I=-0.0395 and $p < 0.01$). In other words, the residuals don't have any significant spatial autocorrelation and eventually, it shows the suitability of GWR over the global model (OLS).

TABLE 8.6: Performance comparison OLS and GWR models in terms of three performance metrics: AICc, R^2, and Adjusted R^2.

Performance metrics	OLS	GWR
AICc	655.835	-66.42
R^2	0.719	0.97
$Adj\ R^2$	0.715	0.964

8.5.2 Geographically varying relationships between COVID-19 deaths and the driving factors

The geographical distribution of R^2 is presented in Fig. 8.4 that shows it varies within a range $0.42 - 0.97$. It is found that more than 86% of local R^2 values are larger than 0.60 and almost 68% of R^2 values are within the range $0.80 - 0.97$. Note that, very high R^2 values are mainly observed in the western and the eastern regions of India. Moreover, low and moderate R^2 values are mainly distributed over the northern and the southern part of India.

Now, the geographical distribution of local coefficient estimates of the GWR model is provided in Fig. 8.5 to further reveal the relationship of the explanatory variables with the COVID-19 deaths. It mainly facilitates understanding of the complex relationship that varies over the geographic space. The results of GWR in Fig. 8.5 not only present positive or negative relationships but also show whether the relationship is strong or weak. A positive relationship indicates that the COVID-19 deaths tend to increase as the value of specific explanatory variable increases. A negative relationship indicates that the COVID-19 deaths tend to decrease as the value of specific explanatory variable increases. Moreover, larger values of a coefficient denote a stronger relationship. In the maps of Fig. 8.5, the regions having deep red shade denote regions in which the specific variable has a strong positive influence (i.e. strong positive relationship) on COVID-19 deaths.

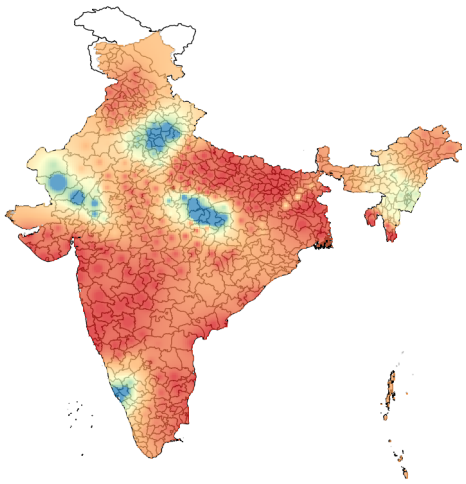


FIGURE 8.4: Geographical distribution of R^2 values for geographically weighted regression (GWR) model.

As shown in Fig. 8.5 (a), the GWR model produces local intercept that can vary within the range -0.78 to 1.13 with a mean of 0.013. In Fig. 8.5 (b), the regions with deep red color (mainly the state of West Bengal) denote those areas where the variable *HH_With_TCMC* has a strong positive relationship with COVID-19 death. The variable *Higher_Edu* is a strong predictor (See Fig. 8.5 (c)) for COVID-19 death in some parts of western India (mainly the state of Gujarat), southern India (mainly the states of Tamil Nadu and Kerala), and Eastern India (mainly the state of West Bengal). On the other hand, in the southern and the south-western part of India, a positive relationship between population and COVID-19 death is found (see Fig. 8.5 (d)). However, in some regions of central and western India (the states of Madhya Pradesh and Gujarat), a strong negative relationship between population and COVID-19 death is also observed. Fig. 8.5 (e) shows that mainly in the western part of India there is a strong positive relationship between $PM_{2.5}$ and COVID-19 death, whereas in the other parts of India there is no such strong relationship. The explanatory variable *Age_Abv_*50 shows a positive relationship in central, eastern, and northern parts of India (see Fig. 8.5 (f)).

Moreover, Table 8.7 represents the district-level results of the local model (GWR) for some of the districts that are severely affected by COVID-19 disease. The local R^2 values revealed district-level variability in GWR model performance. Specifically, the local R^2 values could be helpful here to see where geographically weighted regression predicts well and where it predicts poorly. It is observed that the GWR model yields high local R^2 value for most of the heavily affected districts. For instance, very high local R^2 values are found for the following districts: Pune ($R^2 = 0.982$), Thane ($R^2 = 0.986$), Lucknow ($R^2 = 0.984$), Chittor ($R^2 = 0.973$), Nasik ($R^2 = 0.986$), Solapur ($R^2 = 0.974$), Kolhapur ($R^2 = 0.972$), Sangli ($R^2 = 0.973$), Satara ($R^2 = 0.978$), Latur ($R^2 = 0.967$), Mumbai ($R^2 = 0.981$), Kolkata ($R^2 = 0.987$), Chennai ($R^2 = 0.978$), Jalgaon ($R^2 = 0.966$), Nanded ($R^2 = 0.960$). On the other hand, moderate local R^2 values are found for Dharwad ($R^2 = 0.948$), Nagpur ($R^2 = 0.955$), Srikakulam ($R^2 = 0.920$), Ludhiana ($R^2 = 0.921$), Guntur ($R^2 = 0.95$), Kurnool ($R^2 = 0.856$), Coimbatore ($R^2 = 0.934$), West Godavari ($R^2 = 0.95$), Anantapur ($R^2 = 0.886$), Bhopal

($R^2 = 0.916$), and Krishna ($R^2 = 0.937$). The lowest R^2 values are observed for the following districts: Hassan ($R^2 = 0.683$), Indore ($R^2 = 0.798$), and Jaipur ($R^2 = 0.797$). Note that, for most of the highly COVID-19-affected districts, the variables $PM_{2.5}$ and HH_With_TCMC are usually exhibited positive relationships in regression modeling. On the other hand, the variable $Higher_Edu$ usually exhibits negative relationships for most of the highly affected districts.

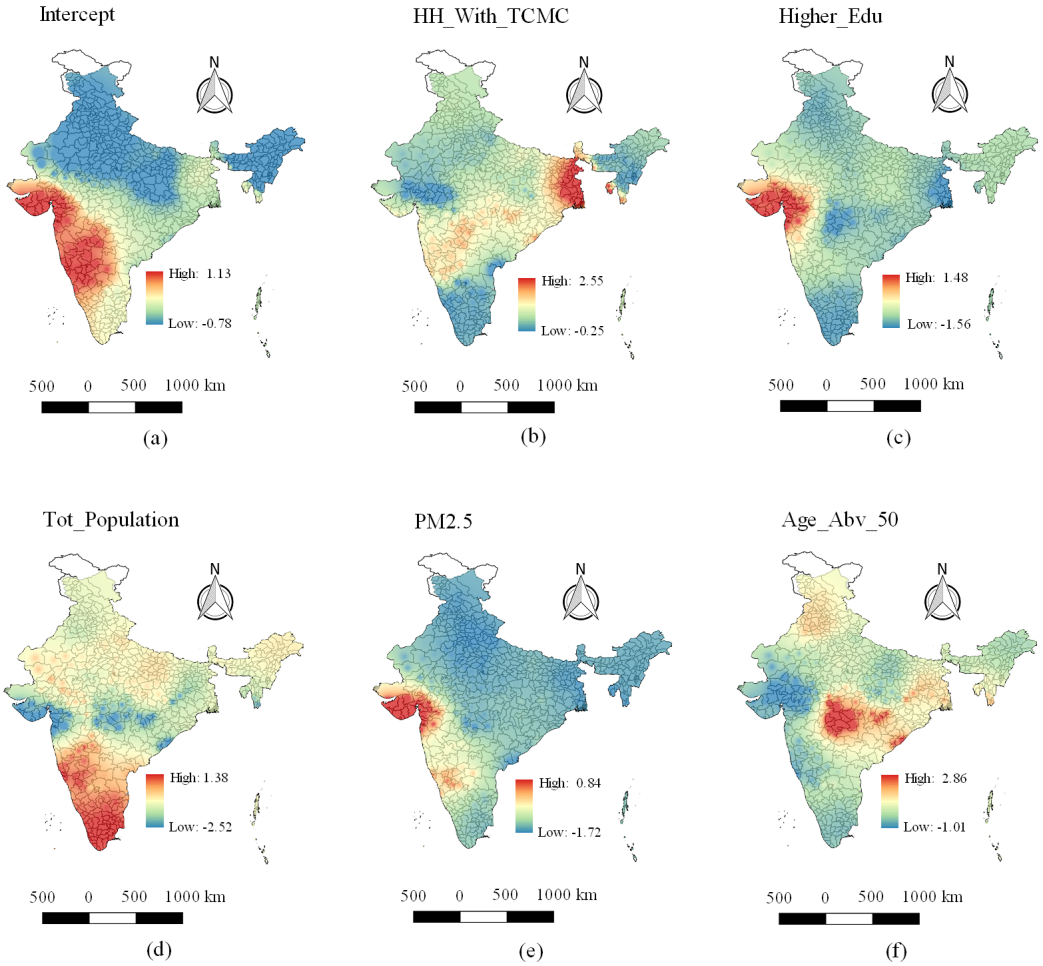


FIGURE 8.5: Local parameter estimates of geographically weighted regression (a) Intercept (b) HH_With_TCMC (c) $Higher_Edu$ (d) $Tot_population$ (e) $PM2.5$ and (f) Age_Abv_50.

8.5.3 Discussion

In order to better understand how different driving factors influence the overall fatalities caused by COVID-19, the geographical distribution of COVID-19-related deaths are investigated. The highest number of COVID-19-related deaths are found primarily in the western part of India (Pune, Thane, Mumbai, Nagpur, Nashik, Raigad, Jalgaon, Kolhapur, Sangli, Satara, Solapur, Ahmedabad, Surat). On the other hand, the number of COVID-19-related deaths is relatively low in the northern and eastern parts of India. This study identified considerable geographical variability of COVID-19 deaths and their heterogeneous relationship at the local level with the driving factors in India. More specifically, the utilization of the

TABLE 8.7: Local R^2 and district-level parameter estimates by geographically weighted regression for some of the districts of India that are severely affected by COVID-19 disease.

District	Local R^2	Parameter estimates					
		Intercept	*PM*2.5	*Tot_population*	*HH_With_TCMC*	*Age_Abv_50*	*Higher_Edu*
Pune	0.982	0.219	-0.221	-0.311	0.661	-0.154	0.768
Mumbai	0.981	0.654	0.712	-0.388	0.859	-0.074	0.719
Thane	0.986	0.086	0.806	-0.958	0.716	-0.084	0.897
Chennai	0.978	0.234	0.421	0.689	1.089	-0.496	-0.575
Kolkata	0.987	0.057	-0.055	-0.146	1.114	0.658	-0.152
Nashik	0.986	0.030	0.800	-0.280	0.742	0.079	0.985
Jalgaon	0.966	0.099	0.357	-0.389	0.744	0.284	0.853
Nagpur	0.955	0.088	-0.132	-1.214	0.685	1.588	-0.571
Solapur	0.974	0.318	0.476	0.516	0.685	-0.073	0.241
Kolhapur	0.972	0.238	0.249	0.483	0.791	-0.142	0.244
Surat	0.976	0.223	0.749	-1.189	0.481	-0.127	1.469
Sangli	0.973	0.281	0.399	0.358	0.576	-0.324	0.127
Ludhiana	0.921	-0.037	-0.175	-0.189	0.287	0.423	-0.225
Chittoor	0.973	-0.029	0.102	0.562	0.128	0.117	-0.286
East Godavari	0.959	0.159	-0.159	-0.541	1.078	0.458	-0.654
Indore	0.798	0.011	0.062	0.141	-0.180	-0.120	0.323
Guntur	0.954	0.150	0.116	0.274	-0.185	0.247	-0.142
Lucknow	0.984	-0.043	-0.104	0.049	0.389	-0.109	0.284
Satara	0.978	0.264	0.370	0.196	0.793	-0.222	0.482
Kurnool	0.856	0.229	0.392	0.366	0.371	0.144	0.096
Madurai	0.950	0.143	0.078	0.629	0.162	-0.059	-0.651
Anantapur	0.886	0.094	0.167	0.226	0.107	0.258	-0.125
Dharwad	0.948	0.193	0.283	0.523	0.741	-0.139	0.105
West Godavari	0.954	0.166	-0.068	0.029	0.322	0.198	-0.284
Coimbatore	0.934	-0.145	0.099	0.578	0.321	-0.127	-0.381
Prakasam	0.953	0.137	0.071	0.324	-0.179	0.314	-0.101
Bhopal	0.916	-0.044	0.037	-0.296	0.475	0.295	0.125
Krishna	0.937	0.151	-0.074	0.221	0.102	0.194	-0.214
Latur	0.967	0.348	0.356	0.571	0.598	0.542	-0.293
Jaipur	0.797	-0.143	-0.131	0.019	0.289	0.225	-0.126
Srikakulam	0.920	0.086	0.108	-1.146	0.854	1.285	-0.276
Nanded	0.960	0.255	0.117	-0.076	0.725	1.108	-0.508
Dhule	0.966	0.074	0.533	-0.746	0.385	-0.278	1.175
Hassan	0.683	0.065	0.214	0.417	0.401	0.112	-0.419

GWR method successfully found the geographically varying relationship of COVID-19 mortality with various potential socio-economic, demographic, and environmental pollution related factors. This study reveals five important local factors are significantly related with district-level COVID-19 deaths as follows: (i) population (ii) $PM_{2.5}$ level (iii) households having TV, computer (or laptop), mobile phones and car (iv) persons with age 50 years or more (v) number of persons having higher education. Furthermore, this study also validates the effectiveness of local parameter estimation by comparing the global OLS method with the local GDR method. To the best of our knowledge, this is the first study that explores geographically varying relationships of COVID-19 deaths with various potential driving factors in India.

Rigorous analyses are performed to demonstrate the shortcomings of global technique (OLS) as compared to the local technique (GWR) in terms of several performance metrics. The OLS model only explains 71.9% of the variance of district-level COVID-19 deaths. It is found that the predictive efficiency and model accuracy are further enhanced by implementing the GWR method. The GWR model explains 97% of the variance of district-level

COVID-19 deaths. Moreover, Moran's I index verifies that no significant spatial autocorrelation is present in the residuals of the GWR model. Note that, a key advantage of such a local method is its capability to visualize the geographically varying heterogeneous relationships between the dependent and the independent variables. In other words, it enables us for a better understanding of relationships based on geographical contexts and study area's known features.

The findings of this study reveal that there are strong positive relationships of COVID-19 deaths with the explanatory variables $PM2.5$ and $Tot_population$ across the regions of the COVID-19 death hotspots in the western part of India. The positive association of COVID-19 deaths with long term exposure of $PM_{2.5}$ is consistent with the previous works [295, 296]. Note that, long-term $PM_{2.5}$ exposure is substantially associated with some of the comorbidities (e.g. chronic lung disease, cardiovascular disease, etc.) that may lead to COVID-19 deaths [297, 298]. Similarly, a positive association between COVID-19-related deaths and $Tot_population$ is also observed in other studies [273, 299]. However, the reverse association is found for these two variables ($PM_{2.5}$ and $Tot_population$) in the other parts of India. The explanatory variable HH_With_TCMC is found to be an important factor that may be a measure of the number of households with the upper class and rich people. A strong positive relationship is observed between HH_With_TCMC and COVID-19 death in the hotspots of eastern and western parts of India (Kolkata, North 24 Parganas, Pune, Thane, Surat, Nagpur, etc.). Note that, in those hotspots, the value of HH_With_TCMC is substantially high. An interesting observation reveals that a strong negative relationship exists between COVID-19 death and $Higher_Edu$ in the eastern, central, and southern parts of India. It is expected that the higher educated people are well aware of the symptoms and the complications of COVID-19 that may lead to the fewer number of fatalities in those regions. Now, in some regions of the south-eastern part of India, the number of COVID-19 deaths is also seen to be high. In those regions, significant positive relationships are found between $COVID-19$ deaths and $Tot_population$, whereas significant negative relationships are observed for the variable $Higher_Edu$.

This research work inherits certain shortcomings that need to be resolved in future research. For instance, there may have high possibilities of under-reporting in COVID-19 death counts that may introduce bias in the study [300]. Moreover, due to data unavailability, we were not able to include some significant district-level driving factors in our study, such as health care system quality, number of hospital beds, household income, and poverty data. Despite the above-mentioned shortcomings, this is the first study that explores geographically varying relationships of COVID-19 mortality with different socioeconomic, demographic, and environmental pollution related factors in India. This research work also highlights the significance of the geographically weighted regression in the geographical analysis of the health outcome of COVID-19 disease.

8.6 Summary

COVID-19 pandemic is one of the most serious global public health catastrophe of the century. In this chapter, the geographically varying relationships between COVID-19 deaths and different potential driving factors are assessed across India. The geographical distribution of reported COVID-19 death cases is found to be heterogeneous over India. This heterogeneity in distribution is related to many underlying factors, including demographic, socioeconomic, and environmental pollution related variables in different parts of India. The GWR model makes it possible for the regression coefficients to differ across the geospace, creating geographical patterns about the strength of the relationship. The geographical heterogeneity and non-stationary of the relationships between COVID-19 deaths and the driving factors are demonstrated by mapping the local parameter estimates. The local parameter estimates reflect the quality of local model fitting and the nature of the association. The local method (GWR) yields better performance with smaller AICc as compared to the global method (OLS).

Publication(s) related to this Chapter:

- *A.I. Middya, and S. Roy, 2021. Geographically varying relationships of COVID-19 mortality with different factors in India. Scientific reports, 11(1), p.7890. (DOI: https://doi.org/10.1038/s41598-021-86987-5)*

Chapter 9

Conclusion

Rapid urbanization and the associated challenges in effective urban planning and decision-making necessitate novel approaches to understanding and managing complex urban dynamics. The advent of Internet of Things (IoT)-based participatory sensing provides an effective way of sensing urban dynamics due to the widespread availability of smartphones among citizens. In this thesis, the effectiveness of participatory sensing in analysing various urban dynamics is assessed. More specifically, this work investigated the predictive modeling for spatiotemporal analysis of various urban dynamics, including air pollution, noise pollution, and health dynamics. We have addressed various research issues in this work including framework design for participatory sensing, privacy-aware solutions, incentive mechanism design, analysis of high-granular spatio-temporal variability in dynamics, exploration of the geographical association between urban dynamics and various risk factors, etc. In the subsequent sections, the major outcomes of the thesis along with the limitations and the future works are presented.

9.1 Major Outcomes

The research outcomes presented in the thesis hold significant importance in the field of predictive modeling for spatio-temporal analysis of participatory and traditional sensing data. This section provides the significance of major findings of the thesis.

1. The conceptualization and development of the unified framework (i.e., JUSense) integrate diverse subsystems rooted in participatory sensing, incorporating smartphones and IoT devices. JUSense brings the independent applications under the same architecture concentrating on the benefits that would result from their mutual interactions. The results show the effectiveness of predictive methodologies in the performance of individual applications (e.g., multiview fusion model for the pollution dynamics) as well as the potential of the combined inferences (e.g., overall pollution map) and recommendations (e.g., less polluted routes).

2. The thesis comes up with novel spatial interpolation methods (GLIDW, I-GLIDW, GLIDW-OK) tailored for participatory-based noise pollution dynamics, addressing scenarios of dense, medium, and highly sparse datasets. Proposed techniques can address diverse scenarios like sparse datasets, better response time, high accuracy, etc.

These techniques demonstrate superior performance, with GLIDW excelling in dense datasets, I-GLIDW in medium sparsity, and GLIDW-OK in highly sparse datasets.

3. A gait-based privacy-aware user authentication system is developed using smartphone accelerometer data for participatory sensing, employing a deep Convolutional Neural Network (CNN) model. The study explores the effectiveness of Fourier transform in feature extraction, enhancing the model's accuracy significantly. The gait recognition approach addresses privacy concerns and contributes to non-contact biometrics, utilizing inbuilt MEMS sensors in smartphones. Major contributions include the proposed CNN model, extensive real-world dataset experiments, and comparative evaluations against baseline methods, affirming its efficacy in user identification for crowd sensing applications.

4. The thesis also introduces TATA (Truthful Double Auction for Task Allocation), a double auction-based incentive mechanism designed for participatory sensing systems. TATA stands out for efficiently addressing incentive challenges, ensuring fairness by satisfying desirable properties such as truthfulness, budget balance, individual rationality, and computational efficiency. Its double-sided auction approach facilitates active engagement from both task providers and executors, validated through rigorous experiments. Simulations using real-world data demonstrate TATA's superiority over existing mechanisms, yielding high utility and satisfaction for both parties.

5. This research explores role of deep learning methodologies in spatio-temporal predictive analytics, specifically in forecasting COVID-19 health dynamics. By examining popular models such as CNN, RNN, Vanilla LSTM, LSTM Autoencoder, and Bidirectional LSTM, the study reveals variations in their performance across different temporal resolutions.

6. Geographically weighted regression (GWR) method is developed that effectively explore the geographical relationships between COVID-19 health dynamics and different driving factors. The findings highlights that the heterogeneity in distribution is related to many underlying factors, including demographic, socioeconomic, and environmental pollution related variables in different parts of India.

9.2 Limitations and Future scope

In this subsection, we explore the limitations and unresolved issues stemming from the present thesis work. We outline specific identified issues that demand further exploration and attention in future research endeavors.

- *Spatial uniformity of samples in participatory sensing*: Participatory sensing cannot ensure spatial uniformity among the samples in an urban setting i.e. data samples may not be uniformly distributed over the space. Our work did not pay attention on the effect of the lack of uniformity of the samples on the models' performance. This is an

issue on which dedicated research is required in the future to achieve better uniformity of samples that are collected using participatory sensing.

- *Context-based interpolation*: The proposed interpolation techniques such as GLIDW, I-GLIDW, and GLIDW-OK do not consider static and dynamic contexts. However, in different urban settings, the dynamics such as noise pollution levels depend on a diverse set of static contexts (e.g. residential area, commercial area, etc.) and dynamic contexts (e.g. traffic flow, crowd flow, etc). There can also be overlapping among the contexts, that may influence the ambient noise pollution levels differently. Hence, as a future work, the context-based interpolation can be considered for map generation so that the spatial variability of ambient noise pollution levels can be better explored.

- *Dependency on users' mobility pattern* In participatory sensing based data collection process, some areas may suffer from a lack of data samples if those areas are not covered by any participant. It is because the area covered by different participants depends on their mobility patterns. In future, the dependency of the analysis on users' mobility pattern can be explored.

- *Quality-aware incentive mechanisms*: The proposed double auction based incentive mechanism TATA does not consider quality-aware payment mechanism for the task executors. Hence, further work can be undertaken to introduce an quality-aware incentive mechanisms for participatory sensing systems.

- *Inclusion of the influencing factors*: In case of exploring geographically varying relationships of health dynamics with the driving factors using GWR, we do not consider the impacts of other influencing factors such as meteorological factors. Moreover, we do not consider time evolution of the dependent and the independent variables considering more time series data for the effective temporal modelling. Considering the above mentioned aspects might be the direction for future studies.

Bibliography

[1] R. Mahtta, M. Fragkias, B. Güneralp, A. Mahendra, M. Reba, E. A. Wentz, and K. C. Seto, "Urban land expansion: the role of population and economic growth for 300 cities," *npj Urban Sustainability*, vol. 2, no. 1, Feb. 2022. [Online]. Available: https://doi.org/10.1038/s42949-022-00048-y

[2] K. C. Seto, B. Güneralp, and L. R. Hutyra, "Global forecasts of urban expansion to 2030 and direct impacts on biodiversity and carbon pools," *Proceedings of the National Academy of Sciences*, vol. 109, no. 40, pp. 16 083–16 088, Sep. 2012. [Online]. Available: https://doi.org/10.1073/pnas.1211658109

[3] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2419–2465, 2019.

[4] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the internet of things," *ACM Transactions on Cyber-Physical Systems*, vol. 2, no. 3, p. 18, 2018.

[5] F. Montori, P. P. Jayaraman, A. Yavari, A. Hassani, and D. Georgakopoulos, "The curse of sensing: Survey of techniques and challenges to cope with sparse and dense data in mobile crowd sensing for internet of things," *Pervasive and Mobile Computing*, 2018.

[6] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, pp. 1–55, 2014.

[7] J. Dutta, C. Chowdhury, S. Roy, A. I. Middya, and F. Gazi, "Towards smart city: sensing air quality in city based on opportunistic crowd-sensing," in *Proceedings of the 18th international conference on distributed computing and networking*, 2017, pp. 1–6.

[8] V. Sivaraman, J. Carrapetta, K. Hu, and B. G. Luxan, "Hazewatch: A participatory sensor system for monitoring air pollution in sydney," in *38th Annual IEEE Conference on Local Computer Networks-Workshops*. IEEE, 2013, pp. 56–64.

[9] C. Gao, F. Kong, and J. Tan, "Healthaware: Tackling obesity with health aware smart phone systems," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Ieee, 2009, pp. 1549–1554.

[10] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: Challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.

[11] D. Kar, A. I. Middya, and S. Roy, "An approach to detect travel patterns using smartphone sensing," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2019, pp. 1–6.

[12] Q. Gu, D. Sacharidis, M. Mathioudakis, and G. Wang, "Inferring venue visits from gps trajectories," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 1–4.

[13] A. Gupta, A. Khare, H. Jin, A. Sadek, L. Su, and C. Qiao, "Estimation of road transverse slope using crowd-sourced data from smartphones," in *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2020, pp. 48–57.

[14] Y. Li, Y.-S. Jeong, B.-S. Shin, and J. H. Park, "Crowdsensing multimedia data: security and privacy issues," *IEEE multimedia*, vol. 24, no. 4, pp. 58–66, 2017.

[15] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2015.

[16] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*. IEEE, 2014, pp. 593–598.

[17] A. I. Middya, P. Dey, and S. Roy, "Iot-based crowdsensing for smart environments," in *Internet of Things for Smart Environments*. Springer, 2022, pp. 33–58.

[18] A. I. Middya, S. Roy, J. Dutta, and R. Das, "Jusense: A unified framework for participatory-based urban sensing system," *Mobile Networks and Applications*, vol. 25, pp. 1249–1274, 2020.

