Developing Techniques using Machine Learning for Dimensionality Reduction in WiFi-based Indoor Localization

Thesis Submitted

By

Ayan Kumar Panja

Doctor of Philosophy (Engineering)

Department of Computer Science and Engineering

Faculty Council of Engineering & Technology

Jadavpur University Kolkata, India

2024

Index No. 183/19/E

1. **Title of the Thesis:** Developing Techniques using Machine Learning for Dimensionality Reduction in WiFi-based Indoor Localization

2. Name, Designation and Institution of the Supervisor:

Dr. Sarmistha Neogy
Department of Computer Science and Engineering
Jadavpur University Kolkata,
West Bengal-700032, India

Dr. Chandreyee Chowdhury
Department of Computer Science and Engineering
Jadavpur University Kolkata,
West Bengal-700032, India

3. List of Publications

(a) Journal Publications:

Published

1. Parsuramka, S., **Panja, A.K**., Roy, P., Neogy, S. and Chowdhury, C., 2023. FABEL: feature association based ensemble learning for positioning in indoor environment. *Multimedia Tools and Applications*, *82*(5), pp.7247-7266. (**IF** : 3.6)
2. **Panja, A.K.**, Karim, S.F., Neogy, S. and Chowdhury, C., 2022. A novel feature based ensemble learning model for indoor localization of smartphone users. *Engineering Applications of Artificial Intelligence*, *107*, p.104538. (**IF** : 7.5)
3. **Panja, A.K.,** Rayala, A., Agarwala, A., Neogy, S. and Chowdhury, C., 2023. A hybrid tuple selection pipeline for smartphone based Human Activity Recognition. Expert Systems with Applications, 217, p.119536. **(IF :** 7.5**)**
4. **Panja, A.K**., Chowdhury, C. and Neogy, S., 2022. Survey on inertial sensor-based ILS for smartphone users. *CCF Transactions on Pervasive Computing and Interaction*, *4*(3), pp.319-337. **(IF:** 2.1**)**
5. **Panja, A.K**., Biswas, S., Neogy, S. and Chowdhury, C., 2024. Dimensionality Reduction through Multiple Convolutional Channels for RSS based Indoor Localization. *IEEE Sensors Journal*.(**IF:**4.325)
6. **Panja, A.K**., Karim, S.F., Neogy, S. and Chowdhury, C., 2024. Improving the sustainability of WiFi-enabled indoor localization systems through meta-heuristic based instance selection approach. *Expert Systems with Applications*, *257*, p.125063.(**IF:**7.5)

(b) Conference Publications:

1. **Panja, A.K**., Chowdhury, C. and Neogy, S., 2021. A Ubiquitous Indoor–Outdoor Detection and Localization Framework for Smartphone Users. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 1* (pp. 693-701). Springer Singapore.
   https://doi.org/10.1007/978-981-15-9927-9_67

2. **Panja, A.K**., Bhagat, D., Neogy, S. and Chowdhury, C., 2022, September. Framework for Remote Device Localization and Application Level Visualization for Emergency Service Providers. In *Adjunct Publication of the 24th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 1-4).

   https://doi.org/10.1145/3528575.3551445

3. **Panja, A.K.,** Chowdhury, C., Roy, P., Mallick, S., Mondal, S., Paul, S. and Neogy, S., 2021. Designing a framework for real-time wifi-based indoor positioning. In *Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020* (pp. 71-82). Springer Singapore.
   https://doi.org/10.1007/978-981-15-9433-5_8

4. **List of Patents :** None
5. **Copyright:** Optimized Smartphone Positioning using Feature-based Ensemble Learning (SW-18957/2024)

6. **List of Presentation in International Conference:**

   (a) **Panja, A.K.,** Chowdhury, C., Roy, P., Mallick, S., Mondal, S., Paul, S. and Neogy, S., 2021. Designing a framework for real-time wifi-based indoor positioning. In *Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020* (pp. 71-82). Springer Singapore. Venue: University of Engineering & Management Kolkata, India

   (b) **Panja, A.K**., Chowdhury, C. and Neogy, S., 2021. A Ubiquitous Indoor–Outdoor Detection and Localization Framework for Smartphone Users. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, Volume 1* (pp. 693-701). Springer Singapore. Venue: Institute of Engineering & Management Kolkata, India

Statement of Originality

I, **Mr. Ayan Kumar Panja** registered on **26th June, 2019** do hereby declare that this thesis entitled **"Developing Techniques using Machine Learning for Dimensionality Reduction in WiFi-based Indoor Localization"** 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 **3%** .

Signature of Candidate: Ayan kumar Panja

Date : 10.01.2024

Certified by Supervisor(s):

(Signature with date, seal)

1. Sarmistha Neogy. 10.01.2024

(Dr. Sarmistha Neogy,
Professor,
Department of Computer Science and Engineering
Jadavpur University.)

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

2. Chandreyee Chowdhury 10.01.2024

(Dr. Chandreyee Chowdhury,
Associate Professor,
Department of Computer Science and Engineering
Jadavpur University.)

ASSOCIATE PROFESSOR
Dept. of Computer Sc. & Engg.
JADAVPUR UNIVERSITY
Kolkata - 700 032

Certificate from the Supervisor

This is to certify that the thesis entitled ***"Developing Techniques using Machine Learning for Dimensionality Reduction in WiFi-based Indoor Localization"*** *submitted by* ***Shri. Ayan Kumar Panja****, who got his name registered on* ***26th June 2019****, for the award of Ph.D.(Engineering) degree of Jadavpur University, is absolutely based upon his own work under the supervision of* ***Dr. Sarmistha Neogy*** *and* ***Dr. Chandreyee Chowdhury*** *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.*

Sarmistha Neogy. 10.01.2024

Dr. Sarmistha Neogy,
Professor,
Department of Computer Science and Engineering
Jadavpur University.
(Supervisor)

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

Chandreyee Chowdhury 10.1.24

Dr. Chandreyee Chowdhury,
Associate Professor,
Department of Computer Science and Engineering
Jadavpur University.
(Supervisor)

ASSOCIATE PROFESSOR
Dept. of Computer Sc. & Engg.
JADAVPUR UNIVERSITY
Kolkata - 700 032

Acknowledgment

First and foremost, I extend my deepest gratitude and heartfelt thanks to my esteemed supervisors, Dr. Sarmistha Neogy, Professor at Jadavpur University, and Dr. Chandreyee Chowdhury, Associate Professor at Jadavpur University. Their unwavering guidance, invaluable insights, dedicated assistance, and continuous encouragement have been instrumental in steering and completing this research, contributing significantly to the fulfillment of my Doctor of Philosophy. Without their enthusiasm, profound perspectives on research, and unwavering support, this journey toward my PhD would not have been possible.

I am sincerely grateful to the other members of my thesis committee for their invaluable and constructive feedback, which has substantially enhanced the quality of this thesis. Furthermore, I want to express my gratitude to several remarkable individuals whose contributions significantly influenced this research. Firstly, I'd like to acknowledge Dr. Priya Roy, whose benchmark dataset formed the basis for our comparative analysis. I extend my thanks to Ms. Manjarini Mallik, Mr. Satyam Parsuramka, Mr. Adityar Rayala, Mr. Abhay Agarwala, and Mr. Sajan Rajak for their collaborative efforts. In the data collection domain, I'm thankful to Mr. Sakil Mallick, Mr. Sukanto Mondal, and Mr. Soumik Paul for their thorough site survey and fingerprint data collection, crucial elements supporting this thesis. Lastly, I wish to highlight Mr. Syed Fahim Karim's commendable contributions, significantly impacting this research.

I am deeply thankful for my wife, Chandrima's unwavering support. Her encouragement and backing in every aspect of my life have been priceless. I extend my heartfelt appreciation to my my mother, late father and mother-in-law for always believing in my pursuits. Furthermore, I am graced with twin daughters whose very existence is a remarkable blessing. Additionally, I express my gratitude to the nurturing work atmosphere at the Institute of Engineering & Management, which greatly facilitated my research endeavours. Immense thanks to my colleague, Mr. Amartya Mukherjee, whose guidance ignited my research passion and unwavering support has been a constant motivation for me. Lastly, I extend my sincere thanks to the Department of Computer Science and Engineering at Jadavpur University and Head Dr. Nandini Mukherjee for granting me this exceptional opportunity and providing steadfast support and guidance throughout my research expedition.

Dedicated

To my cherished Wife, Chandrima, my revered Parents, and beloved Twin Daughters.

To all the researchers who have contributed in this field

Abstract

In the past decade, WiFi-based Indoor Localization has emerged as a pivotal domain, offering transformative possibilities for location-based services in indoor environments. Most indoor public places are covered by WiFi today. So, reusing the distance sensitivity property of such signals is a sensible approach for localization where GPS signals are not available.

This research explores the requirement of dimensionality reduction within the context of WiFi-based Indoor Localization Systems (ILS). The objective of the work is to propose robust localization approach utilizing the existing Wi-Fi infrastructure. This involves refining the prediction accuracy by gearing towards distilling relevant information while mitigating the influence of noise and irrelevant features.

Dimensionality reduction encompasses both feature and instance space. In the context of feature space both Wrapper and Filter-based AP selections have been investigated. One of the core contribution of this thesis is that it combines meta-heuristic based feature selection with the notion of ensemble learning for addressing the dynamic contexts of the localization phase. The conditional ensemble has been constituted with the feature subsets obtained from the meta-heuristic algorithm-based feature selection approaches. This enhances the system's ability to adapt to diverse contexts, including device heterogeneity and environmental fluctuations. Both Genetic Algorithm and Binary Particle Swarm Optimization have been considered for the purpose. Further innovations unfold in the domain of dataset distillation, involved delving into Convolutional Autoencoding (CAE) principles and k-disagreeing neighbor scores. To tackle the effect of outliers that get induced during data collection, meta-heuristic based instance selection approaches have been proposed. This also ensures effective localization performance in constrained environments.

Experimentation has been carried out on collected dataset through multiple smartphone devices. Results shows that the feature-based pipeline achieved over 95% accuracy, reducing access points (APs) by 50-65%. Error deviation of 2.68m is achieved, which is acceptable for indoor user localization. The instance-based pipeline cuts dataset size by 40%, with a slight 2-3% accuracy dip but significant reduction in error deviation. The designed machine learning approaches are also found to be significantly applicable to other smartphone signal modalities and application domains such as, Human Activity Recognition, illuminating the transformative capacity of refined instance selection techniques. Edge level training analysis has also been employed to validate the performance of the model in constrained environment.

Nomeclature

e_i: Indoor Localization System
$R_{M,N}$: Radiomap Dataset; where M is the no. of tuples and N is the no. of APs
P: Total number of chromosomes
C_i: Chromosome object i
r: Fraction of population selected for crossover
m: Number of population selected for mutation
k: Number of times GA is iterated
k': Number of Feature subset selected from k feature set
$Fset_i$: Selected Feature subset from i-th executed GA process
bl_i: Base Learned i
$rssi_{uN}$: Received signal strength indicator from Nth AP
$L_f - x - y$: Virtual grid coordinate(x,y) for f'th floor $P_i.feature$: AP selection by particle i of the swarm
$P_i.vel$: Velocity vector of particle i of the swarm
$P_i.Pbest$: Particles i best selection
$P_i.fitness$: Particle i current fitness
$Gbest$: Global best fitness
GBF: Global best selection vector
$pSize$: Swarm size
$bcount$: Base Learner count
c_1: Cognitive acceleration component
c_2: Social acceleration component
$avgF$: Swarms average fitness score
$maxIter$:Maximum iteration
ϵ: Threshold for convergence
MT: Meta-model Classifier
F_i: Fingerprint i
R': Standard scaled radiomap dataset
kDN_i: k-disagreeing score of ith tuple
B_enc: Encoded Bottleneck
D: Decoded output of CAE
θ_i: ith parameter of the encoder-decoder function
α: Learning rate
FM_t: Feature map of the t-th element of the output tensor
I_i: Instance i
$q_{I_r,k}$: k-Nearest neighbor query for Instance I_r
$D'_{m',n}$: Dataset after first phase pipeline of GA process
Ch_i: Chromosome object i
$maxGen$: Maximum no. of generations
$genC$: Current generation
Y_v: Virtual grid label of vth fingerprint instance
S_i: Particle object vector for instance selection
$S_i.f$: ith particle's selected instances
$S_i.v$: ith particle's velocity vector

$percR$: Percentage reduction of fingerprint instances
ec: Extra cost imposed on difference of accuracy score metric
α_i: Classification accuracy of fold i
δ: Acceptable mean distance threshold percentage
β: Percentile mean distance
W: Weight parameter of BPSO velocity update
$score$: k-disagreeing score
$orgScore$: Original score of accuracy on whole dataset

Acronym

ILS: Indoor Localization System
IPS: Indoor Positioning System
AP: Access Point
RSS: Received Signal Strength
RSSI: Received Signal Strength Indicator
CONN: Condensed Nearest Neighbor
GA: Genetic Algorithm
PSO: Particle Swarm Optimization
FABEL: Feature Association Based Ensemble Learning
MAE: Mean Absolute Error
PCA: Principle Component Analysis
SVD: Singular Value Decomposition
SMOTE: Synthetic Minority Oversampling
RBF: Radial Bias Function
CNN: Convolutional Neural Network
SVM: Support Vector Machines
RFC: Random Forest Classifier
RF: Random Forest
kNN: k-Nearest Neighbor
CAE: Convolutional Autoencoding
KDN: k-disagreeing score
ReLU: Rectified Linear Unit
SGD: Stochastic Gradient Descent
AECCNN: Autoencoded Compounded Convolutional Neural Network
CCNN: Compounded Convolutional Neural Network
DNN: Deep Neural Network
FHM: Fingerprint Hardness Score
CDF: Cumulative Distribution Function

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Positioning or localization is one of the most researched domains in the recent era. The Global Positioning System (GPS) works by using signals from at least four satellites, which a GPS receiver uses to calculate its distance from each satellite, enabling it to determine its precise location coordinates on Earth. In case of indoor environment it has been observed from experimentation that the GPS signal strength decreases by about 10-12 decibels as the device enters a semi-indoor environment after which the signal strength decreases further for deep-indoor environment [1]. Scattering and multi-path effect are two of the most common phenomena for which GPS is not used for Indoor Localization Systems(ILS). Hence, various localization models are proposed by researchers to achieve the positioning of devices in an indoor environment. Sensors are practically used everywhere; a smartphone consists of numerous sensing sub-devices. It is crucial to develop a modelling approach for the positioning and location identification of such devices to serve various application-specific needs. WiFi RSS-based Indoor Localization Systems (ILS) capitalize on the widespread availability of WiFi infrastructure in various indoor settings, thus eliminating the need for any additional expensive or invasive hardware installations. At the heart of this process lies the fundamental principle of the signal propagation model. RSS values inherently carry information about the spatial context, given their correlation to the distance from the WiFi Access Points (APs) [2]. A WiFi-enabled device can measure RSS values from multiple APs and, through trilateration or multilateration methods, ascertain its relative position within a defined space.

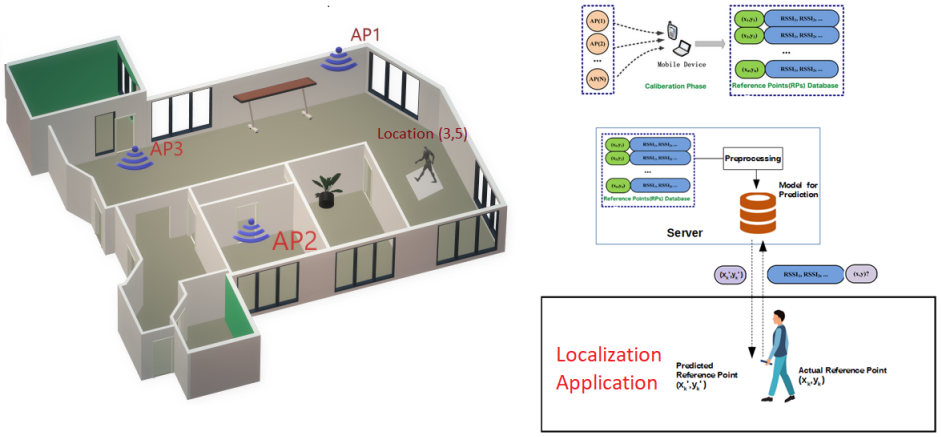


Figure 1.1: Wi-Fi based Indoor Positioning Framework

The realization of WiFi RSS-based indoor localization is not without its complexities. Indoor environments present a uniquely challenging landscape due to multipath propagation, signal fading, and interference caused by walls, furniture, and even human occupants. These factors can introduce significant inconsistencies and fluctuations in the RSS values, thus impeding the accuracy and reliability of indoor localization.

1.1 Major Application Areas of Indoor Localization

The technologies primarily used in the development of indoor localization are WiFi-based, Bluetooth-based, Vision-based, Lo-Ra techniques, and so on. Indoor Localization Systems (ILS) have a multitude of application areas. They are used to pinpoint the location of objects or people inside buildings using radio waves, magnetic fields, acoustic signals, or other sensory information collected by mobile devices [3]. The major domains are healthcare, localization and navigation systems(Indoor), asset tracking, and robot navigation which are also depicted in Figure 1.2. Here are several important application areas in detail:

- **Commercial and retail:** ILS can enhance customer experiences by providing personalized deals when customers are near certain products or helping them navigate large stores. Proximity-based advertising [4] [5] and personalized services are gaining

popularity in the indoor domain. In [6], a work is reported that specifically focuses on Wi-Fi based framework for intelligent shopping experience in a mall. The work integrates the notion of Augmented Reality experience to enhance shopping experiences in malls.

On the operations side, ILS can help with asset tracking and inventory management. Asset management [7] and tracking [8], particularly in warehouses, manufacturing facilities, and big commercial spaces is an essential application domain of ILS. ILS are critical in controlling and tracking assets, equipment, and inventory within constrained indoor environments, thereby optimizing operations, and improving operational efficiency. ILS can optimize warehouse management by tracking inventory in real time, managing space more efficiently, and assisting in route planning for picking up and delivering items.

- **User localization and navigation:** Researchers used various ways for user navigation and tracking based on the environment and ambience. User localization provides a variety of services, ranging from regular way finding at shopping malls, hospitals, railway stations, airports and other public areas to emergency evacuation. ILS enhances visitor experience in public venues like convention centers, stadiums, and malls, providing location-based information and guiding visitors to their desired destinations.

  ILS in healthcare area [9][10] is used to find elderly and differently abled people and assist them in navigating unfamiliar situations. Furthermore, by continuous monitoring, falls and other minor mishaps can be identified and immediate actions can be taken thereof. Indoor localization improves patient care [11] by enabling location-based services for staff, patients, and medical equipment, leading to efficient work-flow and reduced waiting times.

- **Safety and emergency response:** During emergencies, ILS can help responders navigate complex buildings quickly. It can also assist in locating individuals who

may need help. User localization in emergency scenarios in an indoor environment, and even daily commuters require navigation help. As a result, crowd formation is natural. Understanding crowd trajectory and anticipating mobility patterns is a key application domain of the ILS system.

- **Robot Navigation:** Indoor mobile robot navigation [12] is critical in personal assistant robots, self-driving cars, industrial applications such as automated manufacturing, huge aerospace structure assembly, and many more. Indoor mobile robot localization and navigation refers to the robot's capacity to detect its own position and orientation within its frame of reference and then search for an appropriate path to its destination location. Robot navigation involves using sensors, such as cameras or LiDar, along with mapping and localization algorithms to enable autonomous movement and navigation in indoor environments. Employing technologies such as Wi-Fi, UWB, and sensors to enable robots to navigate autonomously forms the base upon which the systems are built. Robot Navigation finds application in various other domains, including warehouse logistics, healthcare, and smart homes, enhancing efficiency, safety, and convenience.

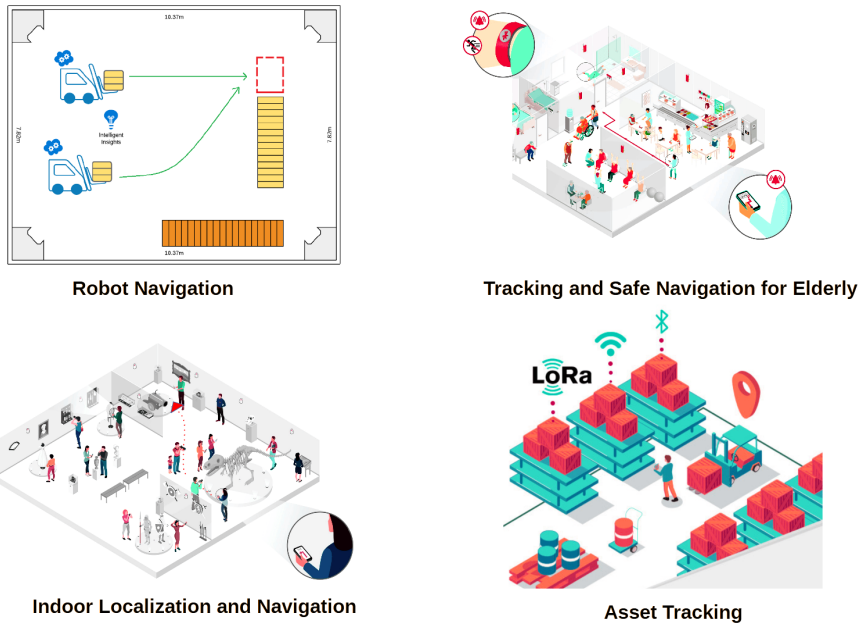


Figure 1.2: Major Application domain of Indoor Positioning Systems(IPS)

1.2 Evolution of Wi-Fi based Indoor Positioning

In this subsection the evolution of Wi-Fi based indoor positioning has been discussed since its inception. In the year 2000, researchers from Microsoft proposed RADAR [13] the first major development of Wi-Fi based ILS. The system utilized signal strength information from multiple Wi-Fi access points to estimate the user's location. The framework combined measurements with signal propagation modelling to locate the users on a particular floorplan. Youssef and Agrawala proposed Horus [14] in year 2005. The Horus system is a software solution built on top of the WLAN. It achieves high accuracy by identifying and addressing the causes of wireless channel variations. Specifically, the Horus system for a target signal strength vector $rss{=}rss_1, .., rss_L$ from L access points tries to find the target location x with the maximum posterior probability. The approach used location-clustering techniques to reduce the computational requirements of the localization process.

Distance-based approaches such as, trilateration [15, 16] involve determining the position of a device by measuring the distances to three or more known Wi-Fi APs. Majority of the past researches is focused on improving the accuracy of distance estimation and localization algorithms based on trilateration. Other popular range based approaches such as WiFi Angle-of-Arrival(AoA) [17], Time-of-Flight (ToF), had also emerged as potential solutions for more precise indoor positioning. Array Track [18]- one of the earliest approaches, utilized a large array of physical MIMO antennas to measure the angle of arrival of WiFi signals, enabling more precise location estimation. RSS fingerprinting based approach gained popularity from 2013 onwards. The major focus of the approach lies in constructing a radiomap database with respect to a registered set of Wi-Fi APs for a particular floor, based upon which localization approaches are built. One of the earliest approach proposed by Microsoft Research Asia is the EZ Localization [19]. It was designed with the objective of eliminating the need for site surveying, which was a significant drawback of earlier systems like RADAR and Horus. EZ implemented a semi-supervised learning model that utilized the correlation between signal strengths of different WiFi access points. Hybrid approaches combining Wi-Fi with other sensing modalities has also gained popularity. Data fusion integrates information from multiple sensing modalities.

Sensors are calibrated and selected to align and enhance measurements. Sensor fusion algorithms combine data, considering context, for accurate positioning. Indoor Atlas [1] a Finnish startup, presented a unique approach that used the variations in the Earth's magnetic field inside buildings (caused by structural elements like steel beams) and combining it with Wi-Fi RSS for localization.

Machine learning techniques gained traction in indoor positioning research majorly from 2016. Researchers started exploring different machine learning algorithms, such as k-nearest neighbors (KNN) [20] [21], Naive Bayes [22], Support Vector Machines (SVM) [23] [24], Decision Tree [25], Random Forests [26], and Artificial Neural networks [27], to improve localization accuracy. Machine learning models require extensive data collection and site survey. Numerous work on advanced machine learning has been carried out to reduce the site survey. Modelling signal propagation for signal prediction at various interior locations is a promising strategy to cut the cost of the survey. In order to estimate the target position and learn the signal propagation parameters, WiGEM [28] uses the Gaussian mixture model (GMM) and expectation maximisation. Following that, it will be possible to forecast the signal strength at various locations. As a result, given implicit user signal data, the survey process is significantly shortened. It does, however, require specific knowledge of AP sites.

Deep learning approaches can handle complex relationships. One of the first deep learning based approaches DeepFi [29] was proposed which employed deep learning techniques to model the relationship between Wi-Fi signal strength and location. The deep learning architecture was modelled on Channel State Information(CSI). CSI is a measure of how a signal propagates from a transmitter to a receiver in a wireless communication system. These algorithms were used to classify and predict the position of a device based on WiFi signal characteristics. WiDeep [30] is another approach proposed in 2017 that combined deep learning with crowdsourcing to eliminate the need for well annotated data collection. The system used smartphone-based WiFi signal strengths and the pedestrian

[1] https://www.indooratlas.com/

dead reckoning (PDR) system to provide better indoor localization. Apple[2] introduced indoor mapping for selected locations in iOS 11, which uses hybrid technologies such as WiFi, GPS, and Bluetooth for indoor positioning in the year 2018.

Transfer learning based approaches to reduce site survey have also been reported. Transfer learning is used to utilize the knowledge gained previously from the first training phase. MoLoc [31] is popular work that used on-device transfer learning to adapt to new environments quickly and efficiently, reducing the need for extensive site surveys.

1.3 Machine Learning framework for ILS

The objective of the supervised learning framework designed for indoor localization systems is to establish a mapping between received Wi-Fi signals and their respective positions within an indoor setting. It is possible because of the distance sensitivity of the Wi-Fi signals. This comprehensive framework encompasses multiple distinct phases like data collection, data preprocessing, feature extraction, model training, and finally, prediction.