[19] A. I. Middya and S. Roy, "Predicting missing value for ozone data based on optimal site-selection strategy," in *2022 IEEE Calcutta Conference (CALCON)*. IEEE, Dec. 2022. [Online]. Available: http://dx.doi.org/10.1109/CALCON56258.2022.10060138

[20] A. I. Middya, S. Roy, S. Mandal, and R. Talukdar, "Privacy protected user identification using deep learning for smartphone-based participatory sensing applications," *Neural Computing and Applications*, vol. 33, no. 24, pp. 17 303–17 313, Jul. 2021. [Online]. Available: https://doi.org/10.1007/s00521-021-06319-6

[21] A. I. Middya and S. Roy, "Truthful double auction based incentive mechanism for participatory sensing systems," *Peer-to-Peer Networking and Applications*, 2024.

[22] ——, "Spatial interpolation techniques on participatory sensing data," *ACM Transactions on Spatial Algorithms and Systems*, vol. 7, no. 3, pp. 1–32, 2021.

[23] ——, "Spatio-temporal variation of covid-19 health outcomes in india using deep learning based models," *Technological Forecasting and Social Change*, vol. 183, p. 121911, Oct. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.techfore.2022.121911

[24] ——, "Geographically varying relationships of covid-19 mortality with different factors in india," *Scientific Reports*, vol. 11, no. 1, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1038/s41598-021-86987-5

[25] Y. Zheng, "Urban computing: enabling urban intelligence with big data," *Frontiers of Computer Science*, vol. 11, no. 1, pp. 1–3, 2017.

[26] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 349–350.

[27] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele, "Participatory air pollution monitoring using smartphones," *Mobile Sensing*, vol. 1, pp. 1–5, 2012.

[28] K. Sha, G. Zhan, W. Shi, M. Lumley, C. Wiholm, and B. Arnetz, "Spa: a smart phone assisted chronic illness self-management system with participatory sensing," in *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, 2008, pp. 1–3.

[29] "Forecast number of mobile users worldwide," https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/, accessed: 2019-07-22.

[30] A. Tamilin, I. Carreras, E. Ssebaggala, A. Opira, and N. Conci, "Context-aware mobile crowdsourcing," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 717–720.

[31] S. Nittel, "Real-time sensor data streams," *SIGSPATIAL Special*, vol. 7, no. 2, pp. 22–28, 2015.

[32] P. E. Brown, T. Dasu, Y. Kanza, and D. Srivastava, "From rocks to pebbles: Smoothing spatiotemporal data streams in an overlay of sensors," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 5, no. 3, pp. 1–38, 2019.

[33] T. Yabe, K. Tsubouchi, A. Sudo, and Y. Sekimoto, "Predicting irregular individual movement following frequent mid-level disasters using location data from smartphones," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016, pp. 1–4.

[34] S. Karagiorgou, D. Pfoser, and D. Skoutas, "A layered approach for more robust generation of road network maps from vehicle tracking data," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 3, no. 1, pp. 1–21, 2017.

[35] E. K. Tokuda, Y. Lockerman, G. B. Ferreira, E. Sorrelgreen, D. Boyle, R. M. Cesar-Jr, and C. T. Silva, "A new approach for pedestrian density estimation using moving sensors and computer vision," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 6, no. 4, pp. 1–20, 2020.

[36] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby *et al.*, "Activity sensing in the wild: a field trial of ubifit garden," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 1797–1806.

[37] H. Cao, J. Sankaranarayanan, J. Feng, Y. Li, and H. Samet, "Understanding metropolitan crowd mobility via mobile cellular accessing data," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 5, no. 2, pp. 1–18, 2019.

[38] R. Verma, S. Ghosh, N. Ganguly, B. Mitra, and S. Chakraborty, "Smart-phone based spatio-temporal sensing for annotated transit map generation," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 1–10.

[39] S. Hemminki, K. Kuribayashi, S. Konomi, P. Nurmi, and S. Tarkoma, "Crowd replication: Sensing-assisted quantification of human behavior in public spaces," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 5, no. 3, pp. 1–34, 2019.

[40] D. Allemann and M. Raubal, "Towards health-optimal routing in urban areas," in *Proceedings of the Second ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health*, 2013, pp. 56–59.

[41] A. I. Middya, S. Roy, and R. Das, "Spatiotemporal variability analysis of air pollution data from iot based participatory sensing," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–16, 2021.

[42] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *ACM Sigmod Record*, vol. 44, no. 4, pp. 23–34, 2016.

[43] Z. Chi, Y. Wang, Y. Huang, and X. Tong, "The novel location privacy-preserving ckd for mobile crowdsourcing systems," *IEEE Access*, vol. 6, pp. 5678–5687, 2017.

[44] L. Zhang, Y. Li, X. Xiao, X.-Y. Li, J. Wang, A. Zhou, and Q. Li, "Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2735–2743.

[45] E. De Cristofaro and C. Soriente, "Short paper: Pepsi—privacy-enhanced participatory sensing infrastructure," in *Proceedings of the fourth ACM conference on Wireless network security*, 2011, pp. 23–28.

[46] Y. Agarwal and M. Hall, "Protectmyprivacy: detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013, pp. 97–110.

[47] Y.-T. Tsou and B.-C. Lin, "Ppdca: privacy-preserving crowdsourcing data collection and analysis with randomized response," *IEEE Access*, vol. 6, pp. 76 970–76 983, 2018.

[48] C. Lin, D. He, S. Zeadally, N. Kumar, and K.-K. R. Choo, "Secbcs: a secure and privacy-preserving blockchain-based crowdsourcing system," *Science China Information Sciences*, vol. 63, pp. 1–14, 2020.

[49] D. Ding, M. Conti, and A. Solanas, "A smart health application and its related privacy issues," in *2016 Smart city security and privacy workshop (SCSP-W)*. IEEE, 2016, pp. 1–5.

[50] H. Xie, L. Kulik, and E. Tanin, "Privacy-aware traffic monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 61–70, 2009.

[51] I. Zavalyshyn, N. O. Duarte, and N. Santos, "Homepad: A privacy-aware smart hub for home environments," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 58–73.

[52] J. M. de Fuentes, L. Gonzalez-Manzano, A. Solanas, and F. Veseli, "Attribute-based credentials for privacy-aware smart health services in iot-based smart cities," *Computer*, vol. 51, no. 7, pp. 44–53, 2018.

[53] R. Attarian and S. Hashemi, "An anonymity communication protocol for security and privacy of clients in iot-based mobile health transactions," *Computer Networks*, vol. 190, p. 107976, 2021.

[54] D. Mocrii, Y. Chen, and P. Musilek, "Iot-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet of Things*, vol. 1, pp. 81–98, 2018.

[55] E. Kanjo, "Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping," *Mobile Networks and Applications*, vol. 15, pp. 562–574, 2010.

[56] J. Picaut, A. Can, N. Fortin, J. Ardouin, and M. Lagrange, "Low-cost sensors for urban noise monitoring networks—a literature review," *Sensors*, vol. 20, no. 8, p. 2256, 2020.

[57] B. Jia, H. Gong, Z. Zong, T. Zhou, T. Baker, A. Al-Shamma'a, and Y. Jia, "An incentive mechanism in expert-decision-based crowdsensing networks," *Applied Soft Computing*, vol. 122, p. 108834, 2022.

[58] Z. Wang, J. Li, J. Hu, J. Ren, Q. Wang, Z. Li, and Y. Li, "Towards privacy-driven truthful incentives for mobile crowdsensing under untrusted platform," *IEEE Transactions on Mobile Computing*, 2021.

[59] Z. Song, E. Ngai, J. Ma, X. Gong, Y. Liu, and W. Wang, "Incentive mechanism for participatory sensing under budget constraints," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2014, pp. 3361–3366.

[60] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE transactions on vehicular technology*, vol. 64, no. 9, pp. 4203–4214, 2014.

[61] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2018.

[62] C. T. Chou, N. Bulusu, S. Kanhere *et al.*, "Sensing data market," *Proceedings of Poster Papers*, p. 13, 2007.

[63] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM transactions on networking*, vol. 24, no. 3, pp. 1732–1744, 2015.

[64] N. Jiang, D. Xu, J. Zhou, H. Yan, T. Wan, and J. Zheng, "Toward optimal participant decisions with voting-based incentive model for crowd sensing," *Information Sciences*, vol. 512, pp. 1–17, 2020.

[65] H. Cai, Y. Zhu, Z. Feng, H. Zhu, J. Yu, and J. Cao, "Truthful incentive mechanisms for mobile crowd sensing with dynamic smartphones," *Computer Networks*, vol. 141, pp. 1–16, 2018.

[66] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *International Conference on Pervasive Computing*. Springer, 2010, pp. 138–155.

[67] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, Jun. 2015. [Online]. Available: https://doi.org/10.1145/2746285.2746310

[68] J.-S. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2010, pp. 60–68.

[69] R. Esmaeilyfard and M. Moghisi, "An incentive mechanism design for multitask and multipublisher mobile crowdsensing environment," *The Journal of Supercomputing*, Oct. 2022. [Online]. Available: https://doi.org/10.1007/s11227-022-04852-2

[70] Y. Li, F. Li, L. Zhu, H. Chen, T. Li, and Y. Wang, "Fair incentive mechanism with imperfect quality in privacy-preserving crowdsensing," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 188–19 200, Oct. 2022. [Online]. Available: https://doi.org/10.1109/jiot.2022.3164664

[71] Y. Liu, H. Li, G. Zhao, and J. Duan, "Reverse auction based incentive mechanism for location-aware sensing in mobile crowd sensing," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[72] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *2013 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 2013, pp. 76–84.

[73] J. Sun and D. Wu, "Online truthful incentives for heterogeneous and k-submodular crowdsensing," *IEEE Systems Journal*, pp. 1–10, 2022. [Online]. Available: https://doi.org/10.1109/jsyst.2022.3217803

[74] R. Zhou, R. Zhang, Y. Wang, H. Tan, and K. He, "Online incentive mechanism for task offloading with privacy-preserving in UAV-assisted mobile edge computing," in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. ACM, Oct. 2022. [Online]. Available: https://doi.org/10.1145/3492866.3549715

[75] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, Apr. 2015. [Online]. Available: https://doi.org/10.1109/infocom.2015.7218676

[76] T. Ni, Z. Chen, G. Xu, S. Zhang, and H. Zhong, "Differentially private double auction with reliability-aware in mobile crowd sensing," *Ad Hoc Networks*, vol. 114, p. 102450, Apr. 2021. [Online]. Available: https://doi.org/10.1016/j.adhoc.2021.102450

[77] S. Chen, M. Liu, and X. Chen, "A truthful double auction for two-sided heterogeneous mobile crowdsensing markets," *Computer Communications*, vol. 81, pp. 31–42, May 2016. [Online]. Available: https://doi.org/10.1016/j.comcom.2015.11.010

[78] C. Chen and Y. Wang, "Sparc: Strategy-proof double auction for mobile participatory sensing," in *2013 International Conference on Cloud Computing and Big Data*. IEEE, 2013, pp. 133–140.

[79] X. Deng, P. Goldberg, B. Tang, and J. Zhang, "Revenue maximization in a bayesian double auction market," *Theoretical Computer Science*, vol. 539, pp. 1–12, 2014.

[80] H. Jin, L. Su, and K. Nahrstedt, "Centurion: Incentivizing multi-requester mobile crowd sensing," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[81] G. Guillaume, A. Can, G. Petit, N. Fortin, S. Palominos, B. Gauvreau, E. Bocher, and J. Picaut, "Noise mapping based on participative measurements," *Noise Mapping*, vol. 3, no. 1, 2016.

[82] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phones," in *Information technologies in environmental engineering*. Springer, 2009, pp. 215–228.

[83] Y. Zhu, J. Li, L. Liu, and C.-K. Tham, "ical: Intervention-free calibration for measuring noise with smartphones," in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2015, pp. 85–91.

[84] J. Buelvas, D. Múnera, D. P. Tobón V, J. Aguirre, and N. Gaviria, "Data quality in iot-based air quality monitoring systems: a systematic mapping study," *Water, Air, & Soil Pollution*, vol. 234, no. 4, p. 248, 2023.

[85] D. Núñez-Alonso, L. V. Pérez-Arribas, S. Manzoor, J. O. Cáceres *et al.*, "Statistical tools for air pollution assessment: multivariate and spatial analysis studies in the madrid region," *Journal of analytical methods in chemistry*, vol. 2019, 2019.

[86] N. S.-N. Lam, "Spatial interpolation methods: a review," *The American Cartographer*, vol. 10, no. 2, pp. 129–150, 1983.

[87] E. H. Isaaks and R. M. Srivastava, "Applied geostatistics," *(No Title)*, 1989.

[88] P. Aumond, A. Can, V. Mallet, B. De Coensel, C. Ribeiro, D. Botteldooren, and C. Lavandier, "Kriging-based spatial interpolation from measurements for sound level mapping in urban areas," *The Journal of the Acoustical Society of America*, vol. 143, no. 5, pp. 2847–2857, 2018.

[89] H. J. Miller, "Tobler's first law and spatial analysis," *Annals of the Association of American Geographers*, vol. 94, no. 2, pp. 284–289, 2004.

[90] D. C. Wheeler and A. Páez, "Geographically weighted regression," in *Handbook of applied spatial analysis*. Springer, 2010, pp. 461–486.

[91] A. S. Fotheringham, C. Brunsdon, and M. Charlton, *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons, 2003.

[92] G. D. Hutcheson, "Ordinary least-squares regression," *L. Moutinho and GD Hutcheson, The SAGE dictionary of quantitative management research*, pp. 224–228, 2011.

[93] Q. Zhao, E. A. Wentz, S. Fotheringham, S. T. Yabiku, S. J. Hall, J. A. Glick, J. Dai, M. Clark, and H. Heavenrich, "Semi-parametric geographically weighted regression (s-gwr): A case study on invasive plant species distribution in subtropical nepal," 2016.

[94] A. S. Fotheringham, W. Yang, and W. Kang, "Multiscale geographically weighted regression (mgwr)," *Annals of the American Association of Geographers*, vol. 107, no. 6, pp. 1247–1265, 2017.

[95] R. Das, A. I. Middya, and S. Roy, "High granular and short term time series forecasting of pm 2.5 air pollutant-a comparative review," *Artificial Intelligence Review*, pp. 1–35, 2021.

[96] B. T. Ong, K. Sugiura, and K. Zettsu, "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting pm 2.5," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1553–1566, 2016.

[97] A. I. Middya and S. Roy, "Pollutant specific optimal deep learning and statistical model building for air quality forecasting," *Environmental Pollution*, vol. 301, p. 118972, May 2022. [Online]. Available: https://doi.org/10.1016/j.envpol.2022.118972

[98] L. Mohimont, A. Chemchem, F. Alin, M. Krajecki, and L. A. Steffenel, "Convolutional neural networks and temporal CNNs for COVID-19 forecasting in france," *Applied Intelligence*, vol. 51, no. 12, pp. 8784–8809, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s10489-021-02359-6

[99] A. Mustafi, A. I. Middya, and S. Roy, "Fuzzy-based missing value imputation technique for air pollution data," *Artificial Intelligence Review*, vol. 56, no. 2, pp. 1–38, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s10462-022-10168-7

[100] P. Saeipourdizaj, P. Sarbakhsh, and A. Gholampour, "Application of imputation methods for missing values of pm10 and o3 data: Interpolation, moving average and k-nearest neighbor methods," *Environmental Health Engineering and Management Journal*, vol. 8, no. 3, pp. 215–226, 2021.

[101] B. Zhao, L. Yu, C. Wang, C. Shuai, J. Zhu, S. Qu, M. Taiebat, and M. Xu, "Urban air pollution mapping using fleet vehicles as mobile monitors and machine learning," *Environmental Science & Technology*, vol. 55, no. 8, pp. 5579–5588, 2021.

[102] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, 2010, pp. 105–116.

[103] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 323–336.

[104] J. Dutta, P. Pramanick, and S. Roy, "Noisesense: crowdsourced context aware sensing for real time noise pollution monitoring of the city," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2017, pp. 1–6.

[105] C. Benegui and R. T. Ionescu, "Convolutional neural networks for user identification based on motion sensors represented as images," *IEEE Access*, vol. 8, pp. 61 255–61 266, 2020.

[106] A. Angrisano, M. L. Bernardi, M. Cimitile, S. Gaglione, and M. Vultaggio, "Identification of walker identity using smartphone sensors: an experiment using ensemble learning," *IEEE Access*, vol. 8, pp. 27 435–27 447, 2020.

[107] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, and G. Taylor, "Learning human identity from motion patterns," *IEEE Access*, vol. 4, pp. 1810–1820, 2016.

[108] A. Buriro, B. Crispo, F. Delfrari, and K. Wrona, "Hold and sign: A novel behavioral biometrics for smartphone user authentication," in *2016 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2016, pp. 276–285.

[109] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2011, pp. 141–148.

[110] G. Zhang, F. Hou, L. Gao, G. Yang, and L. X. Cai, "Nondeterministic-mobility-based incentive mechanism for efficient data collection in crowdsensing," *IEEE Internet of*

Things Journal, vol. 9, no. 23, pp. 23 626–23 638, Dec. 2022. [Online]. Available: https://doi.org/10.1109/jiot.2022.3190565

[111] X. Chen, L. Zhang, Y. Pang, B. Lin, and Y. Fang, "Timeliness-aware incentive mechanism for vehicular crowdsourcing in smart cities," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3373–3387, Sep. 2022. [Online]. Available: https://doi.org/10.1109/tmc.2021.3052963

[112] Q. Yang, Q. Yuan, L. Yue, and T. Li, "Investigation of the spatially varying relationships of pm2.5 with meteorology, topography, and emissions over china in 2015 by using modified geographically weighted regression," *Environmental Pollution*, vol. 262, p. 114257, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.envpol.2020.114257

[113] A. Alimissis, K. Philippopoulos, C. Tzanis, and D. Deligiorgi, "Spatial estimation of urban air pollution with the use of artificial neural network models," *Atmospheric Environment*, vol. 191, p. 205–213, Oct. 2018. [Online]. Available: http://dx.doi.org/10.1016/j.atmosenv.2018.07.058

[114] J. Dutta, F. Gazi, S. Roy, and C. Chowdhury, "Airsense: Opportunistic crowd-sensing based air quality monitoring system for smart city," in *2016 IEEE SENSORS*. IEEE, 2016, pp. 1–3.

[115] J. Dutta, P. Pramanick, and S. Roy, "Noisesense: Crowdsourced context aware sensing for real time noise pollution monitoring of the city," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Dec 2017, pp. 1–6.

[116] S. Ghosh, J. Dutta, and S. Roy, "Sensedcity: a participatory sensing based approach," in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*. ACM, 2018, p. 16.

[117] B. Bose, J. Dutta, S. Ghosh, P. Pramanick, and S. Roy, "Smartphone based system for real-time aggressive driving detection and marking rash driving-prone areas," in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*. ACM, 2018, p. 27.

[118] J. Dutta, S. Roy, and C. Chowdhury, "Unified framework for iot and smartphone based different smart city related applications," *Microsystem Technologies*, vol. 25, no. 1, pp. 83–96, 2019.

[119] Z. Zhao, J. Fang, W. Ding, and J. Wang, "An integrated processing platform for traffic sensor data and its applications in intelligent transportation systems," in *2014 IEEE World Congress on Services*. IEEE, 2014, pp. 161–168.

[120] R. Li, S. Ruan, J. Bao, and Y. Zheng, "A cloud-based trajectory data management system," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2017, pp. 1–4.

[121] C. Hochreiner, M. Vogler, P. Waibel, and S. Dustdar, "Visp: An ecosystem for elastic data stream processing for the internet of things," in *2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2016, pp. 1–11.

[122] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware qos prediction with neural collaborative filtering for internet-of-things services," *IEEE Internet of Things Journal*, 2019.

[123] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the iot environment," *Wireless Networks*, pp. 1–17, 2019.

[124] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang, "Diagnosing new york city's noises with ubiquitous data," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 715–725.

[125] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1436–1444.

[126] S. Wang, M. J. Paul, and M. Dredze, "Social media as a sensor of air quality and public response in china," *Journal of medical Internet research*, vol. 17, no. 3, 2015.

[127] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 2008, pp. 29–39.

[128] M. Van Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1040–1045.

[129] L. Kuang, X. Yan, X. Tan, S. Li, and X. Yang, "Predicting taxi demand based on 3d convolutional neural network and multi-task learning," *Remote Sensing*, vol. 11, no. 11, p. 1265, 2019.

[130] L. Kuang, S. Tu, Y. Zhang, and X. Yang, "Providing privacy preserving in next poi recommendation for mobile edge computing," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–11, 2020.

[131] "Open weather map api," https://openweathermap.org/api, accessed: 2019-04-22.

[132] "Overpass api," https://wiki.openstreetmap.org/, accessed: 2019-04-22.

[133] E. H. Isaaks and R. M. Srivastava, "An introduction to applied geostatistics," Oxford university press, Tech. Rep., 1989.

[134] P. D. Dumitru, M. Plopeanu, and D. Badea, "Comparative study regarding the methods of interpolation," *Recent advances in geodesy and Geomatics engineering*, vol. 1, pp. 45–52, 2013.

[135] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.

[136] "Gini impurity," https://scikit-learn.org, accessed: 2020-02-02.

[137] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[138] C. Group, "Equivalent continuous sound pressure level," https://www.castlegroup.co.uk/guidance/noise-at-work-assessments/what-is-leq/, 2019, accessed: 2019-05-02.

[139] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, "An interactive-voting based map matching algorithm," in *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*. IEEE Computer Society, 2010, pp. 43–52.

[140] F. Van Diggen, "Gps accuracy: Lies, damn lies and statistics," *GPS World*, no. 1, p. 41, 1999.

[141] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate gps trajectories," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 352–361.

[142] K. M. Shakya, P. Kremer, K. Henderson, M. McMahon, R. E. Peltier, S. Bromberg, and J. Stewart, "Mobile monitoring of air and noise pollution in philadelphia neighborhoods during summer 2017," *Environmental Pollution*, vol. 255, p. 113195, 2019.

[143] P. Apparicio, J. Gelb, M. Carrier, M.-È. Mathieu, and S. Kingham, "Exposure to noise and air pollution by mode of transportation during rush hours in montreal," *Journal of Transport Geography*, vol. 70, pp. 182–192, 2018.

[144] A. Pro, "Air quality monitor (atlanta healthcare)," https://www.airvisual.com/, accessed: 2019-03-30.

[145] F. Semicon, "Air quality monitor," https://www.forbixindia.com/electronics/, accessed: 2019-03-30.

[146] J. Li and A. D. Heap, "Spatial interpolation methods applied in the environmental sciences: A review," *Environmental Modelling & Software*, vol. 53, pp. 173–189, 2014.

[147] K. Gumus and A. Sen, "Comparison of spatial interpolation methods and multi-layer neural networks for different point distributions on a digital elevation model," *Geodetski Vestnik*, vol. 57, pp. 523–543, 09 2013.

[148] S. Patra, A. I. Middya, and S. Roy, "Potspot: Participatory sensing based monitoring system for pothole detection using deep learning," *Multimedia Tools and Applications*, pp. 1–25, 2021.

[149] P. Nath, P. Saha, A. I. Middya, and S. Roy, "Long-term time-series pollution forecast using statistical and deep learning methods," *Neural Computing and Applications*, pp. 1–20, 2021.

[150] C. Katsini, M. Belk, C. Fidas, N. Avouris, and G. Samaras, "Security and usability in knowledge-based user authentication: A review," in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, 2016, pp. 1–6.

[151] R. Spolaor, Q. Li, M. Monaro, M. Conti, L. Gamberini, and G. Sartori, "Biometric authentication methods on smartphones: A survey." *PsychNology Journal*, vol. 14, no. 2, 2016.

[152] D. Gafurov, "A survey of biometric gait recognition: Approaches, security and challenges," in *Annual Norwegian computer science conference*. Annual Norwegian Computer Science Conference Norway, 2007, pp. 19–21.

[153] A. Nambiar, A. Bernardino, and J. C. Nascimento, "Gait-based person re-identification: a survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–34, 2019.

[154] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2010, pp. 1–7.

[155] C. Yam, M. S. Nixon, and J. N. Carter, "Automated person recognition by walking and running via model-based approaches," *Pattern recognition*, vol. 37, no. 5, pp. 1057–1072, 2004.

[156] W. Tan, J. Hsu, and F. Pinn, "Method and system for token-based authentication," Nov. 29 2001, uS Patent App. 09/792,785.

[157] S. Thavalengal and P. Corcoran, "User authentication on smartphones: Focusing on iris biometrics," *IEEE Consumer Electronics Magazine*, vol. 5, no. 2, pp. 87–93, 2016.

[158] R. Subban and D. P. Mankame, "A study of biometric approach using fingerprint recognition," *Lecture notes on software engineering*, vol. 1, no. 2, p. 209, 2013.

[159] Y. Hwang, H.-J. Mun, and J.-W. Lee, "Face recognition system technologies for authentication system-a survey," *Journal of Convergence Society for SMB*, vol. 5, no. 3, pp. 9–13, 2015.

[160] S. Tao, X. Zhang, H. Cai, Z. Lv, C. Hu, and H. Xie, "Gait based biometric personal authentication by using mems inertial sensors," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 5, pp. 1705–1712, 2018.

[161] Y. Chai, J. Ren, R. Zhao, and J. Jia, "Automatic gait recognition using dynamic variance features," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. IEEE, 2006, pp. 475–480.

[162] M. A. R. Ahad, T. T. Ngo, A. D. Antar, M. Ahmed, T. Hossain, D. Muramatsu, Y. Makihara, S. Inoue, and Y. Yagi, "Wearable sensor-based gait analysis for age and gender estimation," *Sensors*, vol. 20, no. 8, p. 2424, 2020.

[163] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3209–3221, 2017.

[164] L. Middleton, A. A. Buss, A. Bazin, and M. S. Nixon, "A floor sensor system for gait recognition," in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*. IEEE, 2005, pp. 171–176.

[165] "Accelerometer bbiometric competition dataset," https://www.kaggle.com/c/accelerometer-biometric-competition/overview, accessed: 2021-02-02.

[166] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019.

[167] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5372–5381.

[168] "Adam optimizer," https://keras.io/api/optimizers/adam/, accessed: 2021-02-02.

[169] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*. Springer, Cham, 2019, pp. 3–33.

[170] "Python," https://www.python.org/downloads/release/python-370/, accessed: 2021-02-02.

[171] C. Nickel, H. Brandt, and C. Busch, "Benchmarking the performance of svms and hmms for accelerometer-based biometric gait recognition," in *2011 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2011, pp. 281–286.

[172] A. Kececi, A. Yildirak, K. Ozyazici, G. Ayluctarhan, O. Agbulut, and I. Zincir, "Implementation of machine learning algorithms for gait recognition," *Engineering Science and Technology, an International Journal*, vol. 23, no. 4, pp. 931–937, 2020.

[173] L.-F. Shi, C.-X. Qiu, D.-J. Xin, and G.-X. Liu, "Gait recognition via random forests based on wearable inertial measurement unit," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.

[174] G. Gasso, "Logistic regression," 2019.

[175] S. Choi, I.-H. Youn, R. LeMay, S. Burns, and J.-H. Youn, "Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification," in *2014 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2014, pp. 1091–1095.

[176] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2013, pp. 19–26.

[177] S. Patra, A. I. Middya, and S. Roy, "PotSpot: Participatory sensing based monitoring system for pothole detection using deep learning," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 25 171–25 195, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s11042-021-10874-4

[178] N. Shirzad-Ghaleroudkhani, Q. Mei, and M. Gül, "A crowdsensing-based platform for transportation infrastructure monitoring and management in smart cities," in *The Rise of Smart Cities*. Elsevier, 2022, pp. 609–624. [Online]. Available: https://doi.org/10.1016/b978-0-12-817784-6.00005-9

[179] D. Kar, A. I. Middya, and S. Roy, "An approach to detect travel patterns using smartphone sensing," in *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, Dec. 2019. [Online]. Available: https://doi.org/10.1109/ants47819.2019.9118073

[180] P. Nath, P. Saha, A. I. Middya, and S. Roy, "Long-term time-series pollution forecast using statistical and deep learning methods," *Neural Computing and Applications*, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s00521-021-05901-2

[181] R. Das, A. I. Middya, and S. Roy, "High granular and short term time series forecasting of pm2.5 air pollutant - a comparative review," *Artificial Intelligence Review*, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s10462-021-09991-1

[182] B. Chandra, A. I. Middya, and S. Roy, "Spatio-temporal prediction of noise pollution using participatory sensing," in *Emerging Technologies in Data Mining and Information Security*. Springer, 2021, pp. 597–607.

[183] Z. Liu, S. Jiang, P. Zhou, and M. Li, "A participatory urban traffic monitoring system: The power of bus riders," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2851–2864, 2017.

[184] H. Yan, Q. Hua, D. Zhang, J. Wan, S. Rho, and H. Song, "Cloud-assisted mobile crowd sensing for traffic congestion control," *Mobile Networks and Applications*, vol. 22, no. 6, pp. 1212–1218, 2017.

[185] J. Takahashi, Y. Kobana, N. Isoyama, Y. Tobe, and G. Lopez, "Ykob: Participatory sensing-based road condition monitoring using smartphones worn by cyclist," *Electronics and Communications in Japan*, vol. 101, no. 4, pp. 3–14, 2018.

[186] D. E. Nunes and V. F. Mota, "A participatory sensing framework to classify road surface quality," *Journal of Internet Services and Applications*, vol. 10, no. 1, p. 13, 2019.

[187] K. Hu, V. Sivaraman, B. G. Luxan, and A. Rahman, "Design and evaluation of a metropolitan air pollution sensing system," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1448–1459, 2015.

[188] L.-Y. Jiang, F. He, Y. Wang, L.-J. Sun, and H.-p. Huang, "Quality-aware incentive mechanism for mobile crowd sensing," *Journal of Sensors*, vol. 2017, 2017.

[189] W. Jin, M. Li, L. Guoy, and L. Yang, "Dpda: A differentially private double auction scheme for mobile crowd sensing," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[190] S. Chen, M. Liu, and X. Chen, "A truthful double auction for two-sided heterogeneous mobile crowdsensing markets," *Computer Communications*, vol. 81, pp. 31–42, 2016.

[191] R. Rana, C. T. Chou, N. Bulusu, S. Kanhere, and W. Hu, "Ear-phone: A context-aware noise mapping using smart phones," *Pervasive and Mobile Computing*, vol. 17, pp. 1–22, 2015.

[192] E. Bales, N. Nikzad, N. Quick, C. Ziftci, K. Patrick, and W. G. Griswold, "Personal pollution monitoring: mobile real-time air quality in daily life," *Personal and Ubiquitous Computing*, vol. 23, no. 2, pp. 309–328, 2019.

[193] M. R. Dey, U. Satapathy, P. Bhanse, B. K. Mohanta, and D. Jena, "Magtrack: detecting road surface condition using smartphone sensors and machine learning," in *IEEE TENCON*. IEEE, 2019, pp. 2485–2489.

[194] C.-W. Yi, Y.-T. Chuang, and C.-S. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1905–1917, 2015.

[195] A. Allouch, A. Koubâa, T. Abbes, and A. Ammar, "Roadsense: Smartphone application to estimate road conditions using accelerometer and gyroscope," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4231–4238, 2017.

[196] B. Bose, J. Dutta, S. Ghosh, P. Pramanick, and S. Roy, "D&rsense: Detection of driving patterns and road anomalies," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*. IEEE, 2018, pp. 1–7.

[197] D. Vij and N. Aggarwal, "Smartphone based traffic state detection using acoustic analysis and crowdsourcing," *Applied Acoustics*, vol. 138, pp. 80–91, 2018.

[198] M. Y. Alam, S. Imam, H. Anurag, S. Saha, S. Nandi, and M. Saha, "Lisense: Monitoring city street lighting during night using smartphone sensors," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 596–603.

[199] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 127–135.

[200] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1402–1410.

[201] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: Privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2020.

[202] Y. Zhang and X. Zhang, "Incentive mechanism with task bundling for mobile crowd sensing," *ACM Transactions on Sensor Networks*, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3581788

[203] J. Zhang, X. Li, Z. Shi, and C. Zhu, "A reputation-based and privacy-preserving incentive scheme for mobile crowd sensing: a deep reinforcement learning approach," *Wireless Networks*, Sep. 2022. [Online]. Available: https://doi.org/10.1007/s11276-022-03111-9

[204] D. Li, S. Wang, J. Liu, H. Liu, and S. Wen, "Crowdsensing from the perspective of behavioral economics: An incentive mechanism based on mental accounting," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9123–9139, Oct. 2019. [Online]. Available: https://doi.org/10.1109/jiot.2019.2928035

[205] L. Wang, Z. Yu, D. Zhang, B. Guo, and C. H. Liu, "Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation," *IEEE Transactions on Mobile Computing*, vol. 18, no. 1, pp. 84–97, Jan. 2019. [Online]. Available: https://doi.org/10.1109/tmc.2018.2827375

[206] Y. Zhang and X. Zhang, "Price learning-based incentive mechanism for mobile crowd sensing," *ACM Transactions on Sensor Networks*, vol. 17, no. 2, pp. 1–24, May 2021. [Online]. Available: https://doi.org/10.1145/3447622

[207] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 307–319, Feb. 2018. [Online]. Available: https://doi.org/10.1109/tmc.2017.2714668

[208] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2316–2329, Oct. 2020. [Online]. Available: https://doi.org/10.1109/tmc.2019.2927314

[209] Y. Zhan, Y. Xia, J. Zhang, T. Li, and Y. Wang, "An incentive mechanism design for mobile crowdsensing with demand uncertainties," *Information Sciences*, vol. 528, pp. 1–16, Aug. 2020. [Online]. Available: https://doi.org/10.1016/j.ins.2020.03.109

[210] L. Y. Chu and Z.-J. M. Shen, "Truthful double auction mechanisms," *Operations research*, vol. 56, no. 1, pp. 102–120, 2008.

[211] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein, "Auctions and bidding: A guide for computer scientists," *ACM Computing Surveys (CSUR)*, vol. 43, no. 2, pp. 1–59, 2011.

[212] S. Liu, Y. Yu, L. Guo, P. L. Yeoh, Q. Ni, B. Vucetic, and Y. Li, "Truthful online double auctions for mobile crowdsourcing: An on-demand service strategy," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 096–16 112, 2022.

[213] "Amazon mechanical turk (mturk)," https://www.mturk.com/, accessed: 2020-05-06.

[214] M. Śliwińska-Kowalska and K. Zaborowski, "Who environmental noise guidelines for the european region: a systematic review on environmental noise and permanent hearing loss and tinnitus," *International journal of environmental research and public health*, vol. 14, no. 10, p. 1139, 2017.

[215] D. Banerjee *et al.*, "Research on road traffic noise and human health in india: Review of literature from 1991 to current," *Noise and Health*, vol. 14, no. 58, p. 113, 2012.

[216] C. Clark and K. Paunovic, "Who environmental noise guidelines for the european region: A systematic review on environmental noise and quality of life, wellbeing and mental health," *International journal of environmental research and public health*, vol. 15, no. 11, p. 2400, 2018.

[217] B. Hoffmann, "Noise and hypertension—a narrative review," *Current Epidemiology Reports*, vol. 5, no. 2, pp. 70–78, 2018.

[218] M. Basner and S. McGuire, "Who environmental noise guidelines for the european region: a systematic review on environmental noise and effects on sleep," *International journal of environmental research and public health*, vol. 15, no. 3, p. 519, 2018.

[219] E. Van Kempen, M. Casas, G. Pershagen, and M. Foraster, "Who environmental noise guidelines for the european region: a systematic review on environmental noise and cardiovascular and metabolic effects: a summary," *International journal of environmental research and public health*, vol. 15, no. 2, p. 379, 2018.

[220] WHO, "Night noise guidelines for europe," http://www.euro.who.int/__data/assets/pdf_file/0017/43316/E92845.pdf?ua=1, 2009, accessed: 2019-01-21.

[221] WHO, "Burden of disease from environmental noise - quantification of healthy life years lost in europe," https://www.who.int/quantifying_ehimpacts/publications/e94888.pdf?ua=1, 2019, accessed: 2019-01-21.

[222] WHO, "Environmental noise guidelines for the european region," 2018, accessed: 2019-01-19. [Online]. Available: http://www.euro.who.int/__data/assets/pdf_file/0008/383921/noise-guidelines-eng.pdf?ua=1

[223] R. C. Lewis, R. R. Gershon, and R. L. Neitzel, "Estimation of permanent noise-induced hearing loss in an urban setting," *Environmental science & technology*, vol. 47, no. 12, pp. 6393–6399, 2013.

[224] E. Directive, "Directive 2002/49/ec of the european parliament and the council of 25 june 2002 relating to the assessment and management of environmental noise," *Official Journal of the European Communities, L*, vol. 189, no. 18.07, p. 2002, 2002.

[225] E. Salomons, D. Van Maercke, J. Defrance, and F. de Roo, "The harmonoise sound propagation model," *Acta acustica united with acustica*, vol. 97, no. 1, pp. 62–74, 2011.

[226] S. Kephalopoulos, M. Paviotti, and F. Anfosso-Lédée, "Common noise assessment methods in europe (cnossos-eu)," 2012.

[227] K.-T. Tsai, M.-D. Lin, and Y.-H. Chen, "Noise mapping in urban environments: A taiwan study," *Applied Acoustics*, vol. 70, no. 7, pp. 964–972, 2009.

[228] W. Wei, T. Van Renterghem, B. De Coensel, and D. Botteldooren, "Dynamic noise mapping: A map-based interpolation between noise measurements with high temporal resolution," *Applied Acoustics*, vol. 101, pp. 127–140, 2016.

[229] M. Becker, S. Caminiti, D. Fiorella, L. Francis, P. Gravino, M. M. Haklay, A. Hotho, V. Loreto, J. Mueller, F. Ricchiuti *et al.*, "Awareness and learning in participatory noise sensing," *PloS one*, vol. 8, no. 12, p. e81638, 2013.

[230] U. Bhatt, S. Mani, E. Xi, and J. Z. Kolter, "Intelligent pothole detection and road condition assessment," *arXiv preprint arXiv:1710.02595*, 2017.

[231] N. Hassan, I. Siddiqui, S. Mazhar, and H. Hameed, "Road anomaly classification for low-cost road maintenance and route quality maps," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 645–650.

[232] W. Zamora, C. T. Calafate, J.-C. Cano, and P. Manzoni, "Accurate ambient noise assessment using smartphones," *Sensors*, vol. 17, no. 4, p. 917, 2017.

[233] G. Marques and R. Pitarma, "Noise mapping through mobile crowdsourcing for enhanced living environments," in *International Conference on Computational Science*. Springer, 2019, pp. 670–679.

[234] M. A. Oliver and R. Webster, "Kriging: a method of interpolation for geographical information systems," *International Journal of Geographical Information System*, vol. 4, no. 3, pp. 313–332, 1990.

[235] J. Li and A. D. Heap, "A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors," *Ecological Informatics*, vol. 6, no. 3-4, pp. 228–241, 2011.

[236] Li, Jin and Heap, Andrew D, *A review of spatial interpolation methods for environmental scientists*. Geoscience Australia Canberra, 01 2008. [Online]. Available: http://www.ga.gov.au/servlet/BigObjFileManager?bigobjid=GA12526

[237] H. Wackernagel, "Multivariate geostatistics: an introduction with applications," in *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, vol. 33, no. 8. Springer, 1996, pp. 363A–363A.

[238] P. M. Atkinson and C. D. Lloyd, "Geostatistical models and spatial interpolation," *Handbook of Regional Science*, pp. 1461–1476, 2014.

[239] P. Goovaerts, "Geostatistics for natural resources evaluation. oxford univ. press, new york." *Geostatistics for natural resources evaluation. Oxford Univ. Press, New York.*, 1997.

[240] J. W. Hager, L. L. Fry, S. S. Jacks, and D. R. Hill, "Datums, ellipsoids, grids, and grid reference systems," DEFENSE MAPPING AGENCY HYDROGRAPHIC/TOPOGRAPHIC CENTER WASHINGTON DC, Tech. Rep., 1992.

[241] S. Global, "Acoustics-description and measurement of environmental noise," https://shop.standards.ie/store/PreviewDoc.aspx?saleItemID=211887, 2020, accessed: 2020-04-04.

[242] J. W. Hager, J. F. Behensky, and B. W. Drew, "The universal grids: Universal transverse mercator (utm) and universal polar stereographic (ups). edition 1," DEFENSE MAPPING AGENCY HYDROGRAPHIC/TOPOGRAPHIC CENTER WASHINGTON DC, Tech. Rep., 1989.

[243] P. Reed, B. Minsker, and A. J. Valocchi, "Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation," *Water Resources Research*, vol. 36, no. 12, pp. 3731–3741, 2000.

[244] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation." *Encyclopedia of database systems*, vol. 5, pp. 532–538, 2009.

[245] Amprobe, "Amprobe sm-10 sound level meter," http://www.amprobe.com/amprobe/, 2018, accessed: 2019-02-24.

[246] M. Technologies, "Mextech sl-4012 sound level meter," https://www.mextech.co.in/sound-level-meter.html, 2018, accessed: 2019-02-24.

[247] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in canada using LSTM networks," *Chaos, Solitons & Fractals*, vol. 135, p. 109864, Jun. 2020. [Online]. Available: https://doi.org/10.1016/j.chaos.2020.109864

[248] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-series data: A comparative study," *Chaos, Solitons & Fractals*, vol. 140, p. 110121, Nov. 2020. [Online]. Available: https://doi.org/10.1016/j.chaos.2020.110121

[249] A. B. Said, A. Erradi, H. A. Aly, and A. Mohamed, "Predicting COVID-19 cases using bidirectional LSTM on multivariate time series," *Environmental Science and Pollution Research*, vol. 28, no. 40, pp. 56 043–56 052, May 2021. [Online]. Available: https://doi.org/10.1007/s11356-021-14286-7

[250] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of covid-19 using deep learning models: India-USA comparative case study," *Chaos, Solitons & Fractals*, vol. 140, p. 110227, Nov. 2020. [Online]. Available: https://doi.org/10.1016/j.chaos.2020.110227

[251] A. I. Middya and S. Roy, "Pollutant specific optimal deep learning and statistical model building for air quality forecasting," *Environmental Pollution*, vol. 301, p. 118972, May 2022. [Online]. Available: https://doi.org/10.1016/j.envpol.2022.118972

[252] ——, "Geographically varying relationships of COVID-19 mortality with different factors in india," *Scientific Reports*, vol. 11, no. 1, Apr. 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-86987-5

[253] S. Sannigrahi, F. Pilla, B. Basu, A. S. Basu, and A. Molter, "Examining the association between socio-demographic composition and COVID-19 fatalities in the european region using spatial regression approach," *Sustainable Cities and Society*, vol. 62, p. 102418, Nov. 2020. [Online]. Available: https://doi.org/10.1016/j.scs.2020.102418

[254] C. C. John, V. Ponnusamy, S. K. Chandrasekaran, and N. R, "A survey on mathematical, machine learning and deep learning models for COVID-19 transmission and diagnosis," *IEEE Reviews in Biomedical Engineering*, vol. 15, pp. 325–340, 2022. [Online]. Available: https://doi.org/10.1109/rbme.2021.3069213

[255] I. Kavouras, M. Kaselimi, E. Protopapadakis, N. Bakalos, N. Doulamis, and A. Doulamis, "COVID-19 spatio-temporal evolution using deep learning at a european level," *Sensors*, vol. 22, no. 10, p. 3658, May 2022. [Online]. Available: https://doi.org/10.3390/s22103658

[256] B. Nikparvar, M. M. Rahman, F. Hatami, and J.-C. Thill, "Spatio-temporal prediction of the COVID-19 pandemic in US counties: modeling with a deep LSTM neural network," *Scientific Reports*, vol. 11, no. 1, Nov. 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-01119-3

[257] C. C. da Silva, C. L. de Lima, A. C. G. da Silva, E. L. Silva, G. S. Marques, L. J. B. de Araújo, L. A. A. Júnior, S. B. J. de Souza, M. A. de Santana, J. C. Gomes *et al.*, "Covid-19 dynamic monitoring and real-time spatio-temporal forecasting," *Frontiers in Public Health*, vol. 9, 2021.

[258] K. R. Bhimala, G. K. Patra, R. Mopuri, and S. R. Mutheneni, "Prediction of COVID-19 cases using the weather integrated deep learning approach for india," *Transboundary and Emerging Diseases*, vol. 69, no. 3, pp. 1349–1363, Apr. 2021. [Online]. Available: https://doi.org/10.1111/tbed.14102

[259] B. Tang, N. L. Bragazzi, Q. Li, S. Tang, Y. Xiao, and J. Wu, "An updated estimation of the risk of transmission of the novel coronavirus (2019-ncov)," *Infectious disease modelling*, vol. 5, pp. 248–255, 2020.

[260] S. Zhao, Q. Lin, J. Ran, S. S. Musa, G. Yang, W. Wang, Y. Lou, D. Gao, L. Yang, D. He *et al.*, "Preliminary estimation of the basic reproduction number of novel coronavirus (2019-ncov) in china, from 2019 to 2020: A data-driven analysis in the early phase of the outbreak," *International journal of infectious diseases*, vol. 92, pp. 214–217, 2020.

[261] D. Fanelli and F. Piazza, "Analysis and forecast of covid-19 spreading in china, italy and france," *Chaos, Solitons & Fractals*, vol. 134, p. 109761, 2020.

[262] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. IEEE, Aug. 2017. [Online]. Available: https://doi.org/10.1109/icengtechnol.2017.8308186

[263] B. Zhao, , H. Lu, S. Chen, J. Liu, D. Wu, , , and and, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, Feb. 2017. [Online]. Available: https://doi.org/10.21629/jsee.2017.01.18

[264] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, Jan. 2021. [Online]. Available: https://doi.org/10.1016/j.ijforecast.2020.06.008

[265] J. L. Elman, "Finding structure in time," vol. 14, no. 2, pp. 179–211, Mar. 1990.

[266] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006. [Online]. Available: https://doi.org/10.1126/science.1127647

[267] A. I. Middya and S. Roy, "Geographically varying relationships of COVID-19 mortality with different factors in india," *Scientific Reports*, vol. 11, no. 1, Apr. 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-86987-5

[268] WHO Coronavirus Disease (COVID-19) Dashboard, (2020), [Online], Available: https://covid19.who.int/.

[269] F. Zhou, T. Yu, R. Du, G. Fan, Y. Liu, Z. Liu, J. Xiang, Y. Wang, B. Song, X. Gu *et al.*, "Clinical course and risk factors for mortality of adult inpatients with covid-19 in wuhan, china: a retrospective cohort study," *The lancet*, 2020.

[270] S. Garg, "Hospitalization rates and characteristics of patients hospitalized with laboratory-confirmed coronavirus disease 2019—covid-net, 14 states, march 1–30, 2020," *MMWR. Morbidity and mortality weekly report*, vol. 69, 2020.

[271] K. Li, J. Wu, F. Wu, D. Guo, L. Chen, Z. Fang, and C. Li, "The clinical and chest ct features associated with severe and critical covid-19 pneumonia," *Investigative radiology*, 2020.

[272] A. Ehlert, "The socioeconomic determinants of covid-19: A spatial analysis of german county level data," *medRxiv*, 2020.

[273] S. Sannigrahi, F. Pilla, B. Basu, A. S. Basu, and A. Molter, "Examining the association between socio-demographic composition and covid-19 fatalities in the european region using spatial regression approach," *Sustainable cities and society*, vol. 62, p. 102418, 2020.

[274] A. Gupta, S. Banerjee, and S. Das, "Significance of geographical factors to the covid-19 outbreak in india," *Modeling earth systems and environment*, pp. 1–9, 2020.

[275] F. Sun, S. A. Matthews, T.-C. Yang, and M.-H. Hu, "A spatial analysis of the covid-19 period prevalence in us counties through june 28, 2020: where geography matters?" *Annals of epidemiology*, 2020.

[276] C. Brunsdon, S. Fotheringham, and M. Charlton, "Geographically weighted regression: A method for exploring spatial nonstationarity," *Encyclopedia of Geographic Information Science*, p. 558, 2008.

[277] N. Cressie, *Statistics for spatial data*. John Wiley & Sons, 2015.

[278] L. Anselin, "Local indicators of spatial association—lisa," *Geographical analysis*, vol. 27, no. 2, pp. 93–115, 1995.

[279] M. N. Kamel Boulos and E. M. Geraghty, "Geographical tracking and mapping of coronavirus disease covid-19/severe acute respiratory syndrome coronavirus 2 (sars-cov-2) epidemic and associated events around the world: how 21st century gis technologies are supporting the global fight against outbreaks and epidemics," pp. 1–12, 2020.

[280] S. X. Zhang, H. Huang, and F. Wei, "Geographical distance to the epicenter of covid-19 predicts the burnout of the working population: Ripple effect or typhoon eye effect?" *Psychiatry Research*, vol. 288, p. 112998, 2020.

[281] D. Patoulias, A. Katsimardou, K. Stavropoulos, K. Imprialos, M.-S. Kalogirou, and M. Doumas, "Renin-angiotensin system inhibitors and covid-19: a systematic review and meta-analysis. evidence for significant geographical disparities," *Current hypertension reports*, vol. 22, pp. 1–13, 2020.

[282] Y. Zhang, Y. Li, B. Yang, X. Zheng, and M. Chen, "Risk assessment of covid-19 based on multisource data from a geographical viewpoint," *IEEE Access*, vol. 8, pp. 125 702–125 713, 2020.

[283] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet infectious diseases*, vol. 20, no. 5, pp. 533–534, 2020.

[284] Z. Arab-Mazar, R. Sah, A. A. Rabaan, K. Dhama, and A. J. Rodriguez-Morales, "Mapping the incidence of the covid-19 hotspot in iran–implications for travellers," *Travel medicine and infectious disease*, vol. 34, p. 101630, 2020.

[285] D. N. Prata, W. Rodrigues, and P. H. Bermejo, "Temperature significantly changes covid-19 transmission in (sub) tropical cities of brazil," *Science of the Total Environment*, vol. 729, p. 138862, 2020.

[286] L. Orea and I. C. Álvarez, "How effective has the spanish lockdown been to battle covid-19? a spatial analysis of the coronavirus propagation across provinces," *Documento de trabajo*, vol. 3, no. 2020, pp. 1–27, 2020.

[287] D. Giuliani, M. M. Dickson, G. Espa, and F. Santi, "Modelling and predicting the spatio-temporal spread of covid-19 in italy," *BMC infectious diseases*, vol. 20, no. 1, pp. 1–10, 2020.

[288] M. Rezaei, A. A. Nouri, G. S. Park, and D. H. Kim, "Application of geographic information system in monitoring and detecting the covid-19 outbreak," *Iranian Journal of Public Health*, vol. 49, no. Suppl 1, p. 114, 2020.

[289] V. Stojkoski, Z. Utkovski, P. Jolakoski, D. Tevdovski, and L. Kocarev, "Correlates of the country differences in the infection and mortality rates during the first wave of the covid-19 pandemic: evidence from bayesian model averaging," *Scientific Reports*, vol. 12, no. 1, p. 7099, 2022.

[290] "Central pollution control board," https://cpcb.nic.in/, accessed: 2020-07-08.

[291] S. Menard, *Applied logistic regression analysis*. Sage, 2002, vol. 106.

[292] B. L. Golden and E. A. Wasil, "Optimisation [by dm greig (london: Longman, 1980, 179 pp.)]," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, no. 5, pp. 684–684, 1982.

[293] W. Yang, "An extension of geographically weighted regression with flexible bandwidths," Ph.D. dissertation, University of St Andrews, 2014.

[294] T. M. Oshan, Z. Li, W. Kang, L. J. Wolf, and A. S. Fotheringham, "mgwr: A python implementation of multiscale geographically weighted regression for investigating process spatial heterogeneity and scale," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 269, 2019.

[295] X. Wu, R. C. Nethery, B. M. Sabath, D. Braun, and F. Dominici, "Exposure to air pollution and covid-19 mortality in the united states," *medRxiv*, 2020.

[296] C. Magazzino, M. Mele, and N. Schneider, "The relationship between air pollution and covid-19-related deaths: an application to three french cities," *Applied Energy*, p. 115835, 2020.

[297] R. D. Brook, S. Rajagopalan, C. A. Pope III, J. R. Brook, A. Bhatnagar, A. V. Diez-Roux, F. Holguin, Y. Hong, R. V. Luepker, M. A. Mittleman *et al.*, "Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the american heart association," *Circulation*, vol. 121, no. 21, pp. 2331–2378, 2010.

[298] A. Sanyaolu, C. Okorie, A. Marinkovic, R. Patidar, K. Younis, P. Desai, Z. Hosein, I. Padda, J. Mangat, and M. Altaf, "Comorbidity and its impact on patients with covid-19," *SN comprehensive clinical medicine*, pp. 1–8, 2020.