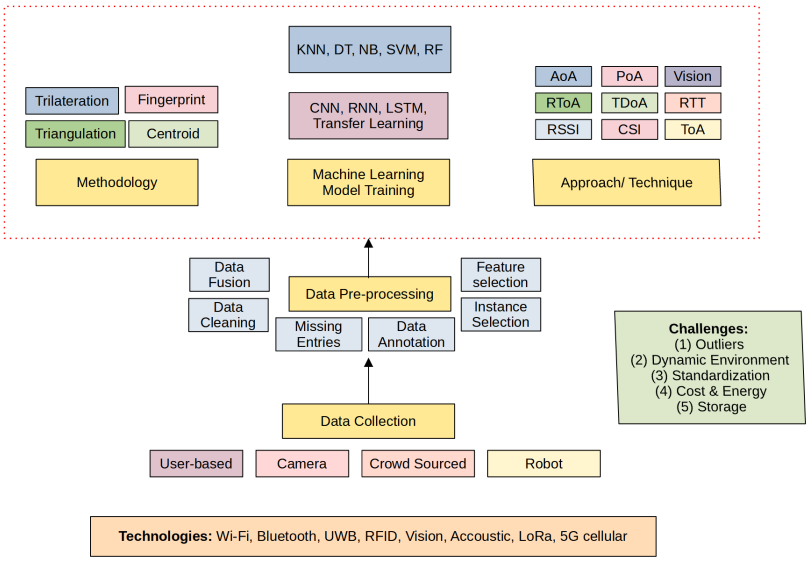


Figure 1.3: Overview of Indoor Positioning System and the associated technologies used in building machine learning frameworks

The data collection is carried out as offline phase where the received signal strength from Wi-Fi APS are recorded. The fingerprint database is created by collecting the RSS

[2]https://register.apple.com/indoor

values from the registered set of APs. A reference point label is allocated with each tuple of the gathered RSS. Fingerprints are collected for a stipulated number of days and in different periods of time in order to capture the environmental context. The data collection phase is carried out using manual site survey/crowdsourcing. A snapshot of published radiomap dataset known as JUIndoorLoc is showcased in figure 1.4. The respective RSS from the APs are the features while the $Cell - id$ is the respective grid label from where the fingerprint is collected. The positioning is estimated by capturing RSS fingerprint from

AP001	AP002	AP003	.....	AP171	AP172	Cell-id
-76	-76	-34	.....	-110	-110	L4-26-8
-76	-74	-34	.....	-110	-110	L4-26-8
-65	-72	-42	.....	-110	-110	L4-22-1

Figure 1.4: Snapshot of JUIndoorLoc radiomap dataset

an unknown location whose label is predicted using a classification model residing in the server. Machine learning algorithms help in capturing complex patterns, environmental variability and extracting features. Supervised Models are trained on radiomap datasets with distinct reference grid labels. Some of the popularly used supervised approaches in context of RSS based Indoor localization:

- **k-nearest neighbor:**The KNN model determines class membership based on the proximity or distance metric between the samples in the class. The distance is calculated using the attribute values or RSSI values assigned to the APs. The hyper-parameter k is adjusted and modified as needed.

- **Decision Tree:** It is a tree-based learning method in which decisions about the feature metric are made against each of the samples in the tree nodes. The chosen set of APs is part of the nodes where decisions are made, and the tree's leaf nodes are the grid labels of the floormap.

- **Support Vector Machine:** In a high-dimensional feature space, SVM identifies the optimal hyperplane that optimally divides various classes. The hyperplane is the best margin between the labels. SVM can be used in Indoor Localization to classify a user's location based on sensor data into distinct rooms or zones, achieving good separation even in complicated indoor environments.

- **Naive Bayes:** Naive Bayes is a probabilistic classifier based on the Bayes theorem with the feature independence assumption. Naive Bayes can be used to estimate the probability of a user's location given particular sensor signals, making it computationally efficient for indoor localization tasks.

- **Ensemble Models:** Ensemble models combine many base classifiers to improve prediction accuracy by decreasing overfitting and increasing generalisation. Indoor localization accuracy can be improved by capturing complex correlations in sensor data and enhancing robustness against noise using ensemble models such as Random Forest or Gradient Boosting.

- **Neural Network:** Neural networks simulate the connections of neurons in our brain. The feature vectors in RSS-based indoor localization are the RSS from the APs. Grids are categorical data that are label encoded and provide the output layer in the grid-based positioning domain. The activation function, and hence the hidden layer's processing, is fed by a weighted sum of each input tuple corresponding to the APs. The forward pass for evaluation and the backward pass for weight correction form the foundation of the ANN's learning process.

- **Deep Learning:** Given the complexity and non-linearity of indoor localization problems, deep learning methods are also popularly used to capture these complex relationships. Deep learning models, such as Deep Artificial Neural Network(DNN), Convolutional Neural Networks (CNNs) or hybrid architectures, can be trained using the preprocessed and feature-extracted WiFi data. Deep learning models can be utilized to learn complex mappings between sensor features and indoor locations, enabling accurate localization in intricate indoor environments.

Unsupervised approaches like clustering (e.g., K-means, DBSCAN) are also popularly used to group together observations (e.g., WiFi signal patterns) that are similar to each other. This can help in identifying regions or zones in an indoor environment. Hybrid approaches such as semi-supervised learning involves clustering the region of interest into clusters and running supervised models within the clusters to locate the position.

1.4 Motivation

ILS systems based on Wi-Fi have several benefits, but comes with its share of challenges. Indoor Wi-Fi signals are highly variable due to multi-path propagation, interference, environmental changes(like moving objects) and Important AP down. No single AP can impart stable RSSI across all areas and conditions. Due to the presence of different environmental context, systems that perform well in one environment may not necessarily do so in another. This variability makes it challenging to model the relationship between signal strength and location accurately, impacting the precision and consistency of these systems. A solution to learn robust and invariant features or access points is needed to handle the different RSS fluctuations on environmental changes.

To guarantee real-time responsiveness, it is crucial for the localization system to operate effectively within constrained environments, with a specific focus on the environmental context. This raises the focus on two research challenges. The first being that the features used in the training process should cover a number of distinct multiple environmental scenarios to make the positioning system more accurate and robust. Therefore, it is necessary to design a system that can cover multiple ambient contexts, providing accurate and reliable localization across different conditions. Secondly, data storage becomes a limiting factor for constrained environment. That is, it becomes expensive to retain the data variability that captures the majority of the environmental context. Furthermore it is essential to store important data that preserves the class boundaries while avoiding outliers.

The above two research challenges necessitates the investigation towards research into dimensionality reduction for ILS. Consequently the above two research challenges are mapped to the below research questions on dimensionality reduction.

1. How to design a feature selection technique that ensures stable performances across various ambient conditions?

2. How to mitigate the influence of noise and ensure localization in constrained envi-

ronments with limited storage?

1.5 Contribution

The above mentioned research questions motivated us to explore dimensionality reduction techniques in the context of radiomap datasets. In the present thesis work, dimensionality reduction is investigated from two perspectives - feature selection and instance selection. Following are the contributions which are part of this thesis.

- **Design of Feature based ensemble learning model that captures different ambient condition:** In order to capture the environmental contexts with respect to any particular floor plan, it is important to retain the important set of APs that defines the floormap. A metaheuristic based AP selection for better exploration and selection has been proposed here. The Binary particle Swarm Optimization(BPSO) and Genetic Algorithm(GA) have been explored for feature selection problem. In the majority of the floor plans selected for positioning, some APs are found to impart similar information to the localization process, that is they are highly correlated w.r.t a given training context. Thus, there can be more than one global optimal selection of APs for that context. With different ambient conditions and devices, the context varies during testing. Considering the variations in ambient conditions and devices during testing, it becomes imperative to ensure generality regardless of the specific context. Hence the need for developing a feature-based ensemble approach is important. The feature-based ensemble classifier is proposed based on the important feature subsets obtained by the use of the meta-heuristic approach to perform positioning.

- **Latent space representation of important features through autoencoding while incorporating instance hardness:** Exploring feature extraction as an avenue for dimensionality reduction constitutes a significant aspect of our research motivation. The latent space representation is a lower-dimensional vector space where each dimension corresponds to a learned feature or characteristic of the input data. We have proposed a 2-channel input representation and a learning pipeline

by utilizing Convolutional autoencoding(CAE) to capture the underlying structure or patterns in the radiomap dataset in its latent space representation. The training pipeline is built in a way where the internal structure is able to learn the inherent instance importance and assess the difficulty or uncertainty associated with classifying that instance. This is done through the representation of k-disagreeing score, which is one of the measure of instance hardness as part of the input channel.

- **Combining nearest neighbor concept with meta-heuristic algorithm to address the problem of instance selection** To address the impact of outliers and enhance training at the edge level, this research delves deeper into instance hardness and meta-heuristic algorithms within the domain of instance selection. A crucial aspect of the thesis is estimating the instance hardness measure by assessing the nearest neighbors of each instance and calculating the k-disagreeing score. The thesis explores both a two-phased and a single-phased approach to instance selection based on meta-heuristic methods. Additionally, the use of nearest neighbor estimation as a standalone reduction phase, as well as its incorporation into the meta-heuristic learning process are investigated. These investigations contribute significantly to the overall findings of the thesis.

1.6 Organization of the Thesis

The thesis makes original contribution in the domain of dimensionality reduction in context of Wi-Fi based indoor localization. The thesis is organized in the following manner.

- *Chapter 2:*In this chapter, the discussion is on the requirement for dimensionality reduction in context of indoor localization problem is presented here. A survey through the past works carried out in the domain of AP selection and instance selection problem.

- *Chapter 3:* This chapter proposes a wrapper based AP selection approach which utilized genetic algorithm based feature selection. The analysis carried out proves the multi-modal presence important AP sets for the selected floor plan. A feature based majority voted ensemble learning for the training phase is proposed.

- *Chapter 4:* A binary particle swarm based AP selection strategy is proposed. The approach takes into consideration both the device and environmental context into count. A feature based stacking ensemble is proposed which is tested on the locally collected dataset. Rigorous experimentation and analysis on benchmark datasets has been carried out.

- *Chapter 5:*In this chapter, we have proposed a convolutional autoencoder-based dimensionality reduction approach. This works is on the basis of feature transformation and instance hardness. The encoding process of the data input involves a two-channel representation of a fingerprint dataset that holds the normalized RSS and an instance hardness measure, that is, a k-disagreeing score. The inclusion of the k-disagreeing score into the training pipeline is made with the objective of injecting instance importance for training using 1D CNN architectures for classification.

- *Chapter 6:* This chapter proposes two approaches on instances selection problem for indoor localization datasets. Both the approaches involve an evolutionary part that tries to explore different selection of instances in search of the optimal selection of tuples. The first approach is a two phased instance selection approach that combines a clustering based outlier de-selection followed by a GA based instance selection. The second approach explores optimal selection of APs with a modified BPSO based approach. Both the approaches have been rigorously experimented on various benchmarks dataset.

- *Chapter 7:* This is the final chapter which concludes the thesis. In this chapter the contribution are summarized and consolidated findings are presented. Scope for further improvement of the present work is also presented.

Chapter 2

Survey of Dimensionality Reduction for Wi-Fi RSS Datasets

Wi-Fi RSS datasets consist of a large number of measurements collected from multiple APs at different locations within a given indoor area. While these datasets hold valuable information for localization, they often suffer from high dimensionality, which can lead to computational inefficiencies and poor performance of localization algorithms. RSS dataset does not pose many dimensions such as as image/video feature set. So, curse of dimensionality in its traditional meaning is not applicable here. The objective of dimensionality reduction in this context is to reduce the number of features or instances in the dataset, while preserving the essential information necessary for accuracy positioning.

Reducing dimensionality in RSS fingerprint based ILS is needed because of the following reasons:

- The distribution of the APs play an important role in localization. Even distribution of WiFi access points ensures uniform signal strength and coverage throughout the indoor space. So, representation from all distinguishing experimental regions should be given equal importance.

- To retain generality, important APs should be maintained by building authorities for sustainable ILS. Hence the requirement of Feature selection is quite important.

- In ILS dataset a unique characteristic can be observed, the number of features is quite high with respect to the number samples per class/grid labels. Furthermore, the data collection process is carried out through manual site survey. Hence, the presence of class imbalance and outliers are inevitable. Furthermore to train the model in constrained edge environment with limited computing resource, it is important to train the model that captures the varying contextual context. Hence, the application of dimensionality reduction both with respect to feature and instance spaces is imminent.

Throughout this chapter, the study of different dimensionality reduction techniques for Wi-Fi RSS datasets is presented. Analysis on the strengths and limitations of feature selection (Section 2.2) and instance selection Section 2.3) approaches have been done, and to provide insights into their impact on the performance of indoor positioning systems. Additionally, some empirical results on state-of-the-art feature extraction and instance selection approaches are presented. The goal of this chapter is to present a comprehensive overview of dimensionality reduction techniques for Wi-Fi RSS datasets and their implications for indoor positioning.

2.1 Overview of Dimensionality reduction

Dimensionality reduction is a process that simplifies the dataset by reducing the number of random variables under consideration. The goal is to transform high-dimensional data into a lower-dimensional space without losing or significantly distorting the original information. Dimensionality reduction techniques generally fall under two broad categories: feature selection and instance selection. Feature selection or AP selection refers to the process of choosing a subset of relevant features (APs, in this context) for use in model construction. This selection may be based on the intrinsic properties of the features themselves or their relevance and contribution to a specific outcome. On the other hand, instance selection targets the reduction of data size by selecting a representative subset of instances (i.e., measurements or samples) from the original dataset. These two approaches are not mutually exclusive and can be employed in combination to achieve

effective dimensionality reduction.

2.2 Feature Selection or AP Selection Techniques

Feature selection, also known as variable selection or attribute selection, is one of the core concepts in machine learning which tremendously influences the performance of your model. The main idea behind feature selection is to choose a subset of input variables by eliminating features with little or no predictive information. By doing this, the complexity of the model is reduced, potentially improving its performance and interpretability.