[299] D. Su, Y. Chen, K. He, T. Zhang, M. Tan, Y. Zhang, and X. Zhang, "Influence of socio-ecological factors on covid-19 risk: a cross-sectional study based on 178 countries/regions worldwide," *Regions Worldwide (4/17/2020)*, 2020.

[300] P. Chatterjee, "Is india missing covid-19 deaths?" *The Lancet*, vol. 396, no. 10252, p. 657, 2020.

Spatial Interpolation Techniques on Participatory Sensing Data

ASIF IQBAL MIDDYA and SARBANI ROY, Jadavpur University, India

Spatial distributions of data of natural phenomena can be estimated by using different spatial interpolation techniques. These techniques can be used for the purpose of developing urban noise pollution monitoring applications, so they can truly describe the actual urban noise pollution scenario of any region of interest to make effective and informed decisions. In this context, our aim is to use IoT-cloud based framework to generate dynamic (i.e., changes in terms of time and space) noise maps as a service with the help of spatial interpolation techniques. Noise map generation is an effective method for visualizing and assessing urban noise pollution. In this article, we have proposed three spatial interpolation techniques (GLIDW, I-GLIDW, GLIDW-OK) that work on participatory sensing-based noise pollution data collected using smartphones as IoT devices to generate dynamic noise maps. Proposed techniques can address diverse scenarios such as *sparse datasets*, *high accuracy*, *better response time*, and so on. Depending on the situation, we can choose an appropriate technique. We evaluate our proposed methods based on a real-world urban noise pollution dataset collected by participants over a period of two years in an urban area of the city Kolkata. The results are compared with inverse distance weighting (IDW) and Ordinary Kriging (OK) methods. The method GLIDW is proposed for a dense dataset. The results validate that in the case of a dense dataset GLIDW dominates over other methods. But, when the data sparsity level is medium, I-GLIDW performs well. However, if the dataset is very sparse, then GLIDW-OK dominates in terms of predictive accuracy. The results also show that Relative Improvement (RI) of I-GLIDW and GLIDW-OK is always positive compared to baseline methods IDW and OK.

CCS Concepts: • **Information systems → Spatial-temporal systems**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; **Smartphones**;

Additional Key Words and Phrases: IoT, smartphone, cloud, spatial interpolation, noise pollution monitoring, participatory sensing

ACM Reference format:
Asif Iqbal Middya and Sarbani Roy. 2021. Spatial Interpolation Techniques on Participatory Sensing Data. *ACM Trans. Spatial Algorithms Syst.* 7, 3, Article 13 (June 2021), 32 pages.
https://doi.org/10.1145/3457609

This research work is partially supported by the project entitled "Participatory and Realtime Pollution Monitoring System For Smart City," funded by Department of Science & Technology, Government of West Bengal, India.
Authors' addresses: A. I. Middya and S. Roy, Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mallik Road, Jadavpur, Kolkata - 700032, West Bengal, India. emails: {asifim.rs, sarbani.roy}@jadavpuruniversity.in.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
2374-0353/2021/06-ART13 $15.00
https://doi.org/10.1145/3457609

1 INTRODUCTION

The rapid growth of urbanization modernizes people's lives, but it also aggravates the problem of noise pollution in urban areas. Noise pollution today has become a major problem that may cause serious health threats such as hearing loss [1], increasing stress levels [2], behavioral and mental disorders [3], hypertension [4], sleep disorders [5], heart disease [6], and many more [7, 8]. It also affects the behavior and habitat of animals. According to a recent report released by **WHO (World Health Organization)**, there is a piece of strong evidence that noise is one of the top environmental hazards to both physical and mental health and well-being [9]. Another report from **EEA (European Environment Agency)** has revealed that environmental noise results in more than 16,600 premature deaths in Europe every year, with approximately 13M adults suffering from sleep disturbance and a further 32M annoyed by it. Almost 15% population (26M people) of the U.S.A. may have suffered from NIPTS (noise-induced permanent threshold shift), which is a permanent reduction in hearing acuity due to excessive exposure to workplace or leisure noise [10]. Now, according to the directive 2002/49/EC [11], city noise maps are the key resource for analyzing and decision-making in the development of action plans to reduce urban noise pollution levels. Here, a city noise map is a geospatial representation of noise pollution level distribution in a city for a particular time interval. Now, in response to the above-mentioned health issues and societal challenges related to environmental noise, the main focus of this work is to come up with an end-to-end system that can provide a high granular and spatially continuous noise map as a service to the users. This high granular and spatially continuous noise map will make it easier to monitor environmental noise in urban areas. It can also enhance citizens' awareness of environmental noise pollution and help in developing mitigation plans to tackle the adverse effect of environmental noise pollution.

1.1 Motivation

Noise pollution monitoring is generally not an easy task to completely and accurately determine the urban noise pollution in a large region. Traditionally, noise assessment in any region is usually performed using a few noise monitoring stations that are sparsely spread over different locations (e.g., near streets, railway stations, hospitals, airports, residential and commercial areas) of a city. In noise pollution monitoring, the noise mapping can be cast as a modelling task (e.g., emission and propagation models [12, 13]). The sparse monitoring stations' data are mainly used to validate the mapping results. Also, there is little existing literature that involves noise mapping techniques for interpolating between noise measurements in urban noise measurement networks [14, 15]. Although the traditional method of noise pollution data collection provides more accurate measurements, it suffers from some limitations. First, using sparse spatio-temporal data, traditional noise mapping models do not help us to create a spatially continuous noise map of high spatial and temporal resolution [16] for a large area like a city. Second, it is hard to monitor and assess unusual local or short-term noise pollution. Third, the traditional approach for noise pollution monitoring is costly due to the need for human expertise, expensive equipment, periodic maintenance of equipment, and effort required to monitor noise pollution.

To overcome the above-mentioned limitations, we can use a new paradigm of sensing that is called *participatory sensing*. In participatory sensing, people and communities use mobile devices (e.g., smartphones, tablets) to collect and cloud services to store and analyze systematic data to explore any phenomena of interest. Taking inspiration from the concept of participatory sensing, we use GPS-enabled smartphones as IoT devices and involve the citizens to carry them to measure, locate, and provide contextual inputs for monitoring urban noise pollution. Eventually, this IoT-based approach helps us to overcome the problem of data sparsity. However, to monitor and

assess unusual local or short-term noise pollution, we propose spatial interpolation techniques for generating an effective noise map. An effective noise map can help citizens not only to assess their individual exposure to ambient noise in their daily environment but also to keep track of noise pollution levels in their surroundings and eventually it helps to improve awareness. Additionally, in the current world scenario smartphones are becoming an essential part of human life, which is cheaper than standard noise-measuring devices. Hence, IoT devices can be used to reduce the overall cost of noise pollution monitoring.

1.2 Related Works

The existing works related to participatory sensing-based monitoring of urban dynamics including urban noise pollution mapping are presented in this section.

Participatory sensing using IoT devices (e.g., smartphones) has recently been used as a sensing method to explore a city's various dynamics (e.g., pollution, human mobility, road condition) at high spatiotemporal resolution [17–20]. The IoT-based participatory sensing generates sensor data stream [21, 22] that could be analyzed in many applications that are related to those dynamics. There are numerous attempts to make citizen-centered improvements through the utilization of such sensing data [23]. This type of sensing generates a huge amount of data that covers a wide variety of information, including human mobility, city pollution, road conditions, and so on. Many works investigate data acquired from participatory sensors such as from smartphone-equipped sensors [24–29], vehicles GPS [30], car-mounted cameras [31], on-body sensors [32], and so on. For instance, References [24, 28] explore different phenomena related to human mobility dynamics using citizens' location data from smartphones. In Reference [29], the authors focus on exploring road condition dynamics using smartphone sensor data. They proposed an efficient way of estimating the transverse slope profile of a road segment using GPS, gyroscope, and accelerometer sensor data of smartphones. In Reference [33], Cao et al. utilize high granular trajectory data (mobile cellular accessing data) to explore daily commuting patterns on a citywide scale. In Reference [31], Tokuda et al. proposed an approach for generating a spatio-temporal pedestrian density map using street-level images of New York City. Moreover, smartphone-based human mobility and activity data are also used for transit map generation [34], quantifying people's behavior in public areas [35], health-optimal routing [36], and so on. Similarly, there are existing works [25, 26] that focus on exploring pollution dynamics of the city using smartphone-based participatory sensing data. A detailed comparison of the existing works related to participatory sensing-based monitoring of urban dynamics is provided in Table 1. This table compares the existing works in terms of the contribution, sensors used, context-awareness, calibration, map generation, map type, spatial interpolation, consideration of data sparsity, and end-to-end system behavior. It is interesting to see that none of the existing literature addresses the problem of data sparsity. Moreover, most of the existing works do not utilize spatial interpolation on participatory sensing data and only have **spatially non-continuous mapping (SNCM)**.

The widespread adoption of smartphones also enables citizens to take part in producing urban noise pollution data. These urban noise pollution data can be then collectively used to offer various services to the citizens. For instance, in Reference [16], Maisonneuve et al. present an approach called NoiseTube for the assessment of noise pollution using the data collected from users' smartphone sensors such as GPS, microphones, and so on. In other words, they use GPS-equipped mobile phones to measure citizens' exposure to noise in their daily life. They combine all the geotagged smartphone measurements and create a **PoI (Point of Interest)**-based map visualization of urban noise pollution data. However, in this work, no data cleaning technique is implemented to eliminate the measurements that are collected from the inappropriate sensing contexts (e.g., when the phone is in a bag/pocket). Eventually, the urban noise map produced in

Table 1. A Detailed Comparison of the Existing Works Related to Participatory Sensing-based Monitoring of Urban Dynamics

City Dynamics	Reference & year	Contribution	Sensors used	Context-aware data cleaning	Calibration	Map generation	Map type	Spatial interpolation	Consideration of data sparsity	End-to-End System
Human mobility	Yabe et al. [24], 2016	Predicts irregular movement of humans after the mid-level disasters.	GPS	×	×	×	×	×	×	×
	Gu et al. [28] 2017	Predict venue(s) visited by citizens from the GPS trajectory.	GPS	×	×	×	×	×	×	×
	Tokuda et al. [31], 2020	Generate spatiotemporal pedestrian density map using street-level images.	GPS, Camera	×	×	SNCM	STA	×	×	×
Road condition	Bhatt et al. [44], 2017	Develop a system for detection and mapping of potholes.	GPS, Acc, Gyro	×	×	SNCM	STA	×	×	✓
	Hassan et al. [45], 2019	Low-cost road maintenance and road condition mapping.	GPS, Acc	×	×	SNCM	STA	×	×	×
	Gupta et al. [29], 2020	Estimate the transverse slope profile of road segments.	GPS Acc, Gyro	×	×	×	×	×	×	×
Air pollution	Hasenfratz et al. [26], 2012	Develop a smartphone-based system for air pollution monitoring.	GPS, Ozone sensor	×	✓	SNCM	STA	×	×	×
	Dutta et al. [46], 2017	Propose a system for crowdsensing-based air pollution monitoring.	GPS, MQ-135, HC-05	×	×	SNCM	STA	×	×	✓
	Middya et al. [25], 2020	Propose a unified framework for urban sensing systems.	GPS, MQ-135, HC-05	×	×	SCM	STA	✓	×	✓
Noise pollution	Maisonneuve et al. [16], 2009	Assessment of noise pollution with mobile phones	GPS, microphone	×	✓	SNCM	STA	×	×	✓
	Becker et al. [37], 2013	The behavior of people engaged in noise sensing are analyzed.	GPS, microphone	×	×	SNCM	STA	×	×	×
	Rana et al. [38], 2015	Reconstruction of noise map from incomplete and random noise samples.	GPS, microphone, Acc., Ptx.	✓	✓	SNCM	STA	×	×	✓
	Zamora et al. [47], 2017	Evaluate the impact of various algorithms for noise computation.	GPS, microphone	×	✓	SNCM	STA	×	×	×
	Dutta et al. [39], 2017	Real-time monitoring of urban noise pollution using MCS.	GPS, microphone	×	✓	SNCM	STA	×	×	✓
	Marques et al. [48], 2019	Real-time monitoring of urban noise pollution using MCS.	GPS, microphone	×	×	SNCM	STA	×	×	✓

✓: The requirement has been met; ×: The requirement has not been met; Acc : Accelerometer sensor; Gyro: Gyroscope sensor; SCM: Spatially Continuous Mapping; SNCM: Spatially Non-Continuous Mapping; STA: Static Map.

NoiseTube may suffer from erroneous data and may not reflect the actual reality. Also, the noise map created here is not spatially continuous and therefore may not help efficient decision-making. Becker et al. [37] developed a smartphone application called WideNoise, which can be used to involve citizens in noise-measuring activities. The application WideNoise is developed to record both objective (noise samples) and subjective (opinions, feelings, etc.) data. Using the data provided by smartphone users, the analysis of awareness and learning is performed about current environmental issues. But the authors mainly focused on presenting a rigorous analysis of participatory sensing-based urban noise data, rather than high granular urban noise mapping. In Reference [38], Rana et al. show the use of different interpolation and regularization methods to reconstruct the environmental noise pollution map from incomplete and random noise samples collected using

Table 2. Notation

Symbols	Description
$d(x, x_i)$	Euclidean distance between points x_i and x
p	Power parameter of IDW
$\widehat{\gamma}(h)$	Estimated semivariance at a separation distance h
$N(h)$	Number of pairs of sample points separated by distance h
Z_i	Observed value at point i
L_{eq}	Equivalent continuous sound level
sm	Smartphone model name
si	Smartphone state information
$\xi^{sm_i}(L_{eq})$	Calibration offset of noise reading L_{eq} in smartphone model sm_i
$L_{eq}^{'}$	Noise level after adding calibration offset ξ to L_{eq}
acc_x	x-axis data of accelerometer reading
acc_y	y-axis data of accelerometer reading
acc_z	z-axis data of accelerometer reading
n	The dimension of each grid cell
S	Sample data
G	Set of all grid cells having size $n \times n$
g_i	ith grid cell
R	Radius that is considered for computing weight
m_i	Midpoint of grid cell g_i
$N_G^{(pred)}$	Predicted noise values for all grid cells in G
N_{Th}	Number of nearest neighbor required to calculate a grid cell's noise
NN_i	A set of sample points that are within radius R from the midpoint m_i
R'	Radius for iterative computation in I-GLIDW
G_{Th}	A predefined threshold for I-GLIDW
V^m	Set of variogram models

participatory data collection. However, they did not make any attempt to calculate spatially continuous noise maps by estimating urban noise levels at unsampled locations. In Reference [39], an end-to-end system called *NoiseSense* is proposed for urban noise monitoring and PoI-based noise mapping. This work mainly focused on the overall monitoring system. The authors did not use any interpolation technique to calculate noise levels at missing locations.

The noise pollution data that are collected using IoT-based approach can be used by spatial interpolation methods to generate spatially continuous noise pollution maps. More specifically, spatial interpolation is the process of using the values at observed locations to predict the values at other unknown locations in geographic space. The rationale behind spatial interpolation is the assumption that everything is related to everything else, but near things are more related than distant things in space (Tobler's Law of Geography [40]). In literature, there are several methods available for spatial interpolation [41]. Among them, the two popular methods are **Inverse Distance Weighting (IDW)** [42] and Kriging [43], which are frequently used in the field of environmental science.

1.3 Challenges for IoT-based Approaches

The above-mentioned IoT-based approaches will help us create better noise pollution maps if they overcome the following challenges:

*C*1: The IoT-based approach that uses citizens' smartphones to continuously collect real-time noise pollution levels usually generates huge data. Eventually, it is required to have a proper framework for storage and processing to generate spatially continuous dynamic noise maps.

*C*2: Low-cost IoT devices like smartphones generate low-quality data. In other words, smartphone measurements of ambient noise levels may not produce true values. To produce true noise levels, IoT device-specific proper calibration mechanisms are required.

*C*3: In participatory sensing, data may be generated from different sensing contexts (e.g., the phone is in hand, in pocket/bag or an active phone call is going on). The ambient noise measurements using this approach may not provide accurate values due to inappropriate sensing context (e.g., when the phone is in pocket/bag). This approach should have a proper context-aware data cleaning technique to remove the measurements collected from inappropriate sensing contexts.

*C*4: The IoT-based approach that uses participatory sensing can generate huge noise pollution data. However, since the process of collecting data depends completely on the mobility pattern of the participants, certain regions may experience a lack of data, leading to a spatially sparse dataset. A sparse dataset makes it difficult to estimate urban noise at high spatial granularity. Note that the spatial missing values in the participatory sensing-based data also prevent us to achieve high accuracy. Moreover, a huge noise pollution dataset can make predictive analysis computationally expensive and slow. This may significantly affect the response time if the predictive models are deployed in an end-to-end framework to provide noise maps as a service to the users. Eventually, it requires designing efficient techniques that can address scenarios such as *sparse datasets*, *high accuracy*, *better response time* to analyze, generate, and visualize a dynamic noise map.

1.4 Contribution

As discussed in Section 1.2, the state-of-the-art works [16, 37–39, 47, 48] for urban noise mapping and monitoring mainly focus on developing a framework for urban noise data collection and directly plot them to a map for visual analysis. However, these works lack in addressing the above-mentioned challenges (Section 1.3) that are directly related to the IoT-based approach for noise monitoring. Compared to the state-of-the-art techniques, our contributions mainly differ in the following aspects: (i) The proposed approach developed smartphone model specific calibration mechanism for urban noise measurements. (ii) The existing works often use all the participatory sensing data including the data that are collected from inappropriate sensing contexts. However, an urban noise map that is generated from such erroneous data may not reflect the actual reality. To address this problem, in this work, a context-aware data cleaning model is developed using the machine learning technique. (iii) The existing approaches on participatory sensing-based noise map generation come up with PoI-based noise maps that are spatially non-continuous in nature, i.e., urban noise pollution levels on different PoIs are visualized on a map. In contrast, spatial interpolation is utilized in this work to produce spatially continuous urban noise pollution maps that can be helpful to make more effective and informed decisions. (iv) No existing literature addresses the scenarios such as *sparse dataset*, *high accuracy*, *better response time* to analyze, generate, and visualize a dynamic noise map. With the aim to address such scenarios, in this article, we have proposed three spatial noise-mapping and inferencing methods, **GLIDW (Grid-based Local Inverse Distance Weighting), I-GLIDW (Iterative Grid-based Local Inverse Distance Weighting)**, and **GLIDW-OK (Grid-based Local Inverse Distance Weighting - Ordinary Kriging)**. All these methods work on participatory sensing noise data for effectively and easily monitoring environmental noise. We have proposed three methods instead of just one, because different methods are suitable in different scenarios (see Section 4.4.2). The first approach GLIDW predicts local grid

noise having a sufficient number of sample points. Though GLIDW suffers from some limitations (see Section 3.6), we can overcome those limitations in I-GLIDW and GLIDW-OK. I-GLIDW iteratively propagates already predicted grid cell's noise to its neighbors having an insufficient number of noise samples. However, GLIDW-OK is based on both GLIDW and **OK (Ordinary Kriging)** [41]. It uses the concept of spatial correlation among different grid cells to capture the variation in the **Region Of Interest (ROI)**. The main idea of this article is to generate urban noise pollution maps of any region based on the participatory sensing data in a low-cost and efficient way using our proposed methods. Our key contributions are summarized as follows:

- To address challenge $C1$, we propose a framework that enables the users to get noise maps as a service based on their query.
- Device-specific calibration mechanism is developed to address challenge $C2$.
- To address challenge $C3$, a classification algorithm is developed that detects the sensing contexts (whether the phone is in the pocket or in hand) so noise data collected during inappropriate sensing contexts can be removed.
- To address challenge $C4$, we propose three methods (GLIDW, I-GLIDW, GLIDW-OK) for dynamic noise map generation using spatial interpolation techniques based on participatory sensing data. The results show that our proposed methods (I-GLIDW, GLIDW-OK) perform more accurately compared to the baseline methods (IDW, OK) for participatory sensing noise data.

1.5 Organization

The rest of this article is organized as follows: Section 2 presents some existing spatial interpolation methods. In Section 3, we provide the proposed approach, which includes participatory sensing-based data collection framework, noise-level measurement, context-aware data cleaning, GPS to MGRS conversion, and details of the proposed methods (GLIDW, I-GLIDW, and GLIDW-OK). In Section 4, we provide experimental results that include a study area and data collection, calibration, parameter and model selection, performance metrics, and performance comparison with two baselines (IDW and OK). In Section 6, we conclude this work.

2 BACKGROUND OF SPATIAL INTERPOLATION METHODS

2.1 Inverse Distance Weighting Method

Inverse distance weighting (IDW) [42] method is one of the simplest and widely used spatial interpolation methods. It is based on an assumption that the value at any unmeasured location can be predicted as a weighted average of measured values surrounding the prediction location. It assigns greater weights to the points that are closer to the prediction location than the points that are far away. The general form to calculate the predicted value u at an unknown point x based on sample $u_i = u(x_i)$ for $i = 1, 2, \ldots, N$ using IDW is given as follows [42]:

$$u(x) = \begin{cases} \dfrac{\sum_{i=1}^{N} \dfrac{1}{d(x,x_i)^p} u_i}{\sum_{i=1}^{N} \dfrac{1}{d(x,x_i)^p}}, & \text{if } d(x,x_i) \neq 0 \text{ for all } i, \\ u_i, & \text{otherwise,} \end{cases} \tag{1}$$

where x denotes prediction point (interpolated point), x_i is a known point (interpolating point), $d(x, x_i)$ is the distance from the interpolating point x_i to the interpolated point x, N is the total number of known points used in interpolation, and p is the power parameter. Table 2 provides a list of key notations used throughout the paper.

2.2 Kriging

Kriging is a frequently applied geostatistical method for spatial interpolation. There is a large body of literature describing the kriging method [41, 43, 49]. Kriging uses the assumption that the distance between interpolating points reflects a spatial correlation and eventually uses the semivariance to measure that spatial correlation. This spatial correlation can help us to capture the variation in the surface. A semivariogram (a plot of semivariance as a function of distance or lag) then permits us to model the similarity of observation points as a function of distance or lag. In other words, a semivariogram allows measuring the degree of similarity among the observation points separated by a given distance. The semivariogram can be computed from sample points by using Equation (2) as given below [41].

$$\widehat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} \{z(x_i) - z(x_i + h)\}^2 \tag{2}$$

Here, $\widehat{\gamma}(h)$ is the estimated semivariance at a separation distance h, and $z(x_i)$ and $z(x_i + h)$ are known values at two points x_i and $x_i + h$ that are separated spatially by distance h. The term $N(h)$ represents the number of pairs of sample points that are separated spatially by the distance h. Now, the next step in kriging is to fit a model to the empirical semivariogram such that semivariances can be obtained at all possible separation distances (i.e., lags). There are popular models (such as Spherical, Exponential, Gaussian, Linear, etc.) that are given in kriging literature [50]. After the model is fitted to the empirical semivariogram, it can be used to compute the weights λ_i that minimize the variance in the predicted value. Now, the estimation can be made by the following formula [51]:

$$\hat{z}(x_0) - \mu = \sum_{i=1}^{k} \lambda_i [z(x_i) - \mu(x_0)], \tag{3}$$

where, μ is called stationary mean, which is computed as the average of the whole dataset [52], λ_i is kriging weight, k and $\mu(x_0)$ are the total numbers of observation points and the mean of the observation points within the search neighborhood. In kriging literature, there exist several variants of kriging [50], among them, some classical methods are **Ordinary Kriging (OK), Simple Kriging (SK)**, and **Universal Kriging (UK)**. In the case of ordinary kriging, Equation (3) is modified by replacing μ with a local mean $\mu(x_0)$, which is the mean of the observation points within the search neighborhood [53, 54]. However, in the case of simple kriging, Equation (3) is modified by replacing $\mu(x_0)$ with the stationary mean μ. In other words, stationary mean μ is assumed to be constant over the whole domain. UK is an extension of Ordinary Kriging, which assumes that the spatial variation in z values has a trend in addition to the spatial correlation between sample points.

In the context of noise map generation, we have proposed three spatial interpolation techniques that are inspired from IDW and OK to address diverse scenarios such as *sparse dataset*, *high accuracy*, *better response time*, and so on.

3 PROBLEM STATEMENT AND PROPOSED APPROACH

3.1 Problem Statement

Given a set of users $U = \{User_1, User_2, \ldots, User_n\}$ and a set of noise samples $S = \{s_1, s_2, \ldots, s_m\}$ collected from those users' smartphones, where any noise sample s_i should be in the form as given in Section 3.2.1. Suppose, $S_{pq} \subseteq S$ is a set of samples within a given time interval $T_{pq} = [t_p, t_q]$, where t_p and t_q are start and end of the time interval. Let us consider, G = $\{g_1, g_2, \ldots, g_k\}$ is a set of grid cells. Our objective is to come up with an efficient technique

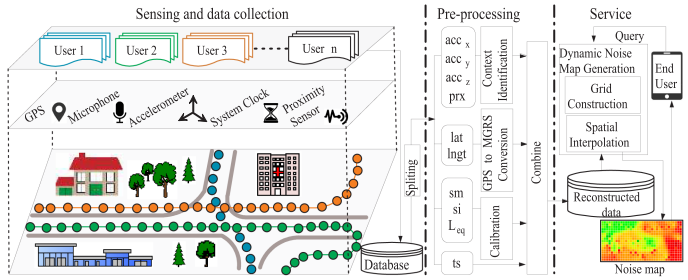


Fig. 1. Participatory sensing-based framework.

(or techniques) that utilizes the samples in S_{pq} to find a set of grid cells' urban noise values $N_{\mathrm{G}}^{(pred)} = \{N_1^{(pred)}, N_2^{(pred)}, \ldots, N_k^{(pred)}\}$ by addressing diverse scenarios such as sparse dataset, high accuracy, better response time, and so on, where $N_i^{(pred)}$ is the predicted noise value of grid cell $g_i \in \mathrm{G}$. Additionally, our aim is to create a noise map for the given interval T_{pq} using those predicted noise values and serve this to the users for visualizing and assessing noise pollution based on their query.

3.2 Participatory Sensing-based Framework

Figure 1 provides an overview of the framework of our participatory-based noise map generation technique, which consists of three parts: sensing and data collection, pre-processing, and service. Noise levels are measured on the smartphones and then transmitted to the server. The server combines spatio-temporal participatory sensing noise measurements collected from different participants. Note that combining ambient noise data from different users is required, because the noise readings are not available for all spatial and temporal resolution for a particular user.