2.2.1 Techniques of Feature Selection

There are two broad categories of feature selection methods: filter methods and wrapper methods. Filter methods are generally used as a preprocessing step and involve statistical methods for ranking and selecting features based on their relationship with the target variable. Wrapper methods involve selection techniques that are usually computationally expensive and involve evaluating subsets of variables which best improve the performance of a chosen predictor.

Filter methods are the most straightforward category of feature selection methods. They evaluate the relevance of the features by their inherent properties. For instance, correlation or mutual information between the features and the target variable can be used as a measure of relevance. The key advantage of filter methods is their computational efficiency. Since these methods operate independently of any learning algorithm, they are particularly suitable for preprocessing steps when dealing with high-dimensional datasets. However, they tend to overlook the potential interactions between features, which can be a disadvantage in some cases.

Contrary to filter methods, wrapper methods take into account the performance of a predetermined learning algorithm to evaluate the usefulness of features. They search for the best feature subset by training a model on different feature subsets and selecting the one that maximizes the model's performance. Although wrapper methods can often

provide better performance than filter methods, they are computationally more intensive, especially when dealing with high-dimensional data. They also carry a higher risk of overfitting due to their tendency to tailor the feature set to a specific classification model.

2.2.2 Application of Feature Selection on Wi-Fi RSS Datasets

In ILS, not all APs necessarily provide information that is relevant for positioning. This is where feature selection or AP selection comes into play. Numerous studies have been done where two individual APs when considered with different subset produced better accuracy than when considered together. Laitinen et al. in [32] demonstrated in their work that weighted centroid[33] based approaches are highly sensitive to AP selection and they have also showcased that 50% of the APs can be reduced with properly chosen selection criterion. Generally, the access points for a particular floor are selected based on criteria, such as signal strength, information gain, fingerprint clustering, etc.

AP selection techniques aim to identify the most relevant APs, i.e., those whose signal strengths provide the most useful information for positioning.

We have listed some of the notable works carried out for the AP selection procedure. One of the very first signal strength-based approaches in AP selection was given by Youssef et al. in [34] utilizing joint clustering approach that utilizes maximum likelihood estimation based on the RSS values received from the APs. Jiang et al. in [35] proposed a selection of APs based on the signal levels; the claimed accuracy was appreciable, which worked perfectly for a big room. In [36], the authors have utilized the clustering and decision tree-based approach in the selection of the APs. The authors proposed an offline AP selection and have studied the effect of selection on four criteria: *MaxMean, Infogain, RndMean, and ReverseInfogain. Infogain* is the selection criterion performed in descending order of APs with respect to the information gain from the APs, *ReverseInfogain* is the reverse of *Infogain* where the APs are selected in the reverse manner in which Infogain is used for selection. APs are ranked in descending order of their average RSS following the MaxMean criterion. *RndMean* criterion is based on random AP selection regardless

of the RSS values from the APs. The authors have claimed to have achieved appreciable accuracy with *Infogain* criteria for a hall by taking a subset of APs.

Clustering-based approaches are applied to divide the regions into clusters and select the best set of APs for the particular cluster. A region-based clustering and AP selection approach based on *Infogain* correction is proposed in [37]. Hybrid Positioning combining one methodology with another is also applied in feature extraction procedure [38]. The authors have developed a Fisher score-stacked sparse autoencoder to extract features followed by a hybrid positioning. The hybrid positioning works on the selected set of APs by dividing the region into sub-regions with the help of Fuzzy C-Means Clustering. For their selected floor map a total of 181 APs were present. On the application of the procedure, they were able to reduce the APs to 119. The mean error deviation was found to be 2.09m for their selected environment containing a corridor and stairway. Chen et al. in [36] proposed a power-efficient AP selection approach, performing experimentation separately using some selection criteria such as MAXMEAN, information gain, reverse information gain, and RndMean. They have used a hybrid clustering and decision tree-based approach with which they have claimed that for probabilistic location estimation information gain-based AP selection approach gave a better result. Meng et al. in [39] proposed selection of APs based on signal distortion. The authors have proposed a cluster-wise AP selection method to increase the robustness of the changes in the environment. Huang et al. in [37] proposed a hybrid clustering and classification based approach for the feature selection problem. The selection criteria are based on information gain. The authors have used k-means clustering to cluster the reference locations and have used the decision tree-based classification to select APs for the particular cluster.

Luo et al. in [40] proposed a Principal Component Analysis(PCA) based AP selection approach and Affinity Propagation Clustering is utilized to constrict the positioning range. PCA based approach to reduce the high correlation between grid points is also found in [23]. The authors have claimed to generate uncorrelated space using Eigen vectors and hence, the work proceeds with various machine learning models. In [41], PCA[42] is used

to reduce the high dimensionality of the dataset followed by parameter tuned Support Vector Machine(SVM)[43] approach. The approach has been compared and tested with feature selection based on Principal Component Analysis (PCA) [23] and autoencoders [44]. The approach gives better results both in terms of accuracy and error deviation. However, the approach is not yet suitable for real-time positioning, as claimed by the author. In [45], AP selection was made considering the correlation of APs. The correlation is calculated by estimating the AP's divergence measure. In [46], the authors have shown the importance of detecting stable APs across different granularity levels.

Region-based AP selection involves segmenting the floor plan into multiple regions of interest. Cluster wise AP selection has also received the attention of the researchers. A group discriminant-based AP selection approach is given in [47], where the authors have considered the positioning capabilities of the APs in the group. The authors have claimed to have utilized the risk function from SVM for estimating a discriminant value of the Group and selecting the best set of APs that can contribute to the positioning.

Another notable work by Cheng et al. in [48] reported the use of Ensemble support vector regression for the approach of AP selection and reconstruction of RSSI of the non selected APs. The selection approach is MAX-MEAN and they have claimed that their model is less sensitive to noise due to the signal reconstruction procedure.

Deep Learning approaches are becoming popular in the domain of positioning. Deep learning methods are expected to learn a model of the RSS based positioning relationship by properly encoding complex environment factors into its model parameters. Feature extraction is an important aspect in deep learning techniques. Convolutional neural networks(CNN) [49] are used in extracting the hierarchical representation of input data. Autoencoder is another form of learning through latent space representation. Many of the past researches have explored the use of Autoencoders in feature extraction process. Xing et al. in [50] has proposed an approach on stacked denoise autoencoding on Hyperspectral images [51].

Table 2.1: Comparative study on AP selection approaches

Year	Type	ML Algorithm	Methodology	Positioning Metric	Discussion
2006 [36]	Filter-based	k-means clustering	Access Point selection is carried out using information theory based criteria- InfoGain, MaxMean, RndMean, and ReverseInfoGain.	Average cluster accuracy 82%	The performance of the algorithm in changing environments is not clear. It is uncertain how well it would adapt to changes in the physical layout or conditions that affect signal propagation. The proposed method relies heavily on the availability and the distribution of Access Points (APs). In areas where APs are sparse or inconsistently available, this method might not be as effective.
2010 [45]	Wrapper-based	k-nearest neighbor	Selection of APs is carried out in a manner to selection APs with lower correlation	Error deviation below 5m.	Not suitable for dynamic environment, such as changes in the physical layout, signal obstruction, or interference that affects WLAN signal strength.
2011 [39]	Filter-based	k-nearest neighbor	Region based reference point selection and probabilistic removal of APs.	2.5-5m error deviation.	(a) Sensitive to dynamic environment as it assumes that the nearest Reference Points(RPs) are the most relevant for localization. (b) Less scalable
2013 [47]	Wrapper-based	Support Vector Machine	Risk function from the SVM to estimate the group discriminant value by maximizing the margin between reference locations. Recursive Feature Elimination (RFE), combined with GD approach (RFE-GD) for AP selection.	4-5 m deviation.	(a) The fast version of the method, RFE-GD, is stated to find a suboptimal solution which makes a trade-off between speed and optimal AP group selection. (b) Not suitable for dynamic environment.
2015 [35]	Filter-based	k-nearest neighbor, Bayesian Probabilistic Model	APs with strongest RSS is selected	Accuracy of 85.90%	Device Heterogeneity can reduce the performance.
2016 [32]	Filter-based	Weighted centroid based positioning	AP selection criteria- Information Gain, Maximum RSS, FFT & KL Divergence.	Error Deviation 5-11 m	Approach is less scalable
2016 [48]	Wrapper-based	Ensemble Support Vector Regression (SVR) and Artificial Neural Networks (ANN)	MaxMean criteria for AP selection. Reconstruction of the received signal strength indicator (RSSI) values of non-selected APs from those of the selected APs.	Error Deviation 1-2.5m	The testing and validation of the method are described as being performed through simulations. Real-world performance, scalability, and the applicability in various types of buildings and environments may differ from simulated results.
2017 [40]	Filter-based Approach	Affinity Propagation Clustering	Precise positioning using maximum likelihood	95% accuracy, error deviation 2-4 m	Suitable for only initial simple positioning.
2019 [46]	Filter-based Approach	IBK, Lib SVM, Bayes Net, K*	Selection of APs carried out using information gain. Considering standard deviation and mean	Accuracy 96.62%	(a) Temporal and Environmental Variability affects the positioning accuracy (b) Less scalable.
2019 [38]	Wrapper Based Approach	FCM Clustering	Zero-mean normalization for AP selection, Fisher–SSAE is applied to extract important features to train the machine learning models.	More than 95% accuracy. Localization error 2.76m	Multipath path propagation and not suitable for dynamic environment.
2020 [52]	Wrapper Based Approach	C4.5 Decision Tree	Information gain based approach	Localization accuracy 92% , 1.6-2 m error deviation	Not suitable for dynamic environment.

The high level and sparse level features have been extracted through ReLU activation

in the Autoencoding process. The classification process has been carried out using Logistic regression. Meng et al. in [53] proposed a Autoencoder based feature extraction that considers the attributes as well as their relationship into consideration. Kunang et al. in [54] proposed the use of Autoencoding in Intrusion Detection System. The classification is carried out using SVM classifier and the approach is able to retain the relation of information that has been compressed.

A comprehensive study of the presented work on AP selection is given in table 2.1. From the table it can be observed that the majority of prior research on AP selection has focused on taking into account variables like Infogain, MAXMean approach, etc. The use of clustering-based procedures is also widespread; these methods include dividing the world into clusters or areas and then executing the selection process, which can take some time. The majority of earlier studies are best suited for static floorplans that are subject to relatively little environmental changes.

2.3 Instance selection approach in Dimensionality Reduction

While feature selection focuses on reducing the dimensionality of the dataset by selecting relevant features, instance selection tackles the problem from a different perspective. It is a process that aims to reduce the volume of data by selecting a representative subset of instances (i.e., measurements or samples) from the original dataset. The primary goal is to maintain the integrity and class distribution of the original data while minimizing redundancy and noise. This not only leads to more efficient computational processing but can also enhance the model's predictive performance by focusing on the most informative instances. This enables an ILS to be extended at the device or edge level that provides a constrained environment but ensures real-time responsiveness.

Instance selection techniques can be broadly divided into two categories [55]: Condensation and Edition. Condensation techniques aim to find a minimal representative subset

of the original data that can correctly classify new instances, preserving the boundaries between different classes. On the other hand, edition techniques aim to remove the noisy instances that are likely to be misclassified or those located in the boundaries between different classes, thereby smoothing the class regions for more accurate future classifications. Different algorithms have been proposed for instance selection, each with its strengths and weaknesses. Some well-known instance selection algorithms include the Condensed Nearest Neighbor (CONN) rule, the Reduced Nearest Neighbor (RENN) rule, and the NearMiss approach. Choosing the right algorithm requires careful consideration of the specific characteristics of the dataset and the requirements of the positioning system.

2.3.1 Past Researches on Instance Selection

In the past decade, numerous work have been carried out in the domain of Instance reduction. One of the prominent approaches is the Condensed Nearest Neighbor(CONN)[56] approach. The approach works by putting the object in a representative state that is close to the class boundaries. As the elements close to the center do not take part in defining the class boundary, they can be removed without perturbing the accuracy of the classifier. Another popular undersampling approach is the Near Miss Approach [57]. The approach works by removing samples from bigger classes when two instances in the distribution belonging to different classes are relatively close.

Filter-based approaches focus on selecting instances without focussing on model accuracy as the base criterion. Cluster approaches are popular filter-based approaches. Ougiaroglou et al. in [58] recursively apply the k-means algorithm to form homogeneous clusters. The proposed selection can achieve more than 70% reduction in the sample count with acceptable accuracy reduction. Chen et al. in [59] proposed another k-means-based clustering approach which is used in multiclass instance selection. The authors have presented numerous results carried out on multiple datasets. The reported results indicate that the approach achieves a reduction of 30-40% with 4-5% decrease in the accuracy metric. The proposed approach also works better with the binary class dataset. Another approach based on retaining class border points is proposed in [60]. In the work, a rank is allocated

to each instance where the border points are given the highest rank. A selection is carried out from good, best, and worst ranked instances. The selected instances are tested using Support Vector Machine [61], Locally weighted Learning [62] and C4.5 [63] classifier and have been reported to achieve appreciable accuracy. Paper [64] presents a Local density-based instance selection that tries to retain the densest instance concerning the class label.

Hybrid approaches involve the combination of more than two procedures. Ryu et al. in [65] proposed an instance selection approach for software defect prediction dataset. The approach combines k-nearest neighbor for local analysis and the Naive Bayes approach for global analysis. Data mining-based approaches are also very popular in extracting essential information from a very large dataset. Carbonera et al. in [66] have proposed an approach that uses the concept of local density to identify the most representative instances of each dataset class. The work has been tested with 20 well-known datasets and the authors have claimed to have achieved an appreciable trade-off between accuracy and instance reduction. The proposed approach has been compared with the Local Density-based Instance Selection(LDIS) algorithm [64], and the results show that LDIS requires a very large no of instance counts for good performance.

Outlier detection is a part of instance selection that has been investigated in a variety of research and application fields. Outliers are samples that deviate from the rest of the samples in the same class and are the major cause of model performance degradation. Song et al. in [67] proposed an approach that exploits the locality of k-neighbors approach for outlier removal. In [68] a clustering-based outlier removal is explored that utilizes density peak reachability and Chebyshev's inequality. The results show that the approach is efficient in identifying both local and global outliers. Ensemble-based approaches [69, 70] are also explored for outlier detection and removal. Ensemble approaches work by merging the findings of disparate models to create more robust models that can effectively detect outliers.

For Wi-Fi RSS datasets, instance selection can play a pivotal role in enhancing the

effectiveness of the Indoor Positioning System (IPS). By choosing the most representative instances, we can ensure that the IPS is trained with high-quality data that reflects the spatial variability of the Wi-Fi signals. Moreover, reducing the amount of data can significantly reduce the computational requirements of the positioning process, making the IPS more scalable and efficient.

2.3.2 Effect of Dimensionality Reduction on Localization Algorithms

The impact of dimensionality reduction on localization algorithms is multifaceted, influencing computational efficiency, robustness, to accuracy. High-dimensional datasets can place substantial computational burdens on localization algorithms, slowing down processing time and demanding more resources. Dimensionality reduction can mitigate these issues by reducing the number of features or instances that the algorithm needs to process. This leads to faster computations, making the Indoor Positioning System(IPS) more scalable and efficient. Besides computational efficiency, dimensionality reduction can improve the robustness of localization algorithms. Overfitting, a common problem where a model learns the training data too closely and performs poorly on new data, is more likely to occur with high-dimensional data. By reducing dimensionality, the algorithm can be made to focus on the most meaningful features or instances, enabling it to learn a more generalized model and perform better on unseen data.

The effect of dimensionality reduction on localization accuracy, however, is more nuanced. An effective dimensionality reduction process can potentially increase positioning accuracy by removing noisy or irrelevant data that could distort the algorithm's output. Yet, there is also a risk of information loss which might degrade localization accuracy. This underscores the need to balance the reduction of dimensionality with the preservation of valuable information for localization.

It is also important to note that different localization algorithms may react differently to dimensionality reduction. For example, machine learning-based algorithms such as k-Nearest Neighbors (k-NN) or Support Vector Machines (SVM), which are sensitive to noisy

data and prone to overfitting, may see substantial benefits from dimensionality reduction. Understanding the interplay between specific localization algorithms and dimensionality reduction techniques is thus crucial.

2.4 Empirical Results

In this section some of the positioning results and the effect of dimensionality reduction approach on localization radiomap dataset is showcased. For experimentation JUIndoorLoc [71] and UJIIndoorLoc [72] dataset is considered. Both of the datasets are benchmark radiomaps. The datasets are collected, pre-processed and involves real world variability and complexities hence enabling standardized evaluation. The UJIIndoorLoc covers the buildings of Universitat Jaume I. They have considered 3 buildings of the university with more than 4 floors. For the data collection phase, they have considered 25 devices for constructing the radio map database. The latitude and longitude value of every grid is mapped as the class label. The dataset has 520 total APs in their feature vector from which the fingerprints were recorded. The APs from which the RSS could not be measured were filled with a value of 100 dBm. For this experimentation, only building 1 was considered for evaluation. For the 2nd Floor of building 1 of UJIIndoorLoc Dataset is considered that has 9493 instances. The JUIndoorLoc dataset has been collected for a selected floors at Jadavpur University. The dataset consists of the RSSI values obtained from different APs present in the experimental region corresponding to different class labels (grid cells). Multiple devices (mobile phones or tablets) were used while collecting the data against different ambient conditions (open or closed door, number of people in the room) at different times of the day over a period of one month. However, during data collection, values from many APs were not available at some locations at certain times. These missing RSSI values were assigned the value of -110dBm. As such, the values of the RSSI in the dataset varied from -24dBm to -100dBm. The JUIndoorLoc [71] dataset has over 172 APs with 6639 instances. The classification algorithms used for modelling are-k-nearest neighbor, Decision Tree and SVM classifiers.

Table 2.2: Data description of number of access points and tuples pertaining in JUIndoorLoc and UJIIndoorLoc radiomap set

Dataset	No. of Aps	No. of instances
JUIndoorLoc	172	6639
UJIIndoorLoc	520	9493

2.4.1 Feature selection analysis

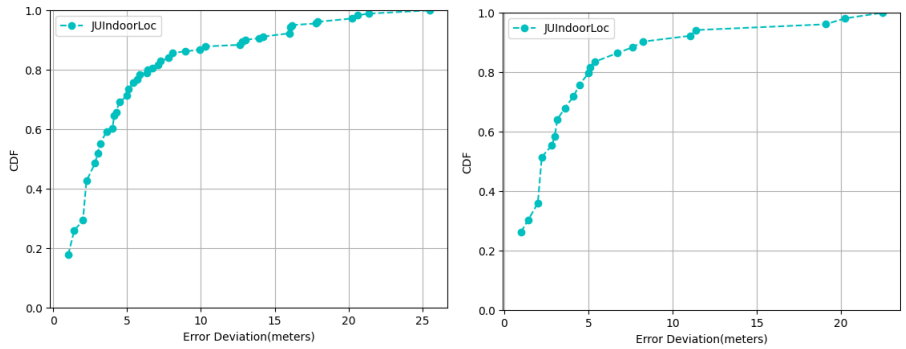
For feature selection and extraction the findings using Principal Component Analysis(PCA) [73] and select-k-best approach [74] is showcased. PCA falls under the filter-based approach because it analyzes the inherent structure and variance in the data to determine the most informative features. Select-k-Best is a feature selection technique that evaluates the statistical relationship between each feature and the target variable. It selects the top k features with the highest statistical scores, where the score represents the importance of the feature. For the presented analysis Chi-square statistics is used. The Chi-square statistics for each feature against the categorical target variable assigns a score based on the magnitude of the statistic. The higher the chi-square statistic, the stronger the association between the feature and the target variable. SelectKBest is a wrapper-based approach as it selects the best features by directly incorporating the performance of a specific machine learning algorithm.

The error deviation is another important metric which helps in testing the performance of the localization system. The error analysis is carried out by estimating the deviation of the predicted location point from the actual location point. This is done by estimating the Euclidean distance of the misclassified instances from the actual label. For both the considered dataset let the predicted and actual grid label be (x_i', y_i') and (x_i, y_i) respectively. The estimated error in meters is evaluated as follows (Equation 3.6,3.7);

$$e_i = \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \tag{2.1}$$

In Figure 2.1, showcases the Cumulative Distribution Function(CDF) of the error deviation after the application of both the approaches on JUIndoorLoc Dataset. The CDF gives the probability that the positioning error takes on a value less than or equal to i meters

deviation. It has been observed through experimentation that setting the component parameter of PCA to 40 and the k parameter of select-k-best to 80 yielded the best performance of the models. Among the selected classifiers -kNN, SVM and Decision Tree, the Decision Tree classifier is performing the best with accuracy ranging between 82-85%. The mean error deviation with PCA is $4.94m$ and with select-k-best is $4.17m$.

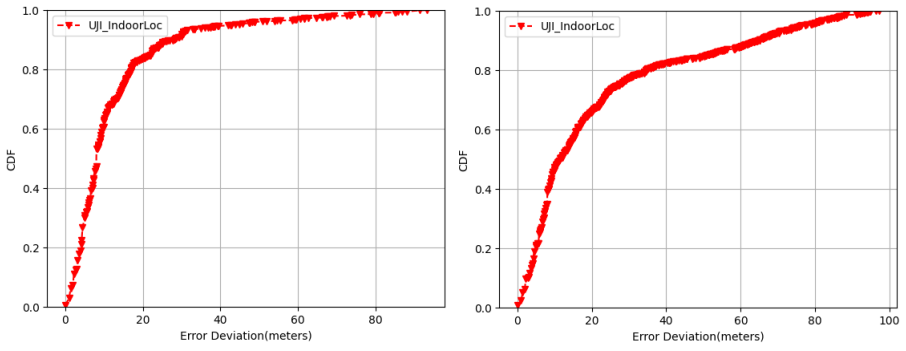


(a) PCA based feature selection approach (b) select-k-best feature selection selection approach

Figure 2.1: CDF analysis of AP selection approach carried out on JUIndoorLoc benchmark dataset.

2.4.2 Instance selection analysis

Similar experiments have been carried out on UJIIndoorLoc dataset whose error deviation is showcased in Figure 2.2. The component parameter of PCA has been set to 60 and the k parameter of select-k-best to 100. The error deviation after the application of PCA as feature extractor is $12.92m$ and with select-k-best $21m$. Since the ground truth error in this dataset is quite high, it is not possible to get fine grained results.



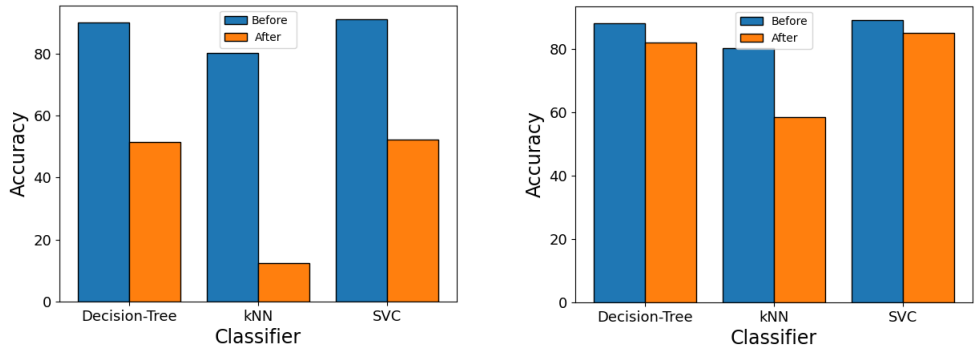
(a) PCA based feature selection approach (b) select-k-best feature selection approach

Figure 2.2: CDF analysis of AP selection approach carried out on UJI-IndoorLoc benchmark dataset.

Following experiments are conducted on application of instance selection approaches on the benchmark datasets. The following experiments reports the accuracy before and

after the application of instance selection approach using the selected set of classifiers. Condensed Nearest Neighbor and Near Miss instance selection approach for experimentation is used. Condensed Nearest Neighbor (CONN) is a classic instance selection algorithm that aims to reduce the size of a dataset while retaining its representativeness. Near Miss approach selectively removes majority class instances and addresses the class imbalance in the WiFi-based indoor localization dataset.

In Figure 2.3 presents the accuracy results on JUIndoorLoc dataset with both the approaches. It can be seen that Near Miss approach is performing better than CONN and is able to reduce the dataset by almost 48%. It can also be seen that the accuracy reduction after the application of the approach is between 10-18%.

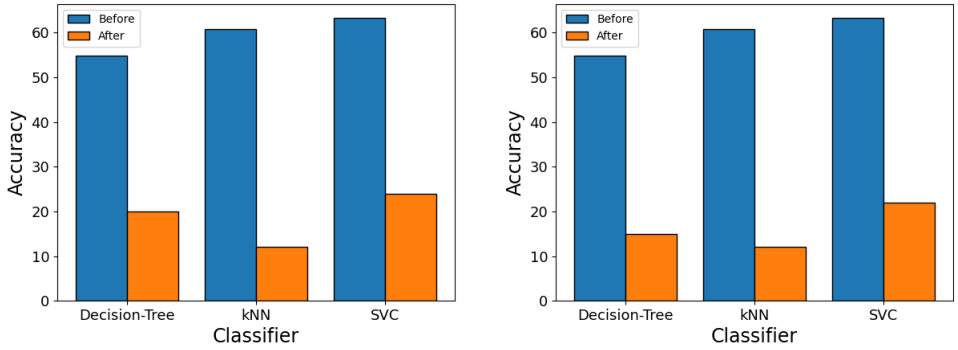


(a) Condensed Near Miss Approach (CONN) for Instance Selection.

(b) Near Miss Approach for Instance selection

Figure 2.3: Accuracy comparison performed using instance selection approach on JUIndoorLoc benchmark dataset

Figure 2.4 reports the accuracy metric before and after the application of selected instance selection approaches on the UJIIndoorLoc dataset. It can be observed that both the approaches has underperformed and the accuracy of the positioning model has decreased by almost more than 50%. Hence, the requirement of adaptive feature and instance based learning needs to be developed to cater to the problem of dimensionality reduction in the context of ILS.



(a) Condensed Near Miss Approach (CONN) for Instance Selection.

(b) Near Miss Approach for Instance selection

Figure 2.4: Accuracy comparison performed using instance selection approach on UJIndoorLoc benchmark dataset

2.5 Discussion

In this chapter discussion on some of the important aspects of dimensionality reduction in the context of Wi-Fi based Indoor Positioning has been carried out. Through the reported works it can be observed that majority of the past researches on AP selection are not able to capture device heterogeneity, dynamic environment and thus are in-turn less scalable. In the case of instance selection problem, it has been observed through past researches that none of the previous approaches has employed instances selection in the domain of Wi-Fi based indoor localization. Hence in the subsequent chapters, the effects of the proposed dimensionality reduction approach in the context of both feature selection/AP selection and instance selection on a Wi-Fi based indoor localization dataset are explored. The aim of the work is to gain insights into the impact of these techniques on the performance of the localization task by conducting extensive experiments and analysis.