3.2.1 Sensing and Data Collection. The sensing and data collection required participants to first install our application NoiseSense [39] for Android. As shown in the leftmost portion of Figure 1, using that Android application, we measure spatio-temporal noise data (see Section 3.3) and trajectories (sequence of GPS coordinates) for participants. The measured noise level and GPS location (latitude and longitude) are further tagged with some other sensor values such as 3-axis accelerometer and proximity sensor values for context-aware (such as whether the phone is in hand or in pocket/bag) data cleaning (see Section 3.4). Using the NoiseSense application, data samples get recorded at an interval of every five seconds in the format $<user_id, sm, si, lat, lngt, ts, L_{eq}, acc_x, acc_y, acc_z, prx>$ and are uploaded to a database. Here, $user_id$ is the unique user identifier; sm denotes smartphone model name; si is the state information (whether an active phone call in progress or idle) of the phone; lat and $lngt$ are latitude and longitude of the sampling location, respectively; ts is the time stamp value; L_{eq} is the recorded noise level; acc_x, acc_y, and acc_z are three axis accelerometer values; and prx is the proximity value. An important thing to note is that users can share all the above-mentioned sensors data anonymously to preserve their privacy.

3.2.2 Pre-processing. In the pre-processing stage, the dataset is first split into four parts. The first part contains 3-axis accelerometer and proximity sensor values for context classification (see Section 3.4) to identify noise sensing contexts i.e., whether the phone was in hand or pocket during data collection. This is necessary to decide whether the noise samples are accurate or not. The second part contains GPS coordinates (latitude and longitude). We convert the GPS coordinates to **MGRS (Military Grid Reference System)** grid index (see Section 3.5). The third part contains the smartphone model name (sm), recorded noise levels (L_{eq}), and state information (si). We discuss

in the later part (see Section 4.2) of this article that the calibration offset may vary from phone to phone and also under different sound levels. We only calculate calibration offset for those *sm* and L_{eq} pair for which $si = idle$. In other words, we are calculating calibration offset for those noise samples that are recorded while there was no phone call in progress. The fourth part contains only the timestamps, and we keep this unchanged. After all pre-processing, we reconstruct the original dataset in such a way that we keep only those noise samples with modified noise levels (i.e., $L_{eq} + offset$), MGRS grid index and timestamps for which context is *"in hand"* and $si = idle$.

3.2.3 Noise Map as a Service. Based on the reconstructed dataset, we can come up with noise map of the **Region Of Interest (ROI)** using our proposed spatial interpolation algorithms. Before creating a noise map, we divide the ROI into grid cells. Then, based on users' query (for example, a user may be interested to view the evening session's noise map of a particular region) any of our proposed algorithms (GLIDW, I-GLIDW, GLIDW-OK) can be used to produce a noise map. We discuss later in the results section that the predictive accuracy and response time of a method depend on the data sparsity level of a dataset. Depending on how sparse the dataset is, we choose an appropriate method for noise map generation (see Section 4.4.2). We use GLIDW if the dataset is dense. However, we prefer to use I-GLIDW and GLIDW-OK when the data sparsity levels of the dataset are medium and high, respectively.

3.3 Urban Noise-level Measurement

It is quite common to measure urban noise levels (or sound levels) as equivalent continuous sound level (L_{eq}). Usually, sound pressure levels fluctuate over time. L_{eq}, which is measured in **decibel (dB)**, is used to represent sound pressure levels that fluctuate over a time interval by taking into account the total sound energy over the time interval. In other words, L_{eq} would result in the same effects as the varying sound levels over a given time interval. The equivalent continuous sound level (L_{eq}) is calculated using the following equation [55]:

$$L_{eq} = 10 \log_{10} \left[\frac{1}{t_e - t_s} \int_{t_s}^{t_e} \left(\frac{p_t}{p_o} \right)^2 dt \right]. \tag{4}$$

Here, t_s and t_e are the start and end of the time interval, respectively, p_t is the acquired sound pressure of the sound signal, and p_o is reference sound pressure level (typically 20 μPa). This noise value (L_{eq}) may not be the true value; we need to add an offset ξ that is calculated by calibrating the smartphone against a standard device (e.g., Sound Level Meter). Hence, the modified noise value is determined as follows:

$$L^{'}_{eq} = L_{eq} + \xi. \tag{5}$$

Later, in Section 4.2, we explain how to calculate the value of ξ.

3.4 Context-aware Data Cleaning

Note that not all noise samples give accurate noise measurements. The noise sensing contexts (for example, whether the phone is in the user's hand or pocket) determine whether the noise samples are accurate. We discard those noise samples that are recorded in sensing contexts that provide inaccurate noise measurements. Some examples of such sensing contexts that provide inaccurate noise measurements are: (i) an active phone call in progress, (ii) phone is kept in pocket, and so on.

It may happen that the noise levels are recorded while there is an active phone call in progress. Hence, we need to correctly identify this scenario and eliminate the noise values that are recorded during that time. To do that, we also record the state information of the phone (idle or active call

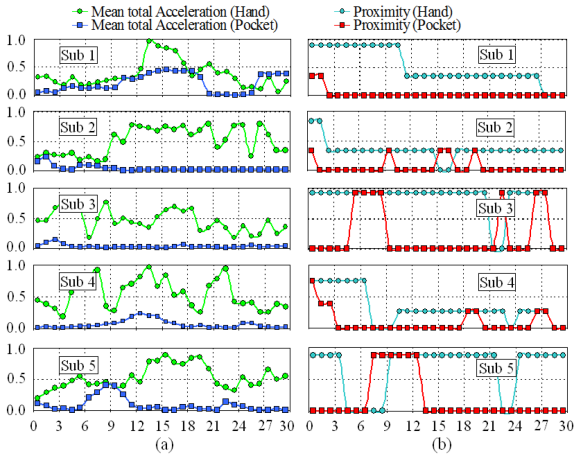


Fig. 2. Normalized mean total acceleration and proximity sensor behavior for hand and pocket contexts for five different subjects.

in progress) along with the ambient noise value. We only keep those data records for which the phone is in the idle state (i.e., no active phone call in progress) and remove others.

The accuracy of the ambient noise data that are collected using the smartphones also depends on the sensing contexts, like whether the smartphone is carried in hand or pocket. Usually, noise samples give accurate noise measurements when the smartphone is carried by hand. However, noise samples give inaccurate noise measurements when the smartphone is kept in the pocket. Hence, we need to correctly identify these two scenarios to eliminate the noise values that are recorded while the phone was in the pocket and keep those noise values that are recorded while the phone was in hand. We use a classification algorithm to identify these two contexts. We examine the feasibility of using the sensor data from accelerometer, gyroscope, light sensor, and the proximity sensor as features for classification. We appoint five subjects to record the readings using smartphone sensors. Each of the subjects is given two identical smartphones each keeping in a different sensing context (i.e., each subject is carrying one smartphone in hand and the other in pocket). Each subject is asked to collect 30 minutes of data. After analyzing the sensor data, we decide to exclude the gyroscope and light sensor, as their values are not significantly different in the two sensing contexts. We decide to use the accelerometer and the proximity sensor values for the purpose of classification. We represent normalized mean total acceleration and proximity sensor values for different subjects in different context in Figure 2. Here, the mean total acceleration is calculated using the following equation (Equation (6)). Here, a_{xi}, a_{yi}, and a_{zi} are x, y, and z axis data of the ith accelerometer reading, respectively.

$$\overline{m} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(a_{xi})^2 + (a_{yi})^2 + (a_{zi})^2} \tag{6}$$

We use Naive Bayes algorithm for context classification. We choose Naive Bayes, as it can not only predict with fair accuracy (81%) but also is easy to interpret and lightweight compared to other effective machine learning algorithms like support vector classification. Since it is not the main focus of our work, we do not pay much attention on improving the accuracy.

3.5 GPS to MGRS Convertion

For analyzing the participatory sensing-based noise data, the **Military Grid Reference System (MGRS)** [56] is used. MGRS provides a way to represent the surface of the Earth into squares of small grid cells. In this work, we convert the GPS coordinates of noise samples into MGRS grid cells of $1m^2$. The major motives for approximating GPS with the small square areas are as follows: First, inferring noise value for every possible GPS coordinate is impractical, because there are numerous such GPS coordinates. Second, the standards for noise pollution monitoring advised to measure the pollution levels in such square areas [57]. Here, we can simply use the MGRS grid reference system without any further modification. But, in the case of other reference systems such as **Universal Transverse Mercator (UTM), Universal Polar Stereographic (UPS)** [58], and so on, some modifications might be required. Now, one thing to be noted here is that the dimension of the grid cell for the proposed methods (namely, GLIDW, I-GLIDW, and GLIDW-OK) may not be the same as the MGRS grid dimension. For instance, using parameter selection (see Section 4.3) it is found that if prediction grid cell's dimension $n = 100$ meter, then the prediction accuracy is maximum. In other words, each of the prediction grid cells is of size 10^4 square meters.

3.6 GLIDW

The first approach we propose is an MGRS-based local spatial interpolation technique, called Grid-based Local Inverse Distance Weighting (GLIDW). GLIDW predicts grid noise locally using a method that is inspired from the concept of Inverse Distance Weighted (IDW) interpolation. Recall from Section 3.5 that we develop grid-based noise prediction, because predicting noise for every GPS coordinate is impractical.

Algorithm 1 describes the procedure of GLIDW. The inputs to this algorithm are sample noise data S, radius R, a threshold N_{Th}, the dimension of each grid cell n, and a time interval $T_{pq} = [t_p, t_q]$. Initially, we divide the study area into grid cells having dimension $n \times n$. We create a set G that contains all such grid cells. Next, a subset $S_{pq} \subseteq$ S of samples is created for which the sampling times are within the time interval $T_{pq} = [t_p, t_q]$, where t_p and t_q are the start and end of the time interval. From line 6–line 10, we calculate distances $d(m_i, p_j)$ of the sampling location p_j of sample $s_j \in S_{pq}$ from the midpoint m_i of grid cell g_i and come up with a set D_i^n of all those distances. In line 11, we construct a set NN_i of sample points that are within radius R from the midpoint m_i. Next, in line 12, we check whether $|NN_i|$ is greater than or equal to some predefined threshold N_{Th}. If $|NN_i| \geq N_{Th}$, then we consider only N_{Th} nearest neighbors for predicting grid cell's noise value. Later, in the results section (see Section 4.3), we will see the optimal value of N_{Th} such that we can calculate noise efficiently. From line 15–line 22, we calculate the weights of N_{Th} closest sample points. Next, we predict noise value of grid cell g_i in line 23. But, if the condition mentioned in line 12 is not satisfied, then it indicates that there are an insufficient number of sample points to predict the grid cell's noise value correctly, and eventually we put NULL as the noise value in line 25. At the end, the algorithm returns G along with the predicted noise values $N_{\mathrm{G}}^{(pred)}$.

Algorithm GLIDW runs in time $O(|\mathrm{G}|n_s log(n_s))$, where n_s and $|\mathrm{G}|$ are the total number of sample points and the total number of grid cells, respectively. The outer for loop (line 3–line 28) iterates over $O(|\mathrm{G}|)$ times. For each iteration of the outer loop, the for loop of line 6–line 10 iterates over $O(n_s)$ times. Moreover, for each iteration of the outer loop, if the condition at line 12 is satisfied, then NN_i can contain n_s sample points in the worst case, and eventually sorting at line 13 can be performed in $O(n_s log(n_s))$ time in this case. The next for loop (line 15–line 22) iterates over $O(N_{Th})$ time, where, N_{Th} is a user-defined threshold (in our case, we set $N_{Th} = 10$). So, overall time complexity is $O(|\mathrm{G}|(n_s + N_{Th}n_s log(n_s))) \approx O(|\mathrm{G}|n_s log(n_s))$. However, the space complexity of GLIDW is $O(|\mathrm{G}| + n_s)$. Set G contains all the grid cells that require $O(|\mathrm{G}|)$ spaces. To store the distances

ALGORITHM 1: GLIDW(S, R, N_{Th}, n, T_{pq})

Input: Sample data S, Radius R, Number of neighbors N_{Th}, Dimension of each grid cell n, Time interval T_{pq}.

Output: Set of grid cells G, Grid cells' noise values $N_{\mathrm{G}}^{(pred)}$

1: Create a set of grid cells G having dimension $n \times n$;
2: $N_{\mathrm{G}}^{(pred)} \leftarrow \emptyset$; $S_{pq} \leftarrow$ samples of S having time interval $T_{pq} = [t_p, t_q]$
3: **for** each grid cell $g_i \in$ G **do**
4: $m_i \leftarrow$ mid point of grid cell g_i;
5: $D_i^n \leftarrow \emptyset$;
6: **for** each sample $s_j \in S_{pq}, m_i \neq p_j$ **do**
7: //p_j is sampling location of sample s_j;
8: $d(m_i, p_j) \leftarrow$ distance between m_i and sample point p_j;
9: $D_i^n \leftarrow D_i^n \cup \{d(m_i, p_j)\}$;
10: **end for**
11: $NN_i = \{p_i^j \mid d(m_i, p_j) \leqslant R\,, d(m_i, p_j) \in D_i^n\}$;
12: **if** $|NN_i| \geq N_{Th}$ **then**
13: Sort the sample points $p_i^j \in NN_i$ to create an ordered list $L_i = \langle p_i^{j_1}, p_i^{j_2}, \ldots, p_i^{j_{|NN_i|}} \rangle$ such that $d(m_i, p_{j_1}) \leq d(m_i, p_{j_2}) \leq \ldots \leq d(m_i, p_{j_{|NN_i|}})$;
14: Create a set $L_i^{'} = \{p_i^{j_k} \mid p_i^{j_k} \in L_i\,, k \leq N_{Th}\}$;
15: **for** each neighbor $p_i^{j_k} \in L_i^{'}$ **do**
16: **if** $d(m_i, p_i^{j_k}) < 1$ **then**
17: $w_i^{j_k} = 1$;
18: **else**
19: $w_i^{j_k} = \left[\frac{1}{d(m_i, p_i^{j_k})} - \frac{1}{R}\right]$;
20: **end if**
21: $noise[p_i^{j_k}] \leftarrow$ noise value of sample $p_i^{j_k}$;
22: **end for**
23: $N_i^{(pred)} = \dfrac{\sum_{p_i^{j_k} \in L_i^{'}} w_i^{j_k} \times noise[p_i^{j_k}]}{\sum_{p_i^{j_k} \in L_i^{'}} w_i^{j_k}}$;
24: **else**
25: $N_i^{(pred)} = NULL$;
26: **end if**
27: $N_{\mathrm{G}}^{(pred)} = N_{\mathrm{G}}^{(pred)} \cup N_i^{(pred)}$;
28: **end for**
29: **return** (G, $N_{\mathrm{G}}^{(pred)}$);

between a particular midpoint and the sample points, we require $O(n_s)$ space (line 9). In the worst case, NN_i takes $O(n_s)$ space (line 11). The set $L_i^{'}$ in line 14 takes $O(N_{Th})$ space. The set $N_{\mathrm{G}}^{(pred)}$ takes $O(|\mathrm{G}|)$ space, as it needs to store the predicted noise values of |G| grid cells. The remaining variables take $O(1)$ spaces. So, overall space complexity is $O(|\mathrm{G}| + n_s + N_{Th}) \approx O(|\mathrm{G}| + n_s)$.

The main advantages of GLIDW are that it is simple to implement and it has a low computational time (i.e., low execution time). Note that low execution time leads to better *response time*

(execution time + communication time) if the method is deployed in the end-to-end framework (see Section 3.2). Despite these advantages, GLIDW suffers from some limitations. First, it is sensitive to the missing values. In other words, it is unable to predict grid cell's noise if the grid cell does not contain sufficient samples. Hence, in the case of a *sparse dataset* (i.e., dataset having missing values), if the data sparsity level (Definition 1) is high, then it usually leads to many grid cells to remain unpredicted. Second, it is a local method for predicting noise and there is no direct way to predict noise value of a grid cell that has an insufficient number of samples. This may severely affect the performance of GLIDW and, eventually, *high accuracy* may not always be achieved. Next, we propose two different methods (I-GLIDW and GLIDW-OK) to overcome the above-mentioned limitations regarding the sparse dataset and accuracy. Each of these two methods has its own advantages and disadvantages.

Definition 1 (Data Sparsity Level). The data sparsity level δ of a dataset can be defined as follows:

$$\delta = \left[1 - \frac{|\mathrm{G}^{(ns)}|}{|\mathrm{G}|}\right] \times 100\%, \tag{7}$$

where $\mathrm{G}^{(ns)}$ is the set of grid cells containing noise samples and G is the set of all grid cells. For example, if data sparsity level of a dataset $\delta = x\%$, then it means $x\%$ of the grid cells having missing values.

3.7 I-GLIDW

I-GLIDW (Iterative GLIDW) is an iterative form of GLIDW that is proposed to overcome the limitations of GLIDW. In this method of interpolation, the local effect of grid cells are propagated to neighboring grid cells in an iterative manner, as described in Algorithm 2.

The inputs to I-GLIDW are sample noise data, radius for GLIDW, radius for I-GLIDW, threshold G_{Th}, and the dimension of each grid cell. The first step of this method is to compute the grid cells' noise that has a sufficient number of samples using GLIDW. In the first iteration, for each grid cell $g_i \in \mathrm{G}$, if the noise value of that cell is not predicted yet (line 10), then we calculate the fraction η of neighboring grid cells having already predicted noise values within radius $R^{'}$ (line 12). Next, we check whether η is greater than or equal to some predefined threshold G_{Th}. If so, then we compute weights w_i^j for each neighbor $g_j \in GN_i$, where GN_i is a set of all nearest neighbors of grid cell g_i having already predicted noise values. Based on the weights w_i^j and corresponding noise value $N_j^{(pred)}$ of each grid cell $g_j \in GN_i$, we compute the predicted noise value $N_i^{(pred)}$ of g_i (line 18). But, if η is less than G_{Th}, then we assign predicted noise value $N_i^{(pred)}$ as NULL (line 21). If, at the end of an iteration all the grid noise values are not computed (line 4), then we go into the next iteration along with the noise values computed in the previous iteration. This process is continued until all the grid cells' noise values are computed. The main objective of this approach is to propagate grid cell's noise to its neighbors.

Algorithm I-GLIDW runs in time $O(|\mathrm{G}|(n_s log(n_s) + I.GN_{R'}^{(total)}))$. Let us consider the while loop iterates over I times. The for loop of line 8 iterates over $O(|\mathrm{G}|)$ times and the for loop at line 15 iterates over $O(GN_{R'}^{(total)})$ times in worst case. Hence, overall time complexity = Complexity due to GLIDW (line 1) + $O(I|\mathrm{G}|GN_{R'}^{(total)}) \approx O(|\mathrm{G}|n_s log(n_s)) + O(I|\mathrm{G}|GN_{R'}^{(total)}) \approx O(|\mathrm{G}|(n_s log(n_s) + I.GN_{R'}^{(total)}))$. However, there can be maximum of $GN_{R'}^{(total)}$ neighbors in GN_i. Hence, it requires $O(GN_{R'}^{(total)})$ space in worst case. Remaining variables take $O(1)$ spaces. Therefore, the overall space complexity of this algorithm is = space required for GLIDW + $O(GN_{R'}^{(total)}) \approx O(|\mathrm{G}| + n_s + GN_{R'}^{(total)})$.

ALGORITHM 2: I-GLIDW(S, R, N_{Th}, n, R', G_{Th}, T_{pq})

Input: Sample data S, Radius R for GLIDW, Number of neighbors N_{Th} for GLIDW, Dimension of each grid cell $n \times n$, Radius R' for iterative computation, Threshold G_{Th}, Time interval T_{pq}.

Output: Set of grid cells G, Grid cells' noise value $N_{\mathrm{G}}^{(pred)}$

$(\mathrm{G}, N_{\mathrm{G}}^{(pred)}) \leftarrow GLIDW(\mathrm{S}, R, N_{Th}, n, T_{pq})$;
$GN_{R'}^{(total)} \leftarrow$ Total number of neighbors within R';
$C = |\mathrm{G}|$;
while $C \neq 0$ **do**
  if $|\mathrm{G}| - C \neq 0$ **then**
    $C = |\mathrm{G}|$;
  end if
  for each grid cell $g_i \in \mathrm{G}$ **do**
    $C \leftarrow C - 1$;
    if $N_i^{(pred)} == NULL$ **then**
      Construct a set $GN_i = \{g_j \mid N_j^{(pred)} \neq NULL\ ,\ i \neq j\ ,\ d(m_i, m_j) \leqslant R'\}$;
      $\eta = \frac{|GN_i|}{GN_{R'}^{(total)}}$;
      if $\eta \geq G_{Th}$ **then**
        $N_{\mathrm{G}}^{(pred)} = N_{\mathrm{G}}^{(pred)} \setminus \{N_i^{(pred)}\}$;
        for each neighbor $g_j \in GN_i$ **do**
          $w_i^j = \left[\frac{1}{d(m_i, m_j)} - \frac{1}{R'}\right]$;
        end for
        $N_i^{(pred)} = \dfrac{\sum_{g_j \in GN_i} w_i^j \times N_j^{(pred)}}{\sum_{g_j \in GN_i} w_i^j}$;
        $N_{\mathrm{G}}^{(pred)} = N_{\mathrm{G}}^{(pred)} \cup \{N_i^{(pred)}\}$;
      else
        $N_i^{(pred)} = NULL$;
        $C \leftarrow C + 1$;
      end if
    end if
  end for
end while
return $(\mathrm{G}, N_i^{(pred)})$;

Figure 3 shows the process of I-GLIDW. In Figure 3(a), we predict the noise of those grid cells that have a sufficient number of noise samples. Figure 3(b) shows a particular grid cell with noise samples. We compute the grid cell's noise if it has a sufficient number of noise samples using GLIDW. Here, $R = \frac{1}{2} \times Length\ of\ diagonal = \frac{1}{2} \times \sqrt{2} \times n = \frac{n}{\sqrt{2}}$ is the radius for GLIDW, and we consider only those noise samples that are within the radius. Figures 3(c) and (d) show a single iteration of I-GLIDW to compute noise values from already predicted grid cells' noise. As shown in Figure 3(d), the total number of neighbors (i.e., neighboring grid cells' midpoints) within radius R' is 8 (i.e., $GN_{R'}^{(total)} = 8$), as the figure indicates the value of $R' = \sqrt{2} \times n$, where n is the grid cell's

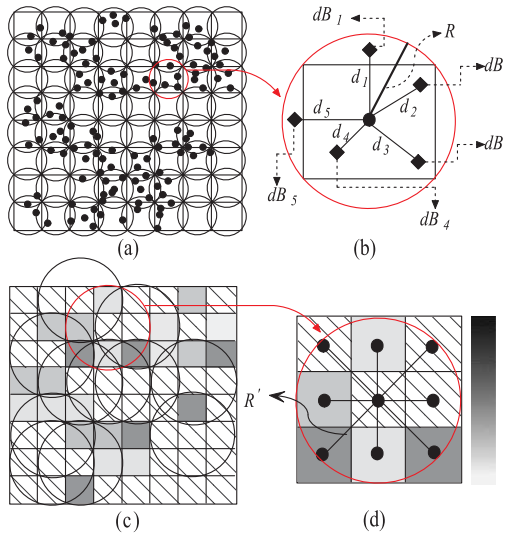


Fig. 3. Process of I-GLIDW (a) Prediction of grid cells' noise having a sufficient number of noise samples. Small dots represent sample points. The samples within each circle are responsible to find the noise value of the grid cell within that circle. (b) A particular grid cell is zoomed in. dB_i are the known noise values at sample points (square dots) having distances d_i for $i = \{1, 2, 3, 4, 5\}$ from the midpoint (circular dot) of the grid cell. (c) First iteration of I-GLIDW. (d) Prediction of an unpredicted grid cell's noise using already predicted neighbors (shaded grid cells) in an iteration.

dimension. We can generalize this as follows: Suppose there exist N number of grid cells in the row and M number of grid cells in the column within any user defined $R^{'}$, then the value of $GN_{R'}^{(total)}$ will be $N \times M - 1$. It should be noted that the threshold value is assumed to be $G_{Th} = \frac{1}{N \times M - 1}$. This assumption allows us to calculate all the missing grid cells' values in I-GLIDW even in the worst case (i.e., when the data sparsity is maximum). A large number of iterations may be required to predict all the grid cells' noise for a sparse dataset. Note that although I-GLIDW can handle the problem related to the sparse dataset, a very high data sparsity level can affect its predictive accuracy (see Section 4.4.2).

3.8 GLIDW-OK

This method is based on both GLIDW and OK methodologies. The main idea behind this method is to analyze the study area in both local and global views of the grid. Initially, we analyze grid cells locally by inferring the noise based on the sample data local to the grid cells (in our case, we are considering sample points within a predefined radius from the midpoints of the grid cells). However, when we are analyzing the grid globally, it means multiple grid cells (already predicted) are taking part in inferring the noise value of an unpredicted grid cell. For local analysis, we apply GLIDW and eventually compute those grid cells' noise value that has a sufficient amount of sample data. The result of GLIDW is then used as input for the OK method. For global analysis, the ordinary kriging method is used to predict grid cells' noise values that are not predicted yet.

Algorithm 3 describes the procedure of GLIDW-OK. Initially, we call GLIDW (line 1) to compute the grid cells' noise values that have a sufficient number of sample data points. Next, we are creating an experimental semivariogram using already predicted noise values. The experimental

semivariogram is calculated using Equation (8):

$$\widehat{\gamma}(h) = \frac{1}{2N(h)} \sum_{N(h)} (Z_i - Z_j)^2, \tag{8}$$

where h is specified lag distance, z_i and z_j are two points that are separated spatially by h. The term $N(h)$ is the total number of points that are separated by the distance h. The semivariogram is the sum of squared differences between values separated by a distance h. After that, we are choosing a model that is the best fit for the experimental semivariogram. Next, we use the OK method to predict the remaining grid cells' urban noise value by using already predicted noise and selected semivariogram model.

Standard ordinary kriging at line 4 takes $O(|\mathrm{G}|n_s^3)$ time and $O(|\mathrm{G}|n_s^2)$ space [59]. So, overall time required for GLIDW-OK is = time required for GLIDW (line 1) + $O(|\mathrm{G}|n_s^3) \approx O(|\mathrm{G}|n_s^3)$. However, the overall space required for GLIDW-OK is = space required for GLIDW (line 1) + $O(|\mathrm{G}|n_s^2) \approx O(|\mathrm{G}|n_s^2)$.