Chapter 3

Wrapper based Feature Selection approach using GA

The Localization accuracy depends highly on environment characteristics. APs with strong signal strength provides better reliability in the positioning process while weak signals barely show distance sensitivity. The signal strength readings are also impacted by environmental factors including humidity, opening and closing of doors and windows, and the presence of nearby interfering devices. As a result, based on the changing surroundings, the signals from an AP change depending on the time of day. One access point, however, would not provide strong signal strength for all the smartphones across the experimental area. Thus, the selection of the combination of APs significantly affects the localization process. Hence, feature selection plays a vital role in retaining the critical set of attributes or access points that significantly contribute to localization and reduce the computation overhead. Feature selection is crucial in maintaining the crucial group of APs that must be appropriately maintained for consistent localization performance.

In the present chapter, a Feature Association Based Ensemble Learning (FABEL) model that combines Genetic Algorithm(GA) based feature selection with ensemble learning model is proposed. To develop solution for any dynamic floorplan which is prone to changes in the environment, the meta-heuristic approach could be a feasible solution. Genetic Algorithm takes into consideration a potentially large search space intending to

explore for an optimal solution. In Indoor Positioning field, the various literature surveys [75][76] reflects that GA has been extensively used for the placement problem of the Wi-Fi APs for Indoor Positioning applications. So, in this work a training pipeline incorporating GA as the part of the feature selection and ensemble learning in model training for stable localization performance subject to a range of contexts is proposed. Accordingly, our contributions are as follows.

- A GA-based feature selection technique is proposed for smartphone-based indoor localization.
- A feature set based weighted ensemble of classifiers is designed that can add better stability to the localization performance by not clinging onto any specific ambient condition.
- Implementation and thorough experimentation of the approaches for real life datasets.

Section 3.1 gives a preliminary overview of the Binary Genetic Algorithm. The AP selection approach is discussed in Section 3.2. The proposed Feature-Based Ensemble Learning approach is discussed in Section 3.2.2. In Section 3.3, the experimental setup and results are discussed.

3.1 Overview of Genetic Algorithm

Genetic Algorithm [77] is a type of optimization technique that is based on the principles of evolution observed in species. GA starts with a collection of chromosomes, also known as the initial population, and iteratively modifying them to improve their fitness for a particular problem. The chromosome encoding process involves mapping *genes* which are components of the chromosomes to the parameters of the selected problem. Some of the ways in which encoding is performed are Binary Encoding, Permutation Encoding, Real value encoding, Priority Encoding, and Tree encoding [78]. Each of the chromosomes are the candidate solution. The population of GA is manipulated with the three defined genetic operators- *selection, crossover* and *mutation*. There are many methods for choosing the initial population, but random selection is the method that is most frequently utilised.

A population-level centre of mass, chromosome-level hamming distance, and genetic level entropy are a few of the commonly utilised methods. For instance, in [79], the work has showcased an initial population for global optimization problem which was generated by using the upper and the lower bound of the variables instead of pseudo-random numbers. The fitness of the newly generated children is determined by an objective function, which evaluates how well the chromosome represents a potential solution to the problem at hand. The fitness value of a chromosome is an indicator of its suitability as a solution, with a higher fitness value indicating a better solution. To improve the overall quality of the population, selection strategies are employed to retain the best solutions (i.e., chromosomes with higher fitness values) and discard the weaker ones. Roulette wheel selection strategy [80] is a common and widely used selection technique. The *crossover* procedure is applied to generate a new generation of chromosomes from the existing chromosomes. It is mainly done after the *selection* procedure, swapping the content of two chromosomes and forming two new generations of the chromosome. During the *mutation* operation, a *gene* within a chromosome of a solution is randomly modified with a probability known as the mutation probability. The purpose of the mutation operator is to introduce new traits or characteristics that were not present in the original population, thereby increasing the genetic diversity of the population. The addition of new characteristics may lead to either selection or de-selection of the very chromosome in next generation. It often involves changing the gene values and is primarily done to prevent premature convergence of the GA to poor solutions by resuming lost exploration for a fresh solution. When using a binary genetic algorithm, the mutation is carried out by flipping the chromosome's 0/1 bit string. It is anticipated that after a specific number of iterations or generations, a better solution to the problem will be achieved with the estimation of fitness and implementation of the three procedures- *selection*, *crossover* and *mutation*.

3.2 Proposed Approach

The proposed approach is divided into two phases. First, GA is used to choose the AP sets, or characteristics, that will best support the location categorization process. To find distinct feature sets that produce similar localization accuracy, this GA-based feature

Table 3.1: Terminologies Used in the Proposed Work

Notation	Description
R	*Radiomap Database*
N	*Total No of Access Points; with respect to chromosome encoding it represents the number of genes*
P	*Total Chromosome Population size*
C_i	*Chromosome Object i*
n	*No of shuffled training set*
r	*Fraction of population selected for crossover.*
m	*Fraction of Population selected for mutation*
k	*No of times GA is iterated; yielding k feature set*
k'	*number of Feature subset selected from k feature set*
$Fset_i$	*Selected Feature subset on ith run of GA process*
Sm	*Similarity matrix*
bl_i	*Base Learner i*

selection process is repeatedly iterated with different population sizes. The unknown location of the test set is then determined utilizing a feature-based ensemble technique that makes use of these several feature sets. In this context, the approaches of feature-based ensemble learning and GA-based feature selection are covered in this section. An overview of the variables utilised in the procedure is provided in Table 3.1.

3.2.1 GA-based Feature Selection

The experimental area is divided into virtual grids of equal size. The radio map database, R, contains the Received Signal Strength Indication (RSSI) values of all the Access Points (APs) detected from each virtual grid. A row of the radio map R is represented as $< rssi_{u1}, rssi_{u2}, ...rssi_{uN}|L_{f-x-y} >$, where u denotes the u-th row of R, N is the number of detected APs, and $rssi_{uv}$ is the RSSI value of the AP_v ($v \in 1...N$). The RSSIs, $< rssi_{u1}, rssi_{u2}, ...rssi_{uN} >$, are received from a specific virtual grid L_{f-x-y} where f is the floor number, and x, y are the X and Y coordinates of the 2D experimental area. In the proposed GA-based feature selection technique, each chromosome consists of N genes or bits, where N is the number of APs or features. Each gene of a chromosome is encoded using 1 or 0, indicating the presence or absence of the corresponding feature. Therefore, a feature set is a set of APs for which the bit of the chromosome is 1. The initial population size, or the number of chromosomes, is represented by P. The GA-based AP selection

procedure is illustrated in Figure 3.1. The effect of varying the number of chromosomes in the population, P, is reflected in the computational time and the convergence metric.

3.2.1.1 Fitness Function

The purpose of the fitness function is to assign fitness values to various chromosomes or feature vectors. To form a training dataset, the features of a chromosome C_i are used. This training dataset is then shuffled n times to produce n shuffled training datasets. The fitness value of a chromosome C_i is calculated by averaging the accuracies obtained through 10-fold cross-validation using the k-Nearest Neighbors (kNN) classifier on the n shuffled training datasets. Therefore, the fitness value(Equation 3.1) of each feature vector or chromosome C_i can be calculated as follows:

$$Fitness(C_i) = \frac{1}{n}\sum_{j=1}^{n} Accuracy_j(C_i) \tag{3.1}$$

3.2.1.2 Selection

In the selection procedure, a subset of the chromosome is selected from the chromosome pool and passed through the crossover operation while rest of the chromosomes in the population are simply passed to the next step. This selection of chromosomes is carried out with the help of a roulette wheel method that allows the fittest chromosome to have a higher probability of getting selected. The probability of selecting a chromosome(Equation 3.2), $\rho(C_i)$, is given as follows:

$$\rho(C_i) = \frac{\text{Fitness}(C_i)}{\sum_{i=1}^{P} \text{Fitness}(C_i)} \tag{3.2}$$

If two chromosomes have the same fitness value, then the chromosome having the minimum number of features is chosen.

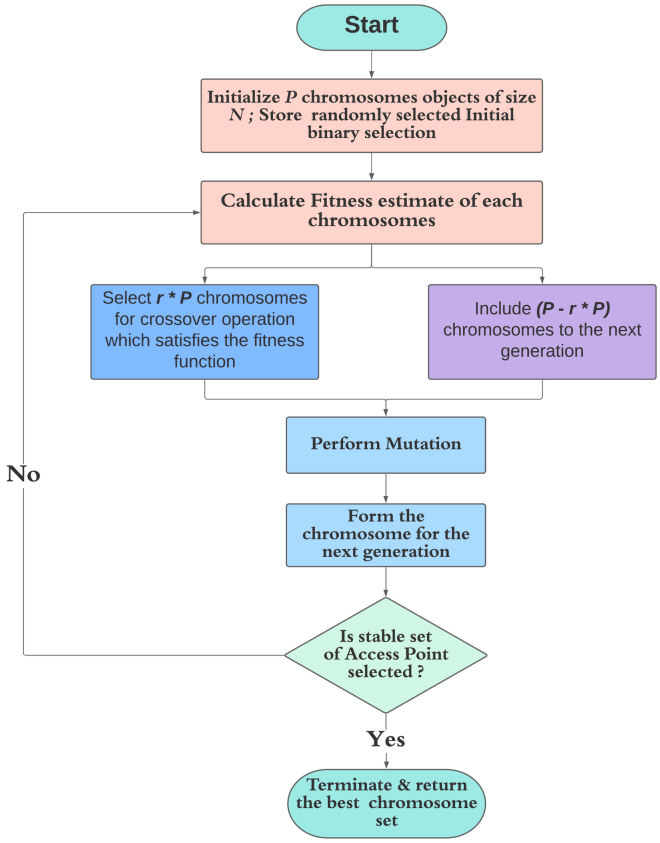


Figure 3.1: The proposed GA procedure for the access point selection problem

3.2.1.3 Crossover

To perform crossover operation in the GA-based feature selection technique, a subset of chromosomes is selected probabilistically from the chromosome pool of the previous generation using Equation 3.2. The chosen set contains $r \times P$ chromosomes, where r is the fraction of the population $(0 < r < 1)$ to be replaced by crossover at each generation, and P is the population size. From this set, a pair of chromosomes is selected for crossover operation using the single point crossover technique. The resultant new sets of chromosomes are illustrated in figure 3.2.

3.2.1.4 Mutation

In the mutation phase $m \times P$ number of chromosomes is selected randomly, where m is the mutation rate $(0 < m < 1)$. For each such selected chromosomes, a random gene is selected and its value is flipped, i.e., if the AP is selected it gets deselected after mutation

or vice-versa as shown in figure 3.2.

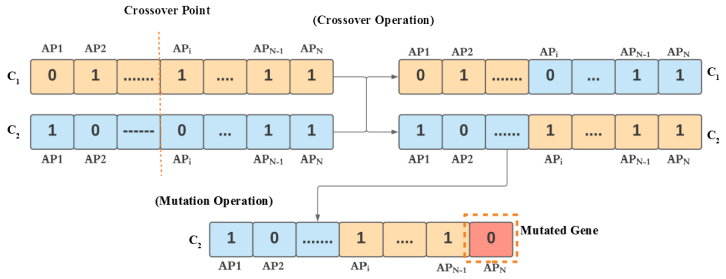


Figure 3.2: An example of Crossover and Mutation Operation

During the initial iteration of the GA, the chromosomes that obtain the highest fitness values is stored. GA is an iterative process; thus, the stopping criteria depends on the convergence of the fitness evaluation process based on a very low threshold difference between the selections of two subsequent generations. Regarding complexity of the selection process, GA is classified as a random-based search algorithm, as randomness is incorporated in each iteration of the algorithm (e.g., selection, crossover, mutation). Therefore, the search complexity of GA [81, 82] depends on the dimensionality of the data (i.e., features). Due to the randomness embedded in the algorithm, exhaustive search is not performed, and the search complexity can be effectively linear to the number of iterations.

3.2.2 Feature-based Ensemble Learning

The proposed approach aims to select a set of access points (APs) that are essential for reliable indoor localization. However, since different ambient conditions and devices can affect the performance of a single set of APs, a feature-based ensemble called FABEL is developed. This considers multiple sets of important APs to be selected using the proposed feature selection technique.

A flow diagram of the proposed feature-based ensemble learning model is depicted in figure 3.3. Using a given training data, GA is iterated for k times with different population sizes (PS) to get k number of feature sets, $\{Fset_1, Fset_2, \ldots, Fset_k\}$. Then, k number of training sets, $\{TRset_1, TRset_2, ..., TRset_k\}$, are generated using k feature sets. In the next step, considering every training set, 10-fold cross-validation is performed using kNN.

After getting the results of every cross-validation process, the similarity value(Equation 3.3), $S(Fset_i, Fset_j)$, between any two feature sets, $Fset_i$ and $Fset_j$, is calculated as follows:

$$S(Fset_i, Fset_j) = \frac{\#\text{ instances for which } Fset_i \text{ and } Fset_j \text{ predicted same labels}}{\#\text{ total instances}} \quad (3.3)$$

In this way, similarity between every pair of AP sets is calculated using Equation3.3. Then, a similarity matrix, Sm, of dimension $k \times k$ is constructed as follows:

$$Sm = \begin{bmatrix} \mathrm{S}(Fset_1, Fset_1) & \mathrm{S}(Fset_1, Fset_2) & .. & \mathrm{S}(Fset_1, Fset_k) \\ .. & .. & .. & .. \\ \mathrm{S}(Fset_k, Fset_1) & \mathrm{S}(Fset_k, Fset_2) & .. & \mathrm{S}(Fset_k, Fset_k) \end{bmatrix} \quad (3.4)$$

Here k' AP sets ($k' < k$) with low similarity values are selected from the k AP sets to build a feature-based ensemble learning approach. Out of these k' feature sets, training sets $TRset_1, TRset_2, ..., TRset_{k'}$ to create k' base learners are chosen. In each base learner, a test set using kNN and a corresponding training set $TRset_i$ ($i = 1...k'$) are classified. After obtaining the classification results, the weighted majority voting among all k' base learners is used to predict the location of a test instance. The weight of a base learner(Equation 3.5), $W(bl_i)$, is calculated as follows:

$$W(bl_i) = \frac{CVaccuracy(TRset_i)}{\sum_{i=1}^{k'} CVaccuracy(TRset_i)} \quad (3.5)$$

Here, $CVaccuracy(TRset_i)$ denotes 10-fold cross-validation accuracy obtained by kNN using training set, $TRset_i$.

The computational time for selection and training is only required for the offline training phase, as the classifiers are tuned during training. During the online phase, the pre-tuned classifiers are used for analyzing the test data. Hence, there is no performance bottleneck during online location prediction. The proposed selection approach is implemented using any standard server and doesn't require any addition hardware support.

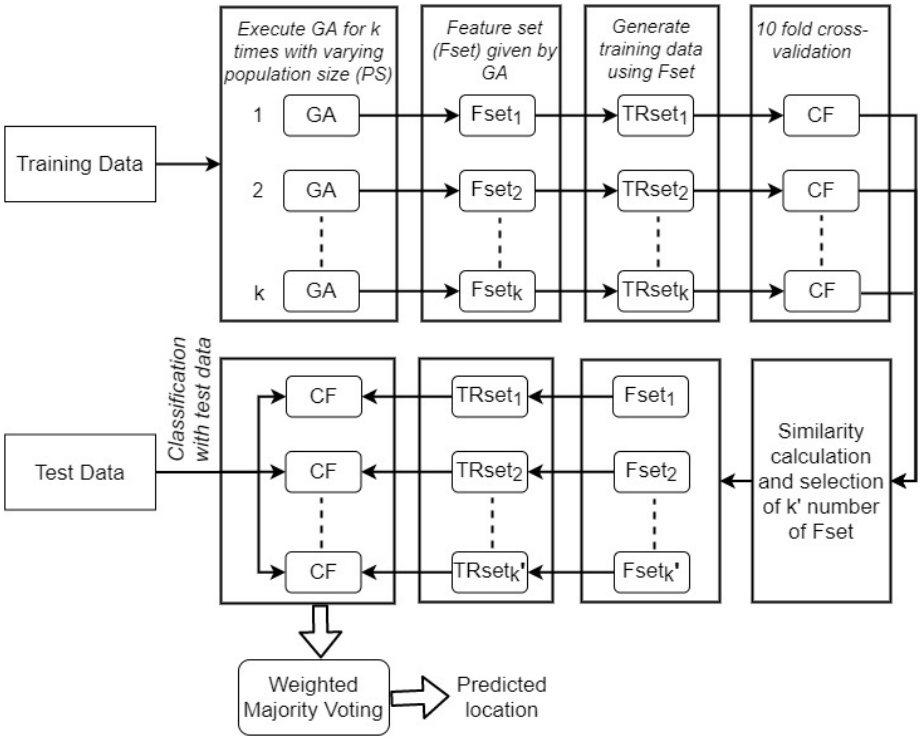


Figure 3.3: A flow diagram of Feature-based Ensemble Learning

3.3 Results and Discussions

The description on the dataset used and the experimental analysis is presented in the following section.

3.3.1 Experimental setup and data description

In this study, the benchmark dataset JUIndoorLoc[71] has been utilized to implement our proposed algorithms. Figure 3.4 shows the experimental region from where the data was collected. A portion of the floor plan is selected for the experimentation where data for each location are reported from four different devices (D1-Samsung Galaxy tab 10, D2-Samsung Galaxy tab E, D3-samsung Galaxy tab 2, and D4-Moto E) respectively. A total of 163 APs out of which 55 APs with non zero standard deviation were there. The analysis was done in Python (3.6) using the Spyder IDE in an Intel Core i5-2320 CPU @3.00GHz, 8 GB memory (RAM), and Windows 10 (64bit) operating system. The built-in libraries of Python were used for the analysis purpose. KNN, SVM, Decision Tree and Random Forest are used for comparing the classification accuracies.

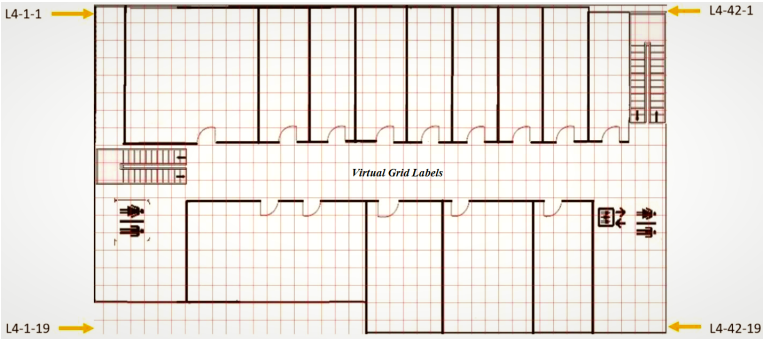


Figure 3.4: Floorplan used for localization

3.3.2 Result analysis

In this section, the performance of the proposed GA-based feature selection algorithm is evaluated and compared with the number of access points required for localization in different areas of the experimental region. The cross-validation accuracies of different feature sets is examined and their performances are compared. Additionally, the accuracies of feature set-based ensembles for different devices is also discussed.

There is no universal approach for determining the optimal values of the primary controlling parameters of the proposed GA. Thus, these parameters were adjusted through experiments, which were repeated several times to ensure the reproducibility of the results. The algorithm was executed and tested for an adequate number of iterations. The crossover ratio and mutation rate were set to 0.8 and 0.3, respectively, and other parameter values were adjusted based on experimentation, as summarized in Table 3.2. Additionally, the parameters of machine learning classifiers were fine-tuned using the cross-validation technique, which is a widely accepted method for model validation in the machine learning field that evaluates the generalization of statistical analysis on an independent dataset.

Table 3.2: GA Procedure Parameter Settings

GA Parameters	***Setting***
Population size	10-100
Generation	500
Selection Approach	Roulette Wheel
Mutation Type	Single Bit Flip
Mutation rate	0.2-0.3
Crossover Type	Single Point
Crossover rate	0.8

The floorplan used for JUIndoorLoc consisted of both corridors and rooms. The cross-validation accuracy of kNN for both rooms and the entire floor map is illustrated in Figure 3.5. Figure3.5(a) illustrates the impact of the number of significant access points (APs) on the localization performance of two devices $D1$ and $D2$ for a specific room in the experimental region. As observed in the figure, stable localization performance can be achieved with up to 8 significant APs. The accuracy remains between 93.92% and 95.05% when the number of APs is equal to or greater than 8. When all the 55 APs of the region were employed, an accuracy of 95.04% was attained.

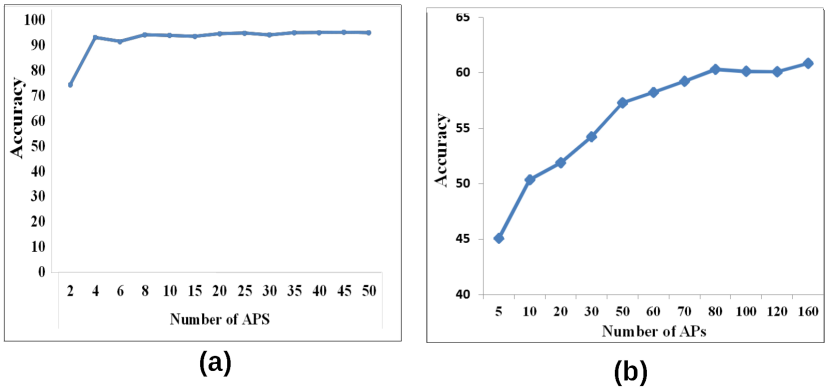


Figure 3.5: Classification accuracy for the kNN classifier subject to different set of important APs selected by the proposed GA based feature selection technique (a)Room Level (b)Entire Floor

3.3.3 GA Performance analysis

Figure 3.5(b) depicts the changes in 10-fold cross-validation accuracies with different predefined limits on the number of APs for the entire experimental region, using devices $D1$ and $D2$. The results indicate that there is no significant change in accuracy when the number of APs is increased beyond 80APs. The accuracy remains between 60.3% and 60.85% when the APs are equal to or greater than 80. An accuracy of 59.56% was obtained when all 163APs of the region were used. Therefore, the findings suggest that around 50% of the APs can be identified as necessary for sustaining localization performance. Consequently, the total maintenance cost of the WiFi infrastructure required for localization can be reduced.

The following experiment was conducted to demonstrate the impact of the proposed

feature selection approach on indoor localization using different classifiers, including k-NN, Support Vector Machine (SVM), Decision Tree, and Random Forest Classifier. Fig 3.6 compares the classification accuracy before and after the application of the GA-based feature selection method. The parameter settings of the classifiers used during the experimentation are summarized in Table 3.3. The models were trained using data from two devices, $D1$ and $D2$, and the accuracy was obtained using different machine learning classifiers. The results show that Decision Tree achieved high accuracy for the selected floor plan. Moreover, it can be observed that the GA procedure reduced the feature set from 55 APs to 22 APs with no significant impact on the accuracy values.

Table 3.3: Tuning Parameters of the Classifiers

Parameter	Setting
number of neighbors for kNN	*5*
SVM Kernel	*Polynomial*
Decision Tree Max Depth	*90*
no of trees in Random Forest	*60*

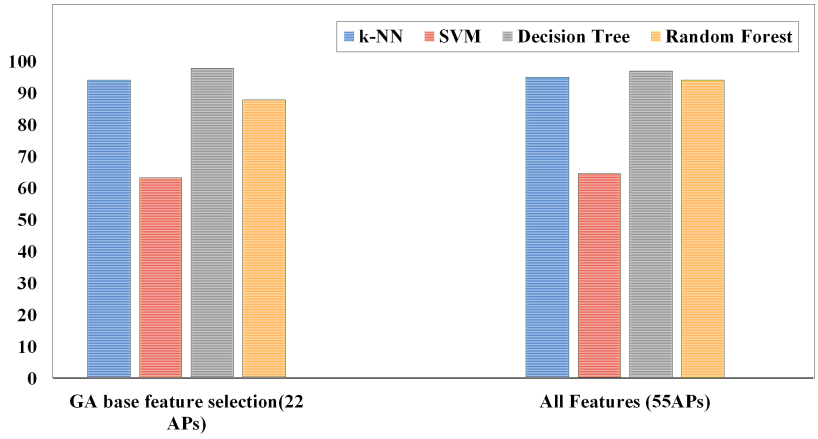


Figure 3.6: Accuracy before and after application of GA for the selected Floor Map

3.3.4 AP selection Analysis

Experiments are conducted by varying the initial population in the range of 10 to 100 for both the room as well as the entire floor plan. Choosing a different set of the initial population gives the GA a chance to explore the experimental region properly and avoid getting stuck at a local optima. Figure 3.7 gives an overview of the variation of the number of APs selected with each iteration of the GA process.

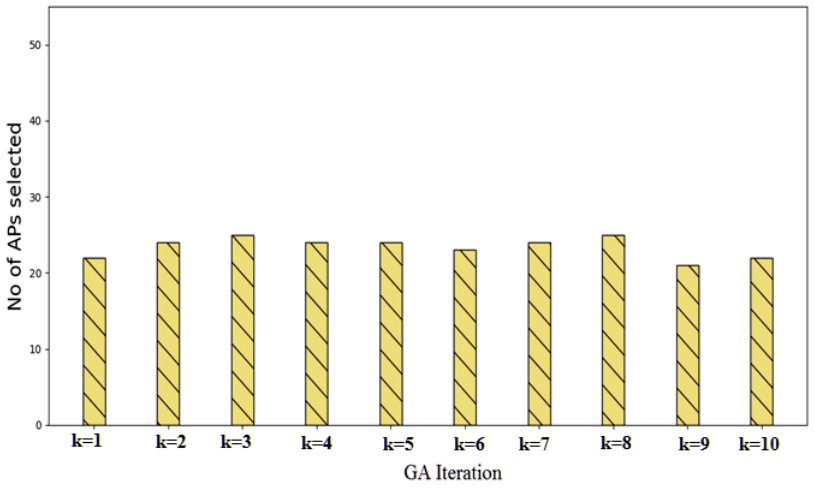


Figure 3.7: Graph depicting the number of features selected on each run of the GA Procedure when k parameter is set to 10

The results show the count wise stability of the selection process. The classification accuracy of the selected features on every iteration are close to 94% with very small standard deviation. For dataset pertaining to a room, the variation in initial population did not quite impact the accuracy metric. However, it has been observed that although the accuracy values are similar, the underlying sets of important APs are different. This justifies the formation of the proposed feature based weighted ensemble of classifiers.