ALGORITHM 3: GLIDW-OK(S, R, N_{Th}, n , V^m, T_{pq})

Input: Sample data S, Radius R for GLIDW, Number of neighbors N_{Th} for GLIDW, Dimension of each grid cell $n \times n$, V^m is the set of variogram models, Time interval T_{pq}.

Output: Set of grid cells G, Grid cells' noise value $N_{\mathrm{G}}^{(pred')}$.

1: $(\mathrm{G}, N_{\mathrm{G}}^{(pred)}) \leftarrow GLIDW(\mathrm{S}, R, N_{Th}, n, T_{pq})$
2: Generate Semivariogram of dataset using Equation (7).
3: $model \leftarrow Modelling(V^m)$.
4: $(\mathrm{G}, N_{\mathrm{G}}^{(pred')}) = Ordinary_Kriging(\mathrm{G}, N_{\mathrm{G}}^{(pred)}, model)$
5: **return** $(\mathrm{G}, N_{\mathrm{G}}^{(pred')})$;

3.9 Performance Metrics

We use 10-fold cross validation [60] method to validate the simulation accuracy for comparing the performance of our proposed spatial interpolation methods with some baselines (e.g., IDW, OK). Some common statistical measures such as **mean absolute error (MAE), relative mean absolute error (RMAE), mean relative error (MRE), root mean squared errors (RMSE)** are used to evaluate spatial prediction accuracy and model simulation effect. MAE, RMAE, MRE, RMSE are defined as follows:

$$MAE = \frac{1}{k} \sum_{i=1}^{k} |z(x_i) - \widehat{z}(x_i)|, \tag{9}$$

$$RMAE = \frac{1}{k} \sum_{i=1}^{k} \frac{|z(x_i) - \widehat{z}(x_i)|}{z(x_i)}, \tag{10}$$

$$MRE = \frac{1}{k} \sum_{i=1}^{k} |z(x_i) - \widehat{z}(x_i)/z(x_i)|, \tag{11}$$

$$RMSE = \sqrt{\frac{1}{k} \sum_{i=1}^{k} \{z(x_i) - \widehat{z}(x_i)\}^2}, \tag{12}$$

where k represents the total number of samples in the validation set, $z(x_i)$ and $\widehat{z}(x_i)$ represent the measured and predicted noise values, respectively. MAE, RMAE, MRE, and RMSE are used to

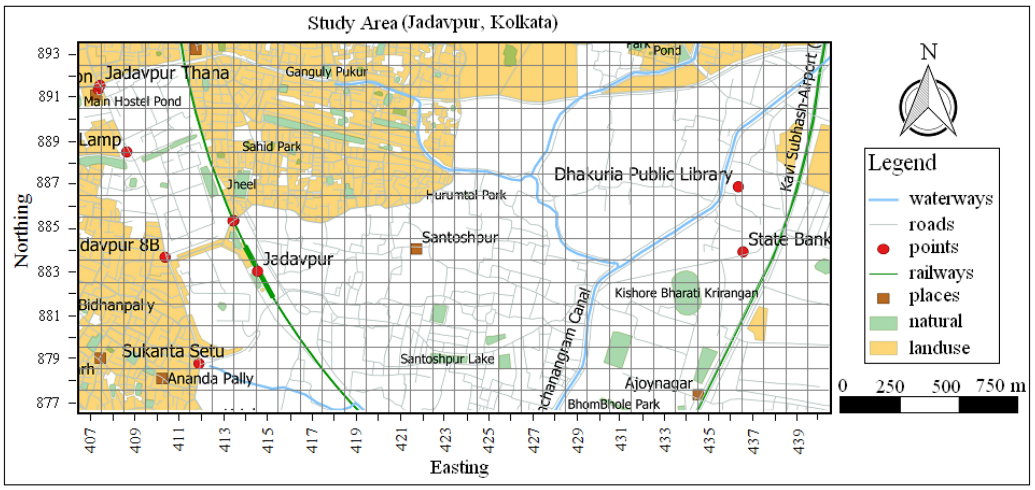


Fig. 4. Map of our study area. This map is created using QGIS [61]. The study area is part of the city of Kolkata, India. Here, Easting and Northing are East-West and North-South positions, respectively, of MGRS coordinate value.

assess the prediction accuracy of interpolation methods. Smaller values of MAE, RMAE, MRE, and RMSE represent greater accuracy in the prediction result. We also measure the **relative improvement (RI)** of our proposed spatial noise prediction models with respect to the baselines (e.g., IDW, OK). For a given model M_i RI of RMSE with respect to another model M_j can be calculated using Equation (13):

$$RI^{(RMSE)}_{M_i/M_j} = \frac{RMSE^{M_j} - RMSE^{M_i}}{RMSE^{M_j}}, \tag{13}$$

where $RMSE^{M_i}$ and $RMSE^{M_j}$ are RMSE values of model M_i and M_j, respectively. If $RMSE^{M_i}$ is smaller than $RMSE^{M_j}$, then the RI will be positive, which indicates M_i is a better model than M_j.

4 EXPERIMENT AND RESULTS

In this section, we discuss our study area and the data collection, calibration, parameter, and model selection, and at the end, we present the results and comparison. In this study, all the experiments are performed on a 64-bit Windows PC with 3.30 GHz Intel(R) Xeon(R) CPU E3-1226 v3 processor and 64 GB of RAM.

4.1 Study Area and Data Collection

The study was conducted in the Jadavpur area, which is extended between $22°48'91''N - 22°50'70''N$ latitude and $88°36'76''E - 88°40'07''E$ longitude covering an area of 7 km^2, which is a part of the city of Kolkata, India. The region is densely populated, with different land use/land cover. Figure 4 shows the study area. A total of 26 participants having diverse demographic characteristics (e.g., occupation, age, gender) are dedicatedly involved in the data collection process for a period of two years (June 2017–June 2019). Here, all the 26 participants (≈35% female and ≈65% male) are asked to collect the noise pollution data within the above-mentioned study area with GPS-enabled smartphones. All the participants live in the city of Kolkata. The demographic data of these participants are presented in Figure 5. Over half of the participants (≈65%) are **undergraduate (UG)** and **postgraduate (PG)** students, approximately 13% of them are employees, and the rest of the others (≈25%) are research scholars. Moreover, approximately

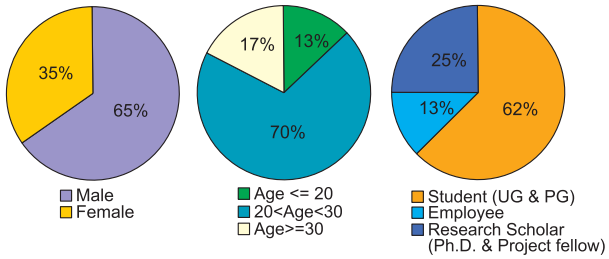


Fig. 5. Demographic data of participants.

13% of the participants have age less than 20 years, 70% of the participants have age between 20 to 30 years, and 17% of the participants have age greater than 30 years.

The data collection required participants to install our application NoiseSense [39] for Android. In the NoiseSense application, data gets recorded at an interval of five seconds in a database in the format $<user_id, sm, si, lat, lngt, ts, L_{eq}, acc_x, acc_y, acc_z, prx>$. The user interface of the application NoiseSense for logging ambient noise data is shown in Figure 7(a). Any smartphone that is calibrated with the standard **sound level meter (SLM)** can be used for data collection at any point of time. Although, in the experiment, we have used the same set of smartphones for the entire data collection process. Here, the noise pollution data samples are mainly collected at three different sessions (**Morning Session (MS), Afternoon Session (AS), and Evening Session (ES)**) of different days. We consider MS, AS, and ES as the time intervals [9:00AM–11:00AM], [3:00PM–5:00PM], and [6:00PM–8:00PM], respectively. Here, our main motive behind selecting those time intervals for data collection is to generate noise pollution maps for the rush hours in the city. After context-aware data cleaning (see Section 3.4), there are a total of 93,786 data samples present in the participatory sensing-based noise pollution dataset. Figure 6 depicts the spatial distribution of sampling locations in the study area. Note that the data sparsity levels (δ) for MS, AS, and ES sessions are 5.1%, 6.2%, and 4.9%, respectively. Whereas, combining all sessions' data reduces the sparsity levels to 2.3%. For ground truth data collection, we use standard noise-measuring devices such as Amprobe SM10 [62] and Mextech SL-4012 sound level meter [63]. In this work, the participants carry both standard device (for ground truth data) and smartphones (for participatory sensing data) during data collection, and they collect ground truth data in the same locations where data are collected using smartphones.

The data that are collected using ground truth devices are used to validate the proposed predictive models. The details of the spatio-temporal aspects of the data along with our approach for the evaluation of spatial noise mapping are illustrated using Figure 8. Note that we have used 10-fold **cross validation (CV)** for the evaluation. As shown in Figure 8, for each iteration of the CV, the data from the spatial locations for model building (i.e., training purpose) and validation (i.e., testing purpose) are disjoint sets. Hence, the ground truth data in the test set (to which the predicted values are compared) are in different spatial locations from those data that are used for the model building. The noise values are predicted on the test locations after the model is built, and then the predicted values are compared to the ground truths at those locations (i.e., test locations) to measure the metric.

4.2 Calibration

As we have discussed in Section 3.3, we need to estimate the calibration offset ξ to correctly measure L_{eq}. Calibration is necessary, because, without good calibration, the smartphone sensors may produce data that are inaccurate. Now, to calculate calibration offset ξ, we measure and

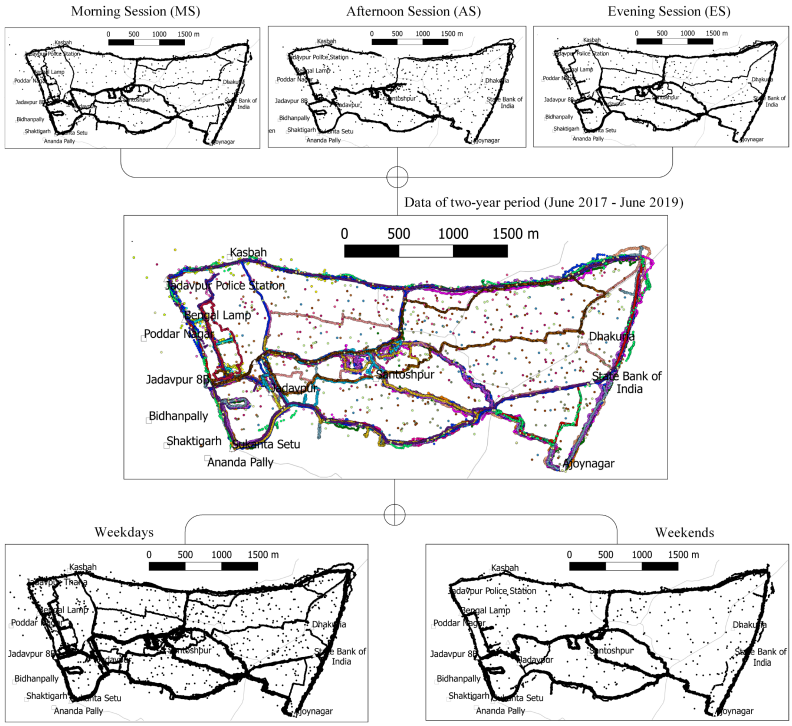


Fig. 6. Spatial distribution of the sampling locations in the study area for the morning session (MS), afternoon session (AS), evening session (ES). Data collected over a two-year period, weekdays, and weekends. Different colors in the map indicate sampling locations corresponding to different participants.

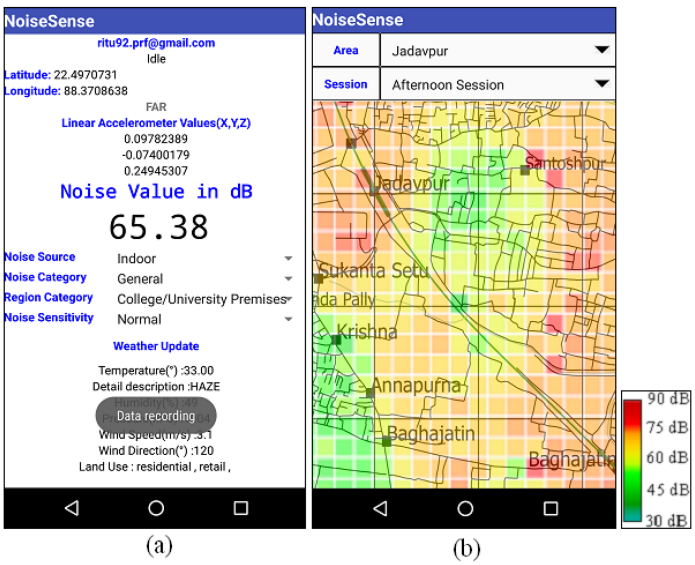


Fig. 7. (a) Screenshot of data logging activity using our Android application (NoiseSense [39]). (b) Noise pollution map as a service.

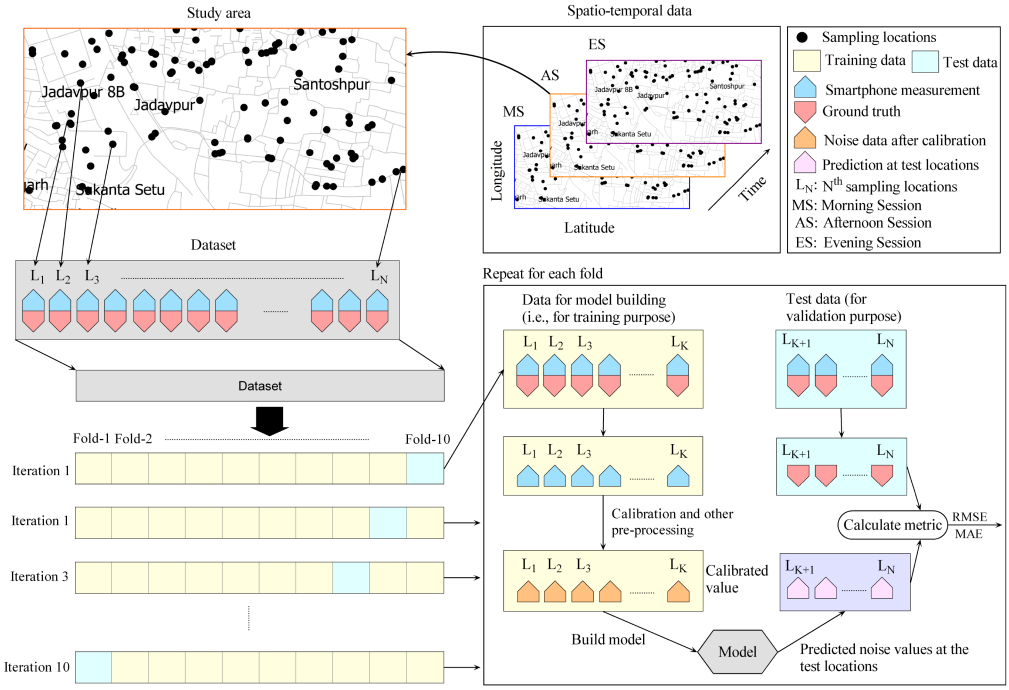


Fig. 8. The details of the spatio-temporal data and the approach for evaluation of the spatial noise mapping techniques.

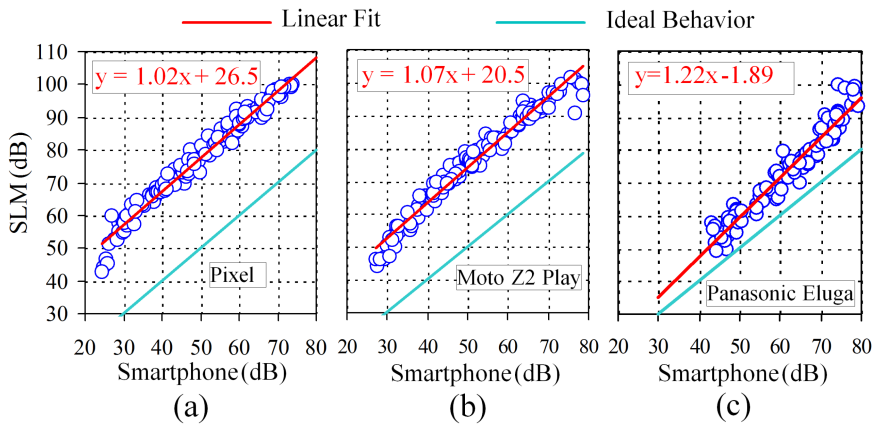


Fig. 9. Scatter plots of SLM readings and corresponding smartphone readings for three different smartphones (a) Google Pixel (b) Moto Z2 Play (c) Panasonic Eluga. A regression line (red line) shows linear relationship between smartphone noise-level measurements and true noise levels. The difference between regression line (red line) and ideal line (cyan) gives calibration offset ξ.

compare the readings from SLM and different smartphones under different sound levels. For example, the scatter plot for SLM readings and their corresponding readings of smartphones are shown in Figure 9 for three smartphone models (Google Pixel, Moto Z2 Play, and Panasonic Eluga). Moreover, we use linear regression (least squares fitting technique) to show that there

Table 3. Linear Models and Quality of Linear Regression

Smartphone Device	Linear Model $[f(x) = p_1x + p_2]$		Quality of Linear Regression [Goodness of fit]			
	p_1	p_2	SSE	R^2	Adjusted R^2	RMSE
Moto Z2 Play	1.073	20.52	1004	0.9653	0.9651	2.845
Panasonic Eluga	1.225	-1.89	1059	0.9397	0.9391	3.27
Google Pixel	1.023	26.55	756.2	0.9713	0.9711	2.47
Lenovo K3 Note	0.969	11.61	760	0.9595	0.9592	2.571
Lava Pixel v3	0.9412	13.88	101.8	0.9843	0.9841	1.302
Redmi 3S Prime	1.013	18.99	338.4	0.9693	0.9688	2.395
Galaxy J7 Nxt	0.9448	16.46	321.7	0.9755	0.9752	2.83
Galaxy J7 Prime	1.0396	15.74	216.1	0.9715	0.9713	2.55

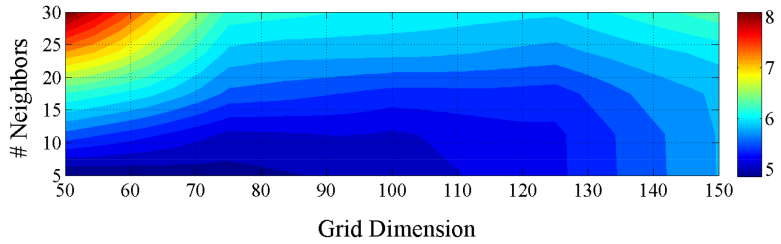


Fig. 10. Prediction accuracy (RMSE values) for different combinations of neighboring threshold N_{Th} and cell dimension n.

exists a linear relationship between smartphone noise-level measurements and true noise levels (i.e., SLM readings). For different smartphone models, we find different linear models and calculate the metrics of goodness of fit to show the quality of linear fittings. These linear models and their goodness of fit for some smartphone models are shown in Table 3. Suppose, for any smartphone model sm_i, the linear model is given by Equation (14), where x is the smartphone noise reading. Now, for any noise reading L_{eq} in a smartphone model sm_i, the corresponding calibration offset is calculated by Equation (15). We observe that for different smartphone models, the calibration offset not only varies from phone to phone but also may vary under different sound levels.

$$f^{sm_i}(x) = p_1x + p_2 \tag{14}$$

$$\xi^{sm_i}(L_{eq}) = f^{sm_i}(L_{eq}) - L_{eq} \tag{15}$$

4.3 Parameters and Model Selection

Two parameters N_{Th} and n of GLIDW need to be known to predict noise within a grid locally. We change the values of the threshold N_{Th} as well as the dimension of the grid cell n to find an optimal combination of them, so this optimizes our prediction accuracy. Figure 10 shows the RMSE values for different settings of N_{Th} and n. It can be seen that, if n is in range $50 \sim 80$ meters and N_{Th} is around 10, then the lowest RMSE is achieved, but the problem is that a small cell dimension like $50 \sim 80$ meters will cause few noise samples within a grid cell, which leads to many grid cells remaining unpredicted. It should be noted that if n is in the range $80 \sim 100$ meters and N_{Th} is around 10, then we also got a decent result. In our work, we have taken $n = 100$ (i.e., 100×100 grid cells)

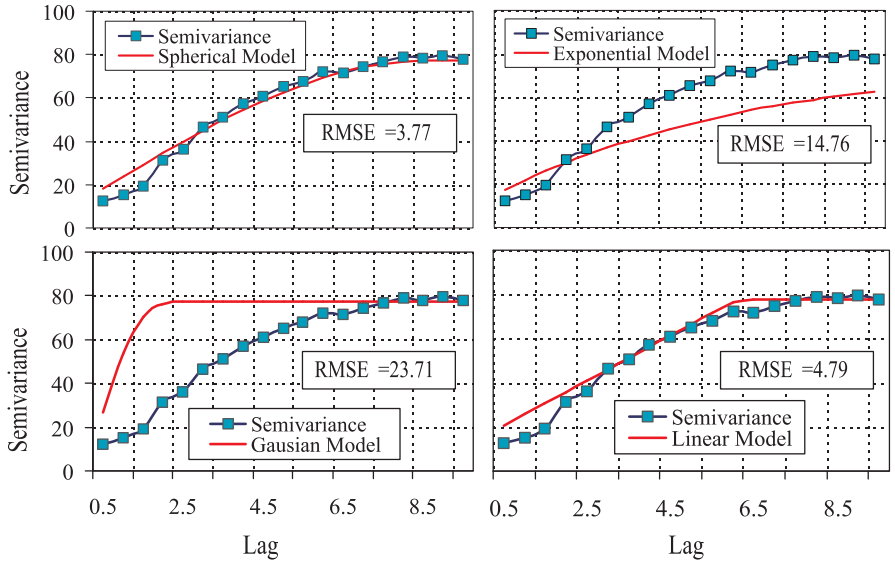


Fig. 11. Fitting of four commonly used semivariogram models (Spherical, Exponential, Gaussian, and Linear) to the experimental semivariogram.

and $N_{Th} = 10$. Note that the value of radius R in GLIDW depends on the cell dimension n. As shown in Figure 3(b), the value of R is half the length of the diagonal of a grid cell. Hence, in this case, $R = \frac{1}{2} \times Length\ of\ diagonal = \frac{1}{2} \times \sqrt{2} \times n = \frac{n}{\sqrt{2}} = \frac{100}{\sqrt{2}}$. It is also important to note that the parameter selection here is performed based on the whole dataset.

For GLIDW-OK, Figure 11 shows the fitting of different models to the semivariogram generated from the sample data. We select the spherical model, which is the best fit compared to other models such as exponential model, linear model, and so on. The parameters of the model are: nugget variance (C_o)= 12.354196, sill (C)= 77.422, and range (a)= 8.633. Hence, the fitted model is as follows:

$$\gamma(h) = 12.354196 + 77.422 \times \left(\frac{3h}{17.266} - \frac{1}{2} \times \left(\frac{h}{8.633} \right)^3 \right). \tag{16}$$

In the case of the baseline IDW, the best value of the power parameter p is found out by testing IDW with different values of p. Note that $p = 2$ is selected, as it provides the best performance. However, spherical variogram is used as a variogram model in case of the baseline OK, as it is found to be the best fit model as compared to other models such as exponential model, linear model, and so on.

4.4 Results

Each of the three methods (GLIDW, I-GLIDW, and GLIDW-OK) that we have proposed produces a fairly different noise pollution map on participatory sensing data of our study area. As shown in Figure 12, GLIDW produces a noise map of our study area that is based on afternoon session data (3:00 PM–5:00 PM). The figure reflects that a lot of grid cells remain unpredicted by GLIDW. Recall from Section 3.6, this is one of the limitations of GLIDW. Figure 13 shows the noise maps produced at different iterations (total 15 iterations, I_1-I_{15}) by I-GLIDW. Two limitations of I-GLIDW can be identified clearly from this figure (Figure 13). We can see at the bottom left portion of the noise map of iteration I_1 that there is an area with high noise values (deep red) surrounded by unpredicted grid

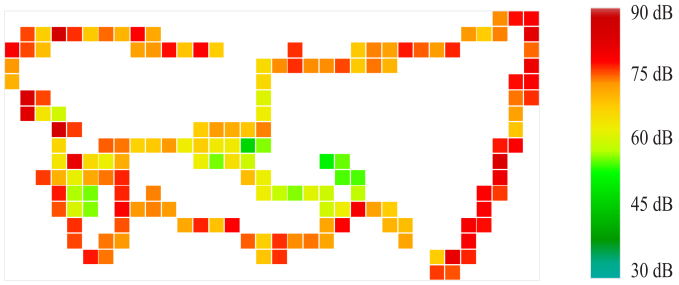


Fig. 12. Noise map produced by method GLIDW. Grid cells having an insufficient number of sample points remain unpredicted (white-colored regions).

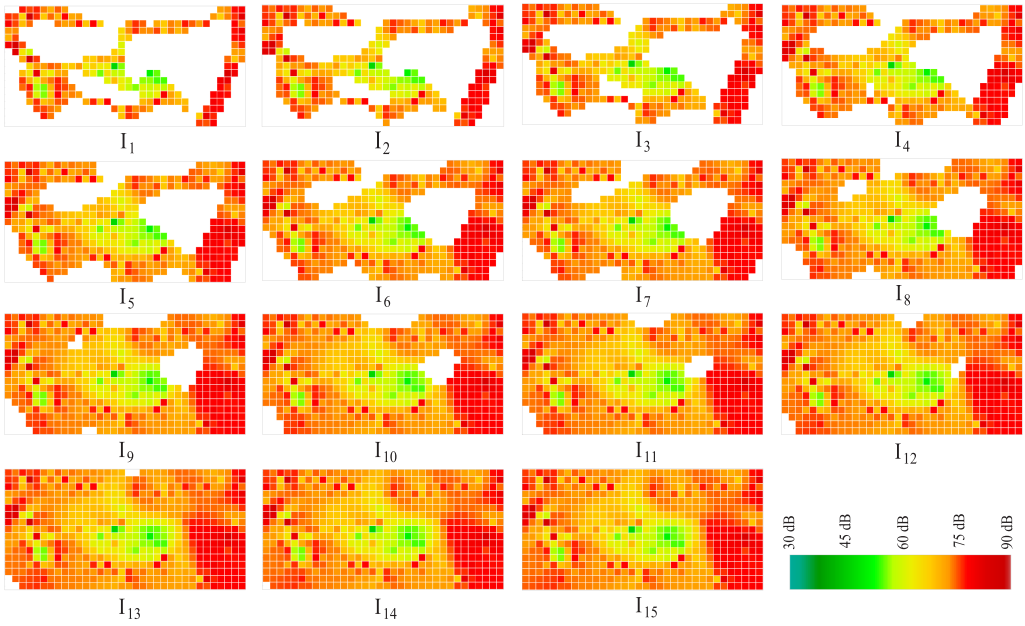


Fig. 13. Noise maps produced in different iterations ($I_1 - I_{15}$) by method I-GLIDW.

cells having missing values. As iterations go on, noise values of those unpredicted grid cells become predicted by I-GLIDW. If we carefully see different iterations, then I-GLIDW propagates the initial noise values (i.e., the deep red portion) to the unpredicted grid cells. This is the first limitation of I-GLIDW. Moreover, we can see, to generate the final noise map, I-GLIDW takes too many iterations (in this case 15; it can be even more for a different distribution of sample points), which increases its computational cost. Now, Figure 14 shows the noise map produced by GLIDW-OK on the same dataset. Later, we evaluate these methods in terms of different performance metrics.

4.4.1 Performance Comparison with Baselines. We compare our proposed spatial interpolation methods with two baselines. The first one is **Inverse Distance Weighting (IDW)** [42] and the second one is **Ordinary Kriging (OK)**[50]. The cross-validation results of noise levels estimation using GLIDW, I-GLIDW, GLIDW-OK, IDW, and OK are shown in Table 4. The results show that there is not a single method that clearly dominates over other methods in terms of predictive accuracy (MAE, RMAE, RME, RMSE) in all three sessions (e.g., Morning Session (MS), Afternoon

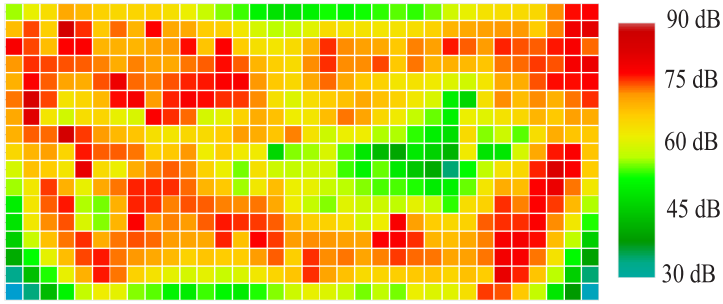


Fig. 14. Noise map produced by method GLIDW-OK.

Table 4. Performance Comparison with Baselines

Session	Methods	Quality of Models			
		MAE	RMAE	MRE	RMSE
MS (9:00AM-11:00AM)	GLIDW	8.1094	0.1099	73.1977	9.6156
	I-GLIDW	6.1134	0.1059	72.4287	7.3737
	GLIDW-OK	5.8477	0.1028	72.4512	7.1441
	IDW	6.4452	0.1063	72.4171	7.8882
	OK	6.0316	0.1053	72.8635	7.2423
AS (3:00PM-5:00PM)	GLIDW	6.9134	0.0992	70.5698	8.4805
	I-GLIDW	5.9648	0.0921	66.9346	7.4226
	GLIDW-OK	6.3137	0.0935	66.9213	7.8568
	IDW	6.3707	0.0956	66.9404	8.4793
	OK	6.3353	0.0943	66.9923	7.9312
ES (6:00PM-8:00PM)	GLIDW	9.5457	0.143	75.4855	10.6806
	I-GLIDW	5.1884	0.0792	72.8963	6.9227
	GLIDW-OK	6.3966	0.1019	72.9158	7.2958
	IDW	7.1925	0.1063	72.8644	8.9272
	OK	6.5141	0.1043	72.4623	7.9513

MS: Morning Session; AS: Afternoon Session; ES: Evening Session; MAE: Mean Absolute Error; RMAE: Relative Mean Absolute Error; MRE: Mean Relative Error; RMSE: Root Mean Squared Error; GLIDW: Grid-based Local Inverse Distance Weighting; I-GLIDW: Iterative Inverse Distance Weighting; GLIDW-OK: Grid-based Local Inverse Distance Weighting- Ordinary Kriging; IDW: Inverse Distance Weighting; OK: Ordinary Kriging.

Session (AS), Evening Session (ES)) of a day. This is because datasets of different sessions may have different data sparsity levels and the impact of data sparsity on the predictive accuracy of different methods may not be the same (see Section 4.4.2). The highest error is obtained with GLIDW, irrespective of the sessions. We have also shown the scatter plots (see Figure 15) of the original noise levels (observations) versus the predicted noise levels (estimates) from GLIDW, I-GLIDW, GLIDW-OK, IDW, and OK in different sessions of a day. In all the methods, the estimated mean noise (μ_y) is almost similar to the observed mean noise (μ_x)(i.e., actual mean). We can also see that estimated standard deviation (σ_y) is smaller than the observed standard deviation (σ_x) (smoothness). Additionally, the correlation coefficients (ρ_{xy}) between observations (x) and estimates (y) are fairly good for both I-GLIDW and GLIDW-OK in all three sessions. The method IDW smoothed out the prediction of noise levels at the most. However, the smoothing effect of prediction is the least for

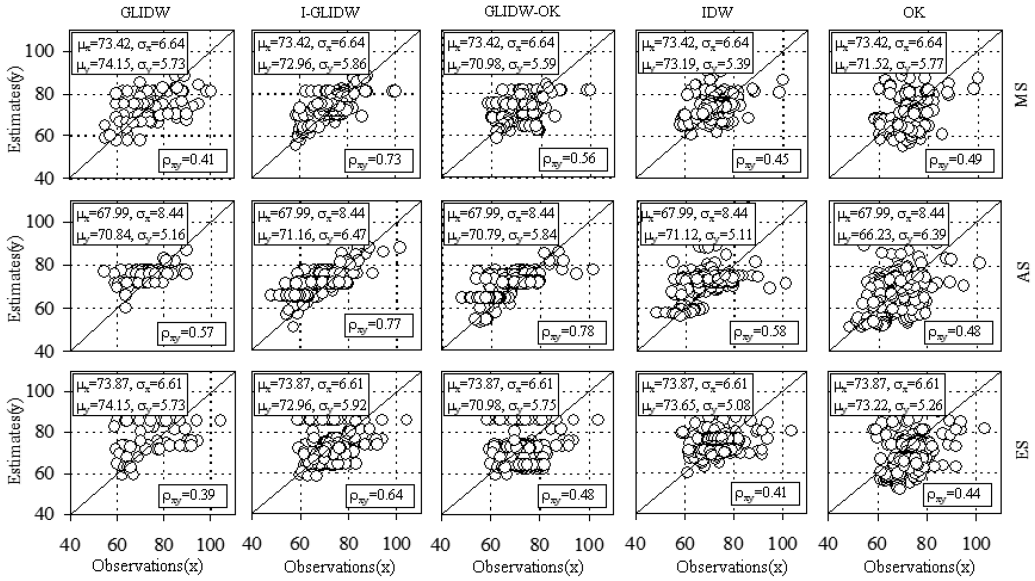


Fig. 15. Observation vs. Prediction plots of different methods (GLIDW, I-GLIDW, GLIDW-OK, IDW, OK) in different sessions (MS-Morning Session, AF-Afternoon Session, ES-Evening Session) of a day.

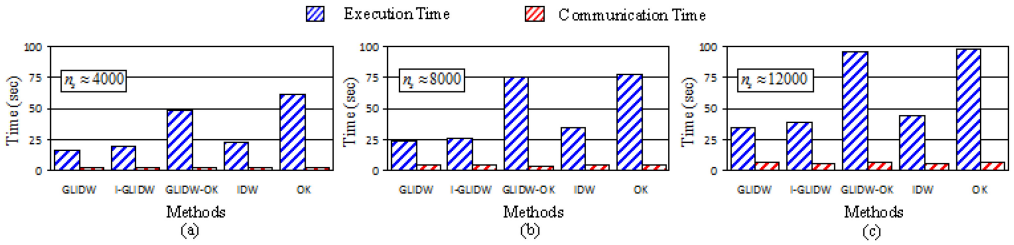


Fig. 16. Execution time (T_E) and communication time (T_c) for different methods by varying the number of data samples n_s. (a) T_E and T_c with $n_s \approx$ 4,000; (b) T_E and T_c with $n_s \approx$ 8,000; (c) T_E and T_c with $n_s \approx$ 12,000.

I-GLIDW. It is also important to note that, for each of the MS, AS, and ES rows of Figure 15, the distribution of observations for GLIDW is different from the rest of the other methods. It is because, as discussed in Section 3.6, one of the major limitations of GLIDW is that it is unable to predict grid cells' noise if the grid cells do not contain a sufficient number of noise samples. In other words, if for any grid cell g_i, the condition $NN_i < N_{Th}$ holds true in Algorithm 1, then GLIDW returns NULL, where NN_i is the number of nearest neighbor samples within radius R. Hence, in the case of GLIDW, some of the predicted values could be NULL. Whereas other methods (I-GLIDW, GLIDW-OK, IDW, OK) always come up with a non-NULL predicted value. Now, Figures 16(a), (b), and (c) compare the execution time (T_E) and communication time (T_c) of the proposed methods with the baselines by varying the number of urban noise samples ($n_s \approx$4,000, $n_s \approx$8,000, $n_s \approx$12,000). Here, the execution time T_E of OK is found to be maximum compared to the execution time of other approaches. However, the communication time (T_c) does not vary significantly across the methods for the same number of samples. We also show the relative improvement between each pair of methods in the form of three heatmaps (see Figure 17) one for each session (MS, AS, ES). We can clearly see from Figure 17 that the relative improvements are fairly good for I-GLIDW and GLIDW-OK.

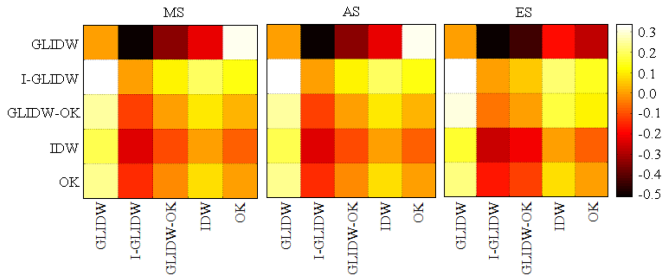


Fig. 17. Relative Improvements between all possible pairs of methods in different sessions of a day.

Table 5. Features and Theoretical Comparison among IDW, OK, GLIDW, I-GLIDW, and GLIDW-OK

Basis of comparison	IDW [42]	OK [41, 49]	GLIDW	I-GLIDW	GLIDW-OK
Type of Interpolation	Deterministic	Geostatistical	Deterministic method	Deterministic method	Combined method (Deterministic + Geo-statistical)
Exact/Inexact[a]	Exact method	Can be exact/inexact	Exact method	Exact method	Exact method
Local/Global[b]	Local method	Local method	Local method	Local (can be forced to be global)	Local method
Convex/non-convex[c]	Convex method	Can be non-convex	Convex method	Convex method	Can be non-convex
Limitation	If data samples are sparse/uneven, then it may not work well	The assessment of error largely depends on semivariogram, data sample distribution, etc.	It is very sensitive to missing values and unable to predict all grid cells' noise	Number of iterations can be very high	Sometimes the predicted values are unrealistic when the dataset is very sparse
Computation time	$O(\beta n_s^2)$	$O(\beta n_s^3)$	$O(\|G\| n_s log(n_s))$	$O(\|G\|(n_s log(n_s) + I.GN_{R'}^{(total)}))$	$O(\|G\| n_s^3)$
Space requirement	$O(\beta n_s)$	$O(\beta n_s^2)$	$O(\|G\| + n_s)$	$O(\|G\| + n_s + GN_{R'}^{(total)})$	$O(\|G\| n_s^2)$
Suitability	It works well when data points are sufficiently dense	If enough data samples are available to compute semivariograms, then it provides good results	If sufficient sample points are available in each grid cell, then it performs well	If spatial variation in noise levels is not abrupt, then it works well	It can handle sparse as well as irregularly spaced data samples very well

n_s: Total number of sample points; β: Total number of unknown locations for estimating noise; $|G|$: Total number of grid cells; I: Number of iterations in $I - GLIDW$; $|GN_i|$: Number of neighbors; $GN_{R'}^{(total)}$: Total number of neighbors within radius R'.

[a] An Exact method predicts a value that is the same as the observed/known value at a sampled point. An inexact method predicts a value that differs from its observed/known at a sampled point.

[b] A local method uses the samples within a search window to estimate an unknown value. A global method uses every known point in the region of interest to estimate an unknown value.

[c] A convex method generates predicted values that lie somewhere in between the minimum and maximum of the observed/known sample values. A non-convex method can generate predicted values that lie outside the range of minimum and maximum of the observed/known sample values.

The features of the spatial interpolation techniques that we have proposed may vary from one technique to another. Hence, we provide a theoretical comparison among GLIDW, I-GLIDW, GLIDW-OK, IDW, and OK in Table 5.

4.4.2 Discussions and Observations. The predictive accuracy of different methods depends on how sparse the dataset is [51]. To study the effectiveness of our proposed methods, the performance (in terms of RMSE and response time) is computed under different data sparsity levels (Definition 1). More specifically, it is necessary to explore how the predictive *accuracy* and *response time* of the methods depend on the data sparsity levels of the *sparse datasets*. To perform the experiment, sparse datasets with different data sparsity levels (δ) must be created. For this purpose, first a sub-region is selected where each of the grid cells of the sub-region must contain some noise samples.

Table 6. Performance of Methods under Different Data Sparsity Levels

Methods	δ = 0%		δ = 10%		δ = 20%		δ = 30%		δ = 40%		δ = 50%		δ = 60%		δ = 70%		δ = 80%	
	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE	T_R	RMSE
GLIDW	25.40	6.15	23.30	14.68	19.62	15.07	17.66	15.28	15.30	15.44	13.28	15.62	10.34	15.86	8.76	16.12	6.61	16.23
I-GLIDW	27.24	6.17	24.21	9.45	23.80	10.01	21.56	11.13	17.75	11.29	15.14	12.76	12.91	13.32	12.11	14.17	10.44	15.69
GLIDW-OK	81.34	6.15	72.88	10.65	63.04	10.86	51.39	11.34	39.15	11.61	33.71	11.69	29.78	11.76	25.17	11.98	21.82	12.62
IDW	35.32	6.87	32.42	14.62	27.44	15.01	23.62	15.23	20.47	15.53	16.37	15.66	13.59	15.83	12.19	16.22	10.64	16.49
OK	77.75	6.69	73.26	12.08	68.28	13.12	62.75	13.77	51.35	14.03	43.72	14.62	36.88	15.11	29.44	15.97	23.39	16.33

δ: Data sparsity level; T_R: Response time (in seconds), which is the sum of method execution time T_E and communication time T_C; RMSE: Root Mean Squared Error.

It should be noted that a sub-region of the above-mentioned type is necessary to be selected for the data sparsity experiment instead of considering the whole study area. It is because the study area already contains many grid cells having no noise samples, hence the whole study area cannot be directly used to achieve the data sparsity levels less than what it originally has. Now, the datasets with different δ values (i.e., data sparsity levels) can be created by randomly eliminating the girds with the noise samples from that sub-region. For instance, a dataset with $\delta = x\%$ can be created by randomly eliminating $x\%$ of the grids (along with the noise samples) from that sub-region. For the experiment, the datasets are generated corresponding to 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, and 80% data sparsity levels. Table 6 shows predictive accuracy (RMSE) and response time (T_R) of our proposed methods under different data sparsity levels. From that table, we can see if $\delta = 0\%$ (i.e., when all grid cells contain noise samples), then GLIDW dominates over other methods in terms of both accuracy and response time. So, if a dataset is very dense, then we prefer to use GLIDW to produce a noise map. When δ is in range (10%, 40%) I-GLIDW produces a high accuracy noise map with a good response time. So, if data sparsity is medium, then we prefer to use I-GLIDW. If data sparsity is high ($\delta > 40\%$), then GLIDW-OK dominates over other methods in terms of predictive accuracy. Though, in this case, GLIDW-OK takes more time to produce the noise map, we prefer to use this method, as it provides much better accuracy than other methods. In summary, from the performance evaluation of the methods with respect to the three scenarios (sparse dataset, high accuracy, response time), it is confirmed that the proposed methods dominate the baselines (IDW, OK). Now, it should be important to mention how we measure RMSE for GLIDW where δ is not zero in the above data sparsity experiment. In the case of $\delta > 0$, there will be some grid cells having no noise samples. In GLIDW, if there are no noise samples in a grid cell, then GLIDW is unable to predict the noise value for that grid cell and returns NULL. But, to calculate overall RMSE, we need to have the predictions for all the grid cells. In GLIDW, to overcome this problem, we replace NULL values with the mean of the non-NULL predictions.

5 POTENTIAL ISSUES

In this section, some potential issues are identified and discussed that are related to the proposed participatory sensing-based dynamic noise map generation. These issues can be addressed for further improvement of the proposed noise map generation techniques. First, in different urban settings, the noise pollution levels depend on a diverse set of static contexts (e.g., residential area, commercial area) and dynamic contexts (e.g., traffic flow, crowd flow). There can also be different overlapping among the contexts, which may influence the ambient noise pollution levels differently. The context-based interpolation may be considered for map generation so the spatial variability of ambient noise pollution levels can be better explored. Second, the effect of buildings and topography is also an important factor not taken into account in this work. Existing literature

ensures that buildings, in particular, can amplify or reduce sound [64]. In this work, the proposed methods make the assumption that the urban noise measurements can be interpolated over a Euclidean grid. The effect of buildings and topography could be incorporated in the model for potential improvements. Third, despite the fact that the participatory sensing-based ambient noise pollution data collection has a lot of advantages, it also suffers from some problems. Participatory sensing cannot ensure spatial uniformity among the samples in an urban setting i.e., data samples may not be uniformly distributed over the space. In this work, spatial uniformity is achieved to some extent by setting an equal sampling interval. Since data are automatically collected using users' smartphones at regular and fixed time interval while the participants are walking, we can loosely say that at normal walking speed the consecutive samples have high chance to be almost equal distance apart. But, we believe that it is not sufficient to ensure spatial uniformity of the samples, and this is an issue on which dedicated research is required to achieve better uniformity of samples that are collected using participatory sensing. Fourth, the participatory sensing-based data collection process creates a dense dataset that is used to generate high granular noise pollution maps. But, some areas may suffer from a lack of noise data samples if those areas are not covered by any participant. It is because the area covered by different participants depends on their mobility patterns.

6 CONCLUSION AND FUTURE WORK

In this article, an IoT-cloud-based end-to-end system is proposed that can provide a high granular and spatially continuous noise map as a service to the users. More specifically, we focus on the IoT-based approach that uses participatory sensing to generate dynamic noise maps with the help of our proposed spatial interpolation techniques (GLIDW, I-GLIDW, GLIDW-OK). The proposed techniques can address diverse scenarios such as sparse datasets, high accuracy, better response time, and so on. The IoT-cloud-based end-to-end system reduces the overall cost of noise pollution monitoring and can aid citizens, policy makers, and city authorities with high granular noise maps. The high granular noise maps not only help citizens to assess their individual exposure to ambient noise in their daily environment but also allow them to keep track of noise pollution levels in their surroundings and eventually help to improve awareness. In this work, proper calibration and context-aware data cleaning techniques are designed and implemented as pre-processing techniques for the proposed participatory sensing-based noise map generation methods. Naive Bayes classifier is employed for context classification. The performances of the proposed techniques are evaluated using participatory sensing data collected by participants over a period of two years (June 2017–June 2019) in the Jadavpur area of the city of Kolkata. Participants of diverse demographic characteristics (e.g., occupation, age, gender, place of residence) take part in the data collection process. We also compare the performance of these spatial interpolation methods for noise map generation with two baseline methods, namely, Inverse Distance Weighting (IDW) and Ordinary Kriging (OK). The experimental results show that the proposed methods outperform the baseline methods. The results also indicate that GLIDW performs well when the dataset is very dense. However, I-GLIDW and GLIDW-OK dominate if data sparsity levels are medium and high, respectively. It is also observed that the Relative Improvements (RI) of I-GLIDW and GLIDW-OK are always positive as compared to the baseline models. It is also interesting to note that if we have spatially dense datasets, then GLIDW dominates over other methods, and for such a scenario, I-GLIDW or GLIDW-OK is not needed. In this case, GLIDW can be used as an independent and separate method, and it is not needed for GLIDW to act as a preprocessing algorithm. However, if the dataset is sparse, only GLIDW is not sufficient. In such cases, the algorithms such as I-GLIDW and GLIDW-OK are required and GLIDW is used as a preprocessing algorithm.

In the future, we would like to extend our work to handle the potential issues (discussed in Section 5) that we have identified during this work. For instance, in IoT-based approach the samples that are collected using participatory sensing-based data collection techniques are not uniformly distributed over the space. These non-uniformly distributed samples need to be handled carefully to achieve further better results. As discussed in Section 5, context-based interpolation may be a good choice to further explore the spatial variability of ambient noise pollution levels. Eventually, as our future work, we would like to improve our proposed noise mapping techniques by using context-based interpolation. Also, we plan to improve the efficiency of the end-to-end system and also want to use the predicted noise values to conduct a personalized pollution free path recommendation system.

ACKNOWLEDGMENTS

We thank Mr. Subhasish Ghosh, Mr. Abhimanyu Kumar, and the volunteers from Jadavpur University for their assistance.

REFERENCES

[1] Mariola Śliwińska-Kowalska and Kamil Zaborowski. 2017. WHO environmental noise guidelines for the European region: A systematic review on environmental noise and permanent hearing loss and tinnitus. *Int. J. Environ. Res. Pub. Health* 14, 10 (2017), 1139.

[2] Dibyendu Banerjee et al. 2012. Research on road traffic noise and human health in India: Review of literature from 1991 to current. *Noise Health* 14, 58 (2012), 113.

[3] Charlotte Clark and Katarina Paunovic. 2018. WHO environmental noise guidelines for the European region: A systematic review on environmental noise and quality of life, wellbeing and mental health. *Int. J. Environ. Res. Pub. Health* 15, 11 (2018), 2400.

[4] Barbara Hoffmann. 2018. Noise and hypertension—A narrative review. *Curr. Epidemiol. Rep.* 5, 2 (2018), 70–78.

[5] Mathias Basner and Sarah McGuire. 2018. WHO environmental noise guidelines for the European region: A systematic review on environmental noise and effects on sleep. *Int. J. Environ. Res. Pub. Health* 15, 3 (2018), 519.

[6] Elise Van Kempen, Maribel Casas, Göran Pershagen, and Maria Foraster. 2018. WHO environmental noise guidelines for the European region: A systematic review on environmental noise and cardiovascular and metabolic effects: A summary. *Int. J. Environ. Res. Pub. Health* 15, 2 (2018), 379.

[7] WHO. 2009. Night noise guidelines for Europe. Retrieved from http://www.euro.who.int/__data/assets/pdf_file/0017/43316/E92845.pdf?ua=1. (2009).

[8] WHO. 2019. Burden of disease from environmental noise—Quantification of healthy life years lost in Europe. Retrieved from https://www.who.int/quantifying_ehimpacts/publications/e94888.pdf?ua=1. (2019).

[9] WHO. 2018. Environmental Noise Guidelines for the European Region. (2018). Retrieved from http://www.euro.who.int/__data/assets/pdf_file/0008/383921/noise-guidelines-eng.pdf?ua=1.

[10] Ryan C. Lewis, Robyn R. M. Gershon, and Richard L. Neitzel. 2013. Estimation of permanent noise-induced hearing loss in an urban setting. *Environ. Sci. Technol.* 47, 12 (2013), 6393–6399.

[11] EU Directive. 2002. Directive 2002/49/EC of the European parliament and the Council of 25 June 2002 relating to the assessment and management of environmental noise. *Offic. J. Eur. Commun. L* 189, 18.07 (2002), 2002.

[12] Erik Salomons, Dirk Van Maercke, Jérôme Defrance, and Foort de Roo. 2011. The harmonoise sound propagation model. *Acta Acust. United Acust.* 97, 1 (2011), 62–74.

[13] Stylianos Kephalopoulos, Marco Paviotti, Fabienne Anfosso-Lédée. 2012. Common noise assessment methods in Europe (CNOSSOS-EU). Publications office of the European union, 180 pages. DOI: 10.2788/31776

[14] Kang-Ting Tsai, Min-Der Lin, and Yen-Hua Chen. 2009. Noise mapping in urban environments: A Taiwan study. *Appl. Acoust.* 70, 7 (2009), 964–972.

[15] Weigang Wei, Timothy Van Renterghem, Bert De Coensel, and Dick Botteldooren. 2016. Dynamic noise mapping: A map-based interpolation between noise measurements with high temporal resolution. *Appl. Acoust.* 101 (2016), 127–140.

[16] Nicolas Maisonneuve, Matthias Stevens, Maria E. Niessen, and Luc Steels. 2009. NoiseTube: Measuring and mapping noise pollution with mobile phones. In *Information Technologies in Environmental Engineering*. Springer, 215–228.

[17] Andrea Capponi, Claudio Fiandrino, Burak Kantarci, Luca Foschini, Dzmitry Kliazovich, and Pascal Bouvry. 2019. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE Commun. Surv. Tutor.* 21, 3 (2019), 2419–2465.