3.3.5 Device Heterogeneity Testing

An experiment to investigate the performance of proposed ensemble as to whether it can really address different training and testing condition has been carried out. Figure 3.8 gives an overview of the classification accuracy metric of the selected base learners and the feature-based weighted majority voted ensemble learning. In figure 3.9, the training dataset on which GA was applied is taken using devices D1 & D2. The test set reports RSS from a different device D4. The results indicate that the ensemble of classifiers provide a relatively better classification outcome even when the training devices and the testing device are same or different. Thus, the proposed feature based ensemble method can cope with heterogeneous devices.

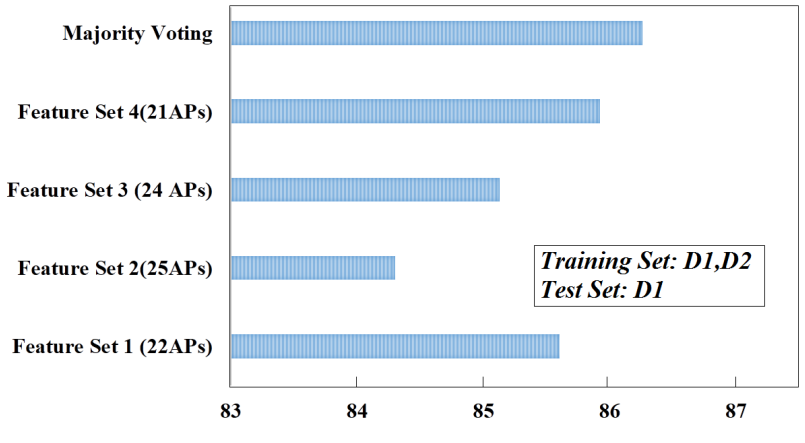


Figure 3.8: Classification accuracy on application of the proposed feature based ensemble model where training is done on data collected by the devices D1,D2 and testing is done on data collected by the device D1

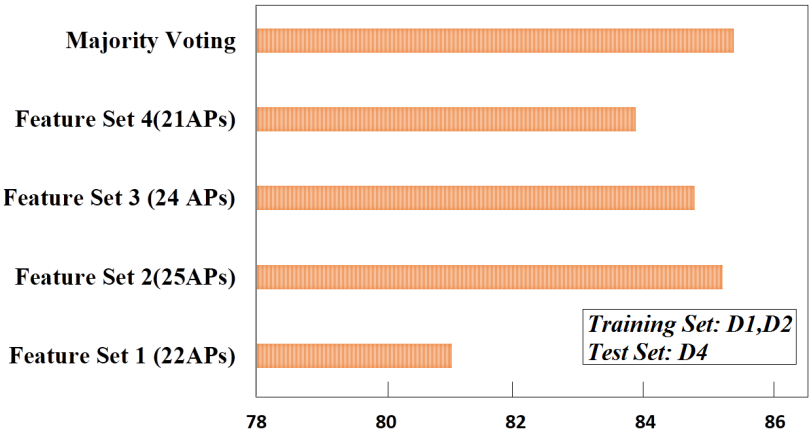


Figure 3.9: Classification accuracy on application of feature based ensemble model where training is done on data collected by the devices D1,D2 and testing is done on data collected by the device D4

3.3.6 Error deviation analysis

There are other performance measures besides classification accuracy that are important for indoor localization. One such measure is the mean absolute error, which considers the deviation between the predicted and actual location. The location is represented as a (P_i, Q_i) coordinate on the floor plan. Let (P_i', Q_i') be the predicted label and the estimated Euclidean distance (Equation 3.6) in meters; where i is a index in the test instance.

$$error = \sqrt{(P_i - P_i')^2 + (Q_i - Q_i')^2} \tag{3.6}$$

For the misclassified instances, the Mean Absolute Error (MAE) is estimated using Equation 3.7; n is the count of test data instances.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{3.7}$$

The mean absolute error (MAE) is evaluated in our indoor localization problem by considering the deviation between the predicted and actual location, where each label is represented as a (P, Q) coordinate on the floor plan. The predicted label and the corresponding Euclidean distance are denoted as (P', Q') and calculated using Equation 3.6, respectively. To estimate MAE, the number of generations and the initial population sizes in the GA procedure are varied. The Cumulative Distribution Function (CDF) for the same is calculated, as shown in Figure 3.10. CDF is a widely used statistical metric to determine the probability of errors that fall within a certain distance, say x meters. The result shows that when the initial population size and the number of generations are fixed at 30, approximately 80% of the errors lie within 4m. It has also been observed that increasing the population size and number of generations does not significantly decrease the error metric.

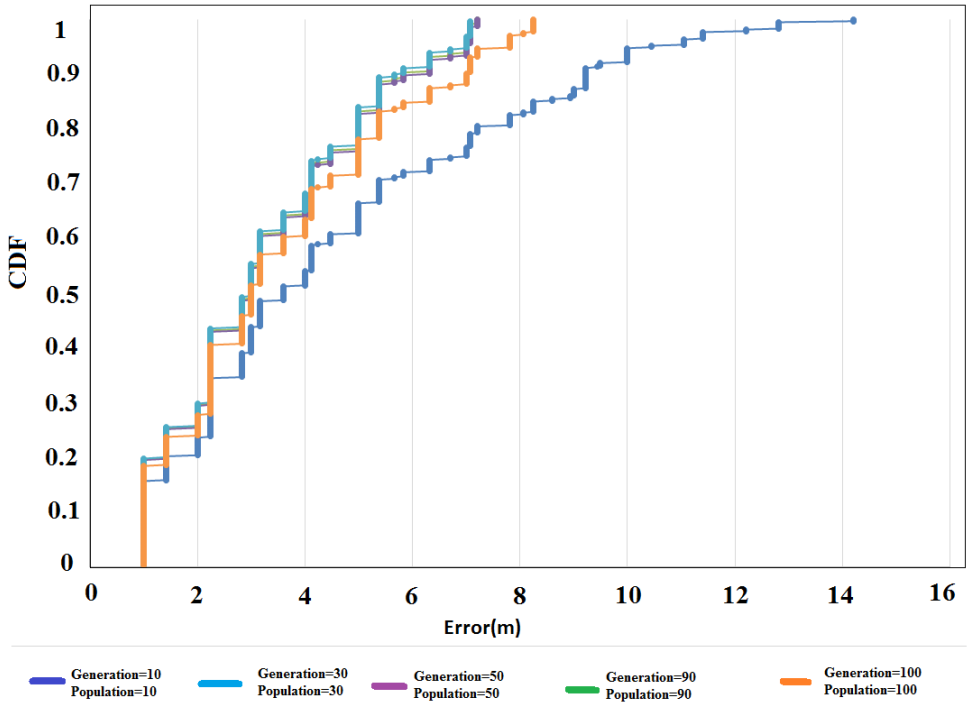


Figure 3.10: CDF Plot on varying Generation and Population size

3.3.7 Comparison with state-of-the-art feature selection approaches

The proposed feature selection approach is compared with popular methods such as PCA [83] and select-k-best [84]. PCA is based on covariance and selects uncorrelated sets of APs or principal components to ensure diversified information for the classification model. On the other hand, select-k-best selects k-best AP sets based on p-value, which indicates the probability of RSS values occurring by chance. F-Test [85] is used as the score function for the selection of APs in select-k-best.

In figure 3.11, the entropy analysis for the JUIndoorLoc dataset is showcased. The number of feature parameters for PCA and k parameter of select-k-best are set to the average number of feature set count of the proposed GA procedure. PCA achieves the highest average entropy (11.604), indicating that it captures the most diverse and complementary information from the features by emphasizing variance. In contrast, SelectKBest yields the lowest average entropy (5.08), which captures individual relevance by sacrificing diversity for precision. While the proposed GA-based approach is striking a balance, with an intermediate entropy of 8.13. Hence, GA provides a balance by combining both diversity and relevance.

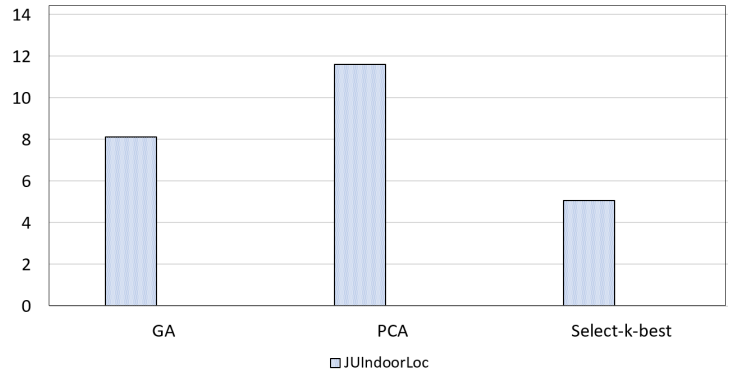


Figure 3.11: Entropy estimation comparison with state-of-the-art feature selection approaches with our proposed approach

The accuracy comparison of the approaches is illustrated in figure 3.12. The kNN procedure is used with the same parameter condition and dataset for fair comparison between the approaches. The results show that the proposed approach outperforms the other methods. This could be attributed to the existence of more than one global best AP feature set. With multiple iterations of the GA process during training, the proposed

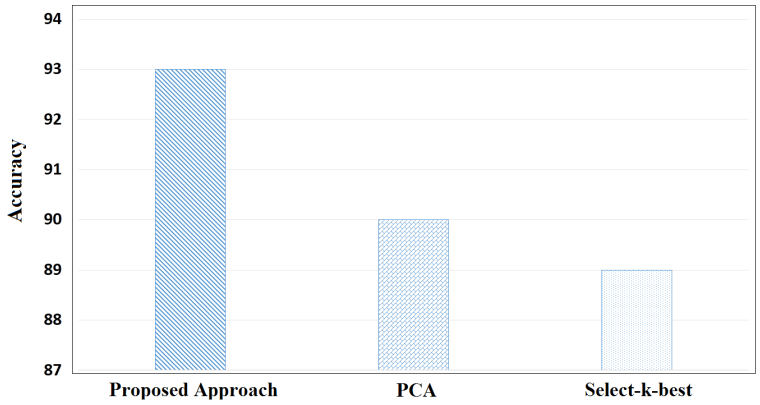


Figure 3.12: Accuracy comparison with state-of-the-art feature selection approaches with our proposed approach

approach is able to capture the majority of the optimal selections, which is not possible with select-k-best or PCA approach. Additionally, the feature-based ensemble is able to consider various conditional contexts, which aids in the classification process. Moreover, if one of the base learners fails to map to a particular location, the other base models contribute to the prediction process.

3.4 Summary

The work presents a novel training pipeline incorporating genetic algorithm to capture the different combinations of important sets of APs to increase the adaptiveness of the system. Selection of the important sets of APs is crucial for positioning in an indoor environment. The GA selection procedure follows binary encoding of the chromosomes represented as the APs. A reduction of almost 50% in the AP count was achieved with appreciable accuracy. The selected AP sets are passed through a feature based ensemble classifier in order to retain generality. The proposed training pipeline has been tested for device heterogeneity.

Wrapper based approach do not take into consideration the intrinsic properties between the features which is catered in filter based approaches. Furthermore, the methods can be computationally expensive for datasets with a large number of features. In contrast, Filter based feature selection methods are computationally efficient and can handle

high-dimensional datasets. In next chapter, the work focuses on the Filter based feature selection approach for indoor localization using Wi-Fi signals.

Published Journal:

1. Parsuramka, S., **Panja, A.K.**, Roy, P., Neogy, S. and Chowdhury, C., 2023. FABEL: feature association based ensemble learning for positioning in indoor environment. Multimedia Tools and Applications, 82(5), pp.7247-7266. *(IF : 3.6)*

Chapter 4

Filter based Feature Selection approach using BPSO

Filter-based feature selection is a type of feature selection technique that involves selecting features based on the intrinsic characteristics between the features. Hence, this can be applied to WiFi RSS-based indoor localization to select the most relevant features (i.e., WiFi access points). These features are able to provide accurate and robust indoor positioning. In filter-based feature selection, various statistical techniques such as correlation-based feature selection, variance thresholding, mutual information-based feature selection, chi-squared test(χ^2), or ANOVA F-test can be applied to select the most relevant WiFi access points. For example, the correlation-based feature selection [86] can be used to select the top k access points that are most correlated with the target variable (i.e., user location), while the variance thresholding can be used to remove the access points with low variance as they may not be informative. Similarly, the mutual information-based feature selection can be used to select the top k access points that have the highest mutual information with the target variable.

Generally, the APs for a particular floor are selected based on the criteria, such as signal strength, information gain, fingerprint clustering, and the like. [23]. Approaches are more focused on clustering the regions into sub-regions and considering criteria such as *information gain* and *power* from APs as the basis for selection [35, 37]. The defined

techniques are suitable for a *static* floorplan. As discussed, it is not always true that a single AP would provide a stable RSSI value over all of the selected areas under any ambient condition. Hence, a sustainable positioning scheme is required for any generalized floor.

In this chapter, a novel positioning strategy for smartphone users using Binary Particle Swarm Optimization (BPSO) is proposed. BPSO is used for AP selection and defining a feature-based ensemble model to cover different ambient variations. It has been observed from the past research that Particle Swarm Optimization (PSO) is used in solving the AP placement problem [87, 88] for WiFi-based indoor positioning and are reported to give significant accuracy than other approaches. The placement of AP well aligns with the problem of AP selection. The motive behind using a BPSO approach is to find the local and global best selection of APs that leads