[18] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. 2015. Incentives for mobile crowd sensing: A survey. *IEEE Commun. Surv. Tutor.* 18, 1 (2015), 54–67.
[19] Yongxin Tong, Lei Chen, and Cyrus Shahabi. 2017. Spatial crowdsourcing: Challenges, techniques, and applications. *Proc. VLDB Endow.* 10, 12 (2017), 1988–1991.
[20] Andrei Tamilin, Iacopo Carreras, Emmanuel Ssebaggala, Alfonse Opira, and Nicola Conci. 2012. Context-aware mobile crowdsourcing. In *Proceedings of the ACM Conference on Ubiquitous Computing*. 717–720.
[21] Silvia Nittel. 2015. Real-time sensor data streams. *SIGSPATIAL Special* 7, 2 (2015), 22–28.
[22] Philip E. Brown, Tamraparni Dasu, Yaron Kanza, and Divesh Srivastava. 2019. From rocks to pebbles: Smoothing spatiotemporal data streams in an overlay of sensors. *ACM Trans. Spatial Algor. Syst.* 5, 3 (2019), 1–38.
[23] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 1–55.
[24] Takahiro Yabe, Kota Tsubouchi, Akihito Sudo, and Yoshihide Sekimoto. 2016. Predicting irregular individual movement following frequent mid-level disasters using location data from smartphones. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.
[25] Asif Iqbal Middya, Sarbani Roy, Joy Dutta, and Rituparna Das. 2020. JUSense: A unified framework for participatory-based urban sensing system. *Mobile Networks and Applications* 25 (2020), 1249–1274. https://doi.org/10.1007/s11036-020-01539-x
[26] David Hasenfratz, Olga Saukh, Silvan Sturzenegger, and Lothar Thiele. 2012. Participatory air pollution monitoring using smartphones. *Mob. Sens.* 1 (2012), 1–5.
[27] Devroop Kar, Asif Iqbal Middya, and Sarbani Roy. 2019. An approach to detect travel patterns using smartphone sensing. In *Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS'19)*. IEEE, 1–6.
[28] Qihang Gu, Dimitris Sacharidis, Michael Mathioudakis, and Gang Wang. 2017. Inferring venue visits from GPS trajectories. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.
[29] Abhishek Gupta, Abhinav Khare, Haiming Jin, Adel Sadek, Lu Su, and Chunming Qiao. 2020. Estimation of road transverse slope using crowd-sourced data from smartphones. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 48–57.
[30] Sophia Karagiorgou, Dieter Pfoser, and Dimitrios Skoutas. 2017. A layered approach for more robust generation of road network maps from vehicle tracking data. *ACM Trans. Spatial Algor. Syst.* 3, 1 (2017), 1–21.
[31] Eric K. Tokuda, Yitzchak Lockerman, Gabriel B. A. Ferreira, Ethan Sorrelgreen, David Boyle, Roberto M. Cesar-Jr, and Claudio T. Silva. 2020. A new approach for pedestrian density estimation using moving sensors and computer vision. *ACM Trans. Spatial Algor. Syst.* 6, 4 (2020), 1–20.
[32] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby et al. 2008. Activity sensing in the wild: A field trial of UbiFit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1797–1806.
[33] Hancheng Cao, Jagan Sankaranarayanan, Jie Feng, Yong Li, and Hanan Samet. 2019. Understanding metropolitan crowd mobility via mobile cellular accessing data. *ACM Trans. Spatial Algor. Syst.* 5, 2 (2019), 1–18.
[34] Rohit Verma, Surjya Ghosh, Niloy Ganguly, Bivas Mitra, and Sandip Chakraborty. 2017. Smart-phone based spatio-temporal sensing for annotated transit map generation. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.
[35] Samuli Hemminki, Keisuke Kuribayashi, Shin'ichi Konomi, Petteri Nurmi, and Sasu Tarkoma. 2019. Crowd replication: Sensing-assisted quantification of human behavior in public spaces. *ACM Trans. Spatial Algor. Syst.* 5, 3 (2019), 1–34.
[36] Dominik Allemann and Martin Raubal. 2013. Towards health-optimal routing in urban areas. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on the Use of GIS in Public Health*. 56–59.
[37] Martin Becker, Saverio Caminiti, Donato Fiorella, Louise Francis, Pietro Gravino, Mordechai Muki Haklay, Andreas Hotho, Vittorio Loreto, Juergen Mueller, Ferdinando Ricchiuti et al. 2013. Awareness and learning in participatory noise sensing. *PLoS One* 8, 12 (2013), e81638.
[38] Rajib Rana, Chun Tung Chou, Nirupama Bulusu, Salil Kanhere, and Wen Hu. 2015. Ear-phone: A context-aware noise mapping using smart phones. *Pervas. Mob. Comput.* 17 (2015), 1–22.
[39] Joy Dutta, Pradip Pramanick, and Sarbani Roy. 2017. NoiseSense: Crowdsourced context aware sensing for real time noise pollution monitoring of the city. In *Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS'17)*. IEEE, 1–6.
[40] Harvey J. Miller. 2004. Tobler's first law and spatial analysis. *Ann. Assoc. Amer. Geog.* 94, 2 (2004), 284–289.
[41] Nina Siu-Ngan Lam. 1983. Spatial interpolation methods: A review. *Amer. Cartog.* 10, 2 (1983), 129–150.

[42] Edward H. Isaaks and R. Mohan Srivastava. 1989. *An Introduction to Applied Geostatistics*. Technical Report. Oxford University Press.
[43] Pierre Aumond, Arnaud Can, Vivien Mallet, Bert De Coensel, Carlos Ribeiro, Dick Botteldooren, and Catherine Lavandier. 2018. Kriging-based spatial interpolation from measurements for sound level mapping in urban areas. *J. Acoustic. Soc. Amer.* 143, 5 (2018), 2847–2857.
[44] Umang Bhatt, Shouvik Mani, Edgar Xi, and J. Zico Kolter. 2017. Intelligent pothole detection and road condition assessment. *arXiv preprint arXiv:1710.02595* (2017).
[45] Naufil Hassan, Ifrah Siddiqui, Suleman Mazhar, and Hadia Hameed. 2019. Road anomaly classification for low-cost road maintenance and route quality maps. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'19)*. IEEE, 645–650.
[46] Joy Dutta, Chandreyee Chowdhury, Sarbani Roy, Asif Iqbal Middya, and Firoj Gazi. 2017. Towards smart city: Sensing air quality in city based on opportunistic crowd-sensing. In *Proceedings of the 18th International Conference on Distributed Computing and Networking*. 1–6.
[47] Willian Zamora, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. 2017. Accurate ambient noise assessment using smartphones. *Sensors* 17, 4 (2017), 917.
[48] Gonçalo Marques and Rui Pitarma. 2019. Noise mapping through mobile crowdsourcing for enhanced living environments. In *Proceedings of the International Conference on Computational Science*. Springer, 670–679.
[49] Margaret A. Oliver and Richard Webster. 1990. Kriging: A method of interpolation for geographical information systems. *Int. J. Geog. Inf. Syst.* 4, 3 (1990), 313–332.
[50] Jin Li and Andrew D. Heap. 2011. A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors. *Ecol. Inf.* 6, 3-4 (2011), 228–241.
[51] J. Li and Andrew D. Heap. 2008. *A Review of Spatial Interpolation Methods for Environmental Scientists*. Geoscience Australia Canberra. Retrieved from http://www.ga.gov.au/servlet/BigObjFileManager?bigobjid=GA12526.
[52] Hans Wackernagel. 1996. Multivariate geostatistics: An introduction with applications. In *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, Vol. 33. Springer, 363A–363A.
[53] Peter M. Atkinson and Christopher D. Lloyd. 2014. Geostatistical models and spatial interpolation. In *Handbook of Regional Science*. Springer, 1461–1476.
[54] P. Goovaerts. 1997. *Geostatistics for Natural Resources Evaluation*.Oxford University Press, New York.
[55] Castle Group. 2019. Equivalent Continuous Sound Pressure Level. Retrieved from https://www.castlegroup.co.uk/guidance/noise-at-work-assessments/what-is-leq/. (2019).
[56] John W. Hager, Larry L. Fry, Sandra S. Jacks, and David R. Hill. 1992. *Datums, Ellipsoids, Grids, and Grid Reference Systems*. Technical Report. Defense Mapping Agency Hydrographic/Topographic Center Washington DC.
[57] SAI Global. 2020. Acoustics-Description and measurement of environmental noise. Retrieved from https://shop.standards.ie/store/PreviewDoc.aspx?saleItemID=211887. (2020).
[58] John W. Hager, James F. Behensky, and Brad W. Drew. 1989. *The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS). Edition 1*. Technical Report. Defense Mapping Agency Hydrographic/Topographic Center Washington DC.
[59] Patrick Reed, Barbara Minsker, and Albert J. Valocchi. 2000. Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation. *Water Resour. Res.* 36, 12 (2000), 3731–3741.
[60] Payam Refaeilzadeh, Lei Tang, and Huan Liu. 2009. Cross-validation. *Encyc. Datab. Syst.* 5 (2009), 532–538.
[61] QGIS. 2019. A Free and Open Source Geographic Information System. Retrieved from https://qgis.org/en/site/. (2019).
[62] Amprobe. 2018. Amprobe SM-10 Sound Level Meter. Retrieved from http://www.amprobe.com/amprobe/. (2018).
[63] Mextech Technologies. 2018. Mextech SL-4012 sound level meter. Retrieved from https://www.mextech.co.in/sound-level-meter.html. (2018).
[64] Gunnar Cerwén, Jacob Kreutzfeldt, and Carola Wingren. 2017. Soundscape actions: A tool for noise treatment based on three workshops in landscape architecture. *Frontiers of Architectural Research* 6, 4 (2017), 504–518.

Received July 2020; revised March 2021; accepted March 2021

Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-024-01681-3

Truthful double auction based incentive mechanism for participatory sensing systems

Asif Iqbal Middya[1] · **Sarbani Roy**[1]

Received: 22 November 2023 / Accepted: 28 February 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
The sensors available in the smartphones are useful to explore a diverse range of city dynamics (e.g. noise pollution, road condition, traffic condition, etc.). The potential of the smartphone sensors coupled with their widespread availability help to emerge a new paradigm of sensing known as participatory sensing. It uses the power of smartphone equipped sensors to collect, store, and analyze data with high spatiotemporal granularity. In a participatory sensing based system, a task provider (also known as a crowdsourcer) may have a set of sensing tasks regarding different dynamics of a city. Here, adequate users' participation is necessary to acquire a sufficient amount of data which is a key factor for the participatory sensing based systems to provide good service quality. The task providers appoint a set of task executors (smartphone users i.e. participants of crowdsensing tasks) to execute those sensing tasks. But, existing works on sensing task allocation suffer from lack of good incentive mechanisms that are attractive for the task executors. In order to address this issue, in this paper, a double auction based incentive mechanism called TATA (**T**ruthful Double **A**uction for **T**ask **A**llocation) is proposed for participatory sensing. TATA performs fair allocation of tasks which is leading to efficient incentive mechanism. In the case of TATA, the fair allocation of sensing tasks of the task providers to the task executers indicates that the proposed double auction mechanism is able to satisfy the truthfulness property in order to resist market manipulation (i.e., untruthful bidding and asking). Specifically, TATA achieves all the desirable properties like individual rationality, truthfulness (i.e. incentive compatibility), budget balance, etc. TATA is also computationally efficient and yields high system efficiency. Additionally, the performance of the proposed incentive mechanism is evaluated and compared with the existing mechanisms through extensive simulations based on the real-world data from Amazon Mechanical Turk. TATA yields high utility and satisfaction for the task providers and executors as compared to the existing mechanisms.

Keywords Participatory sensing · Double auction · Task allocation · Incentive mechanism

1 Introduction

Smart handheld devices (e.g. smartphone, tablet, etc.) are rapidly becoming an integral part of people's daily life. Importantly, today's smart handheld devices are equipped with useful sensors like accelerometer, GPS, gyroscope, microphone, camera, etc. These embedded sensors open up the possibility of monitoring a diverse set of users' activities and ambient environment. Now, participatory sensing (also known as crowdsensing) [1, 2] is a new paradigm of sensing where participants use their smart handheld devices to collect high granular data that can be used to explore different dynamics of a city (e.g. road condition monitoring, pollution monitoring, etc.). The active participation of a large number of smartphone users is extremely important for participatory sensing based systems to acquire sufficient data. But, in participatory sensing tasks, smartphone users not only incur costs due to sensing and uploading data but also suffer from potential privacy issues due to revealing sensitive information (e.g. GPS location of the users). Here, both the privacy and the cost issues greatly demotivate the users to actively participate in the sensing tasks. In this context, a satisfying

This article belongs to the Topical Collection: *1- Track on Networking and Applications*
Guest Editor: Vojislav B. Misic

✉ Sarbani Roy
sarbani.roy@jadavpuruniversity.in

Asif Iqbal Middya
asifim.rs@jadavpuruniversity.in

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

Published online: 20 April 2024

Technological Forecasting & Social Change 183 (2022) 121911



Contents lists available at ScienceDirect

Technological Forecasting & Social Change

journal homepage: www.elsevier.com/locate/techfore





Spatio-temporal variation of Covid-19 health outcomes in India using deep learning based models

Asif Iqbal Middya, Sarbani Roy *

Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

ARTICLE INFO

Keywords:
Covid-19
Deep learning
Spatio-temporal variation

ABSTRACT

Deep learning methods have become the state of the art for spatio-temporal predictive analysis in a wide range of fields, including environmental management, public health, urban planning, pollution monitoring, and so on. Despite the fact that a variety of powerful deep learning-based models can address various problem-specific issues in different research domain, it has been found that no single optimal model can outperform everywhere. Now, in the last two years, various deep learning-based studies have provided a variety of best-performing techniques for predicting COVID-19 health outcomes. In this context, this study attempts to perform a case study that investigates the spatio-temporal variation in the performance of deep-learning-based methods for predicting COVID-19 health outcomes in India. Various widely applied deep learning models namely CNN (convolutional neural network), RNN (recurrent neural network), Vanilla LSTM (long short-term memory), LSTM Autoencoder, and Bidirectional LSTM are considered to investigate their spatio-temporal performance variation. The effectiveness of the models is assessed using various metrics based on COVID-19 mortality time-series from 36 states and union territories of India.

1. Introduction

Spatio-temporal data analysis is becoming increasingly important due to the increasing availability of large scale data in various fields of research. Deep learning models have recently been used in spatio-temporal predictive analysis in a variety of applications, including pollution monitoring (e.g., particulate matter prediction), climate monitoring (e.g., precipitation forecasting), public health monitoring (e.g., COVID-19 health outcome prediction), and so on. However, due to the spatio-temporal variations in the data of target variables, the models' performance varies greatly both spatially and temporally. In the aforementioned context, the primary objective of this research is to investigate spatio-temporal variation in the performance of deep learning-based methods for estimating COVID-19 health outcomes in India.

At the beginning of 2020, the world health organization announced the COVID-19 outbreak a public health emergency of global concern. So far, COVID-19 has been associated with over 256 million confirmed cases and about 5.2 million fatalities globally. Many countries seek to protect their citizens' health by imposing mitigation strategies including social distancing, quarantines, travel restrictions, hard and soft locks, as well as postponing and cancellations of activities. In this context, several deep learning-based models for predicting COVID-19 health outcomes have been introduced in the existing works to aid in healthcare resources management and mitigation planning. Specifically, the accurate prediction could be helpful to decide the goals, guiding principles, and strategies to mitigate future consequences due to COVID-19.

Various deep learning based previous studies (Mohimont et al., 2021; Chimmula and Zhang, 2020; Zeroual et al., 2020; Said et al., 2021; Shastri et al., 2020) from around the world have yielded various top-performing methods for forecasting COVID-19 health outcomes. It is observed that there is no single best deep learning based forecasting model that can outperform everywhere. This is because there is a wide variation that exist in the COVID-19 health outcome time-series. It is worth noting that the variation in COVID-19 health outcome time-series depends on various risk factors that vary greatly across the geographical space. Moreover, several literature (Middya and Roy, 2021; Sannigrahi et al., 2020) highlights that there exists a geographically varying nature of the association between COVID-19 health outcomes and the complex risk factors. Hence, it is expected that there should have some variation in the models' performance. In the above-mentioned context, this study attempts to explore the spatio-temporal variation in the effectiveness of the deep learning based predictive methods for predicting covid-related health outcomes.

It is worth noting that the deep learning techniques can learn and extract features from raw and imperfect data on their own. Specifically,

* Corresponding author.
E-mail address: sarbani.roy@jadavpuruniversity.in (S. Roy).

https://doi.org/10.1016/j.techfore.2022.121911
Received 12 January 2022; Received in revised form 28 June 2022; Accepted 21 July 2022
Available online 2 August 2022
0040-1625/© 2022 Elsevier Inc. All rights reserved.

scientific reports

Check for updates

OPEN

Geographically varying relationships of COVID-19 mortality with different factors in India

Asif Iqbal Middya & Sarbani Roy ✉

COVID-19 is a global crisis where India is going to be one of the most heavily affected countries. The variability in the distribution of COVID-19-related health outcomes might be related to many underlying variables, including demographic, socioeconomic, or environmental pollution related factors. The global and local models can be utilized to explore such relations. In this study, ordinary least square (global) and geographically weighted regression (local) methods are employed to explore the geographical relationships between COVID-19 deaths and different driving factors. It is also investigated whether geographical heterogeneity exists in the relationships. More specifically, in this paper, the geographical pattern of COVID-19 deaths and its relationships with different potential driving factors in India are investigated and analysed. Here, better knowledge and insights into geographical targeting of intervention against the COVID-19 pandemic can be generated by investigating the heterogeneity of spatial relationships. The results show that the local method (geographically weighted regression) generates better performance ($R^2 = 0.97$) with smaller Akaike Information Criterion (AICc = −66.42) as compared to the global method (ordinary least square). The GWR method also comes up with lower spatial autocorrelation (Moran's $I = -0.0395$ and $p < 0.01$) in the residuals. It is found that more than 86% of local R^2 values are larger than 0.60 and almost 68% of R^2 values are within the range 0.80–0.97. Moreover, some interesting local variations in the relationships are also found.

The novel coronavirus disease (COVID-19) has spread rapidly to all parts of the world, causing almost 2.5 million deaths as of mid-February 2021[1]. Because of its unpredictable nature and lack of appropriate medications, COVID-19 is now a global health concern. There is unprecedented urgency to investigate the major factors that are related to COVID-19 death. In this context, recent studies are focusing on exploring person-specific risk factors for COVID-19-related health outcomes[2–4]. Also, there are research works that examine the association of COVID-19-related health outcomes with different socio-economic, environmental, and region-specific factors[5–7]. These factors play a very important role in determining the patterns of COVID-19 mortality.

Both global and local models can be utilized to explore the above-mentioned associations. A global model comes up with a geographically constant relationship across the entire geographic space. On the other hand, a local model can capture the local relationships that can vary across the geographic space. Most of the studies that focus on exploring the relationship of COVID-19 cases with different possible risk factors are based on global models (e.g. Ordinary Least Square)[8,9]. But, the global models assume that the associations between the independent variables and the dependent variable are stationary (i.e. homogeneous) throughout the study area. Besides, these models also assume that there is no spatial autocorrelation in the dataset. Eventually, they yield estimates of the parameters that reflect average behaviour[10]. But, in reality, the relationships between the dependent and the independent variables may not be homogeneous and can be geographically varying[11]. Therefore, such models usually suffer from low accuracy especially in those locations where weak association exists between dependent and independent variables. Now, various local techniques can be utilized in order to overcome the above-mentioned shortcomings of the global models. Some widely encountered local spatial statistics include geographically weighted regression (GWR)[12,13], local Moran's I[14], spatial regressions, etc.

As of 24 February 2021, India is the world's second worst-affected country by COVID-19, with a total number of deaths exceeding 156.7 thousand and a total number of confirmed cases exceeding 11 million[1]. However, in

Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India. ✉email: sarbani.roy@jadavpuruniversity.in

Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06319-6

ORIGINAL ARTICLE

Privacy protected user identification using deep learning for smartphone-based participatory sensing applications

Asif Iqbal Middya[1] · Sarbani Roy[1] · Saptarshi Mandal[2] · Rahul Talukdar[2]

Received: 13 April 2021 / Accepted: 8 July 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
In smartphone-based crowd/participatory sensing systems, it is necessary to identify the actual sensor data provider. In this context, this paper attempts to recognize the users' identity based on their gait patterns (i.e. unique walking patterns). More specifically, a deep convolution neural network (CNN) model is proposed for the user identification with accelerometer data generated from users smartphone sensors. The proposed model is evaluated based on the real-world benchmark dataset (accelerometer biometric competition data) having a total of 387 users accelerometer sensor readings (60 million data samples). The performance of the proposed CNN-based approach is also compared with five baseline methods namely Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbours (KNN). It is observed that the proposed model achieves better results (accuracy = 98.8%, precision = 0.94, recall = 0.97, and F1-score = 0.95) as compared to the baseline methods.

Keywords User identification · Gait · Participatory sensing · Convolutional neural network

1 Introduction

In smartphone-based crowdsensing [1] and participatory sensing [2] systems, participants share surrounding dynamics or personal information with others. More specifically, in these paradigms of sensing, participants use their smartphones to collect high granular data that can be used to explore different dynamics of a city (e.g. road condition monitoring [3], pollution monitoring [4–6], travel pattern monitoring [7], etc.). Now, as sharing smartphones with guest users (e.g. members of the family, colleagues, friends, etc.) is a common scenario, it is necessary for the crowdsourcing platforms to identify the actual data provider. It is because the raw participatory/crowd sensing data may suffer from inaccuracies due to the variety of expertise, reliability, and trust value of users. In this context, smartphone user identification is of utmost importance for personalized services and sensing data authentication in participatory sensing systems.

There exist explicit knowledge-based user identification mechanisms [8], such as personal identification numbers (PINs), passwords, or answering specific security questions. Also, there is some biometric-based authentication [9] such as fingerprint, face recognition, voice recognition, signature, and iris print. However, the participatory sensing-based applications sometimes running in the background for data collection from smartphone equipped sensors, in such context, may be annoying for the users to authenticate themselves at frequent time intervals. Moreover, several issues are associated with each of the above-mentioned authentication mechanisms. For instance, password- or PIN-based approaches require a lot of memorizing and remembering, or they can be lost or stolen away. According to a survey [10], an average of 21 passwords (a few of those up to 70) must be remembered by a high-end

✉ Sarbani Roy
sarbani.roy@jadavpuruniversity.in

Asif Iqbal Middya
asifim.rs@jadavpuruniversity.in

Saptarshi Mandal
m.saptarshi30@gmail.com

Rahul Talukdar
rtalukdar2000@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

[2] Department of Electrical Engineering, Jadavpur University, Kolkata, India

Published online: 21 July 2021

Mobile Networks and Applications
https://doi.org/10.1007/s11036-020-01539-x

JUSense: A Unified Framework for Participatory-based Urban Sensing System

Asif Iqbal Middya[1] · Sarbani Roy[1] · Joy Dutta[1] · Rituparna Das[1]

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
Participatory sensing has become an effective way of sensing urban dynamics due to the widespread availability of smartphones among citizens. Traditionally, separate urban sensing applications are designed to monitor different urban dynamics like environment, transportation, mobility, etc. However, combining these applications to aggregate information can lead to various new inferences. The main objective of this work is to improve urban sensing applications by overcoming their individual limitations. A unified framework called JUSense (Judicious Urban Sensing) is proposed that can derive benefits from these applications by combining their functionalities. JUSense provides the opportunity for applications to tackle the challenges associated with data collection, aggregation of data in cloud, calibration, data cleaning, and prediction. A multi-view fusion model is proposed for spatiotemporal urban air and noise pollution map generation. Further, a random forest classifier is built to classify the driving events. Here, large scale experiments are performed to evaluate the efficacy of JUSense on real-world dataset. Both the fusion model and the random forest classifier yield better accuracies compared to the baseline methods. Additionally, case studies are conducted to show the advantages that can arise out of the mutual interactions among the applications.

Keywords Urban sensing system · Participatory sensing · Smartphone · IoT · Cloud

1 Introduction

An urban sensing system can be considered to be a system that utilizes information and communication technologies (ICT) like sensors and actuators, wireless sensor networks (WSN), Internet of Things (IoT), etc. to enhance the quality and performance of urban services and utilities. The rapid growth of the worldwide urban population would bring to the cities to face some increased variety of risks, concerns, and problems. For example, degrading environmental conditions of air, water and noise pollution, deteriorating transportation system, poor road condition etc. In this context, an urban sensing system seeks to address these issues by using the data generated from different city dynamics. Here, city dynamics can be defined as those aspects of a city that directly influence citizens' quality of urban life like environment, transportation, mobility, driving and road condition, etc. Traditionally, it is usual to have independent urban sensing systems for different city dynamics. The study of urban dynamics at high granularity in a large area like a city has a significant role to correctly tackle the issues related to different dynamics. Thus, combining these independent urban sensing systems to aggregate information can help us to draw various new inferences. In this case, inferences can also be drawn with a higher accuracy that will give numerous benefits to society. Thus, an urban sensing system needs to be visualized as a collection of applications related to different city dynamics. For example, data from the environment through applications like air pollution monitoring and noise pollution monitoring can impact the transportation or mobility of people. More specifically, it integrates urban

✉ Sarbani Roy
sarbani.roy@jadavpuruniversity.in

Asif Iqbal Middya
asif.md.ju@gmail.com

Joy Dutta
joydutta.ju@gmail.com

Rituparna Das
ritu92.prf@gmail.com

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

Published online: 14 May 2020

Index No: 102/19/E

Response to the Reviewer's Comments on the Thesis titled "Predictive Modeling for Spatio-Temporal Analysis of Urban Sensing Data"

By

Asif Iqbal Middya

Comments/Queries:

In Chapter 2/3/4/5/6/7 Summary is 2.6/3.6/4.6/5.6/6.6/7.6 but Chapter 8 is having 8.6 as Discussion and Summary is not numbered. So, some uniformity may be kept. In core cases the Discussions

Some spellings and typos like "Inentive" for "Incentive" need to be corrected with through check with Grammarly or software aid of similar stature.

The problems faced in implementation of the work may be mentioned if the committee feels due to various ground problems in non-ideal cases.

The work in overall is very impressive and if you look into the volume of data collected and analyzed.

Response and Changes:

Thank you for your valuable comments and feedback on the PhD thesis. We have carefully considered your suggestions and made the necessary revisions to enhance the quality and clarity of the document.

The structure of Chapter 8 has been revised to ensure consistency with the rest of the thesis. A properly numbered "Summary" section, now labeled as 8.6, has been added following the "Discussion" section (8.5.3), aligning with the format used in Chapters 2 through 7.

Additionally, we have conducted a thorough review of the entire document for spelling errors, typographical mistakes, and inconsistencies. Specifically, the typo "Inentive" has been corrected to "Incentive,".

Once again, thank you for your insightful feedback, which has contributed to improving the overall presentation of the work.

Signature of the Research Scholar



Asif Iqbal Middya

Signature of the Supervisor



Dr. Sarbani Roy
Professor,
Dept of Computer Science and Engineering
Jadavpur University
(Supervisor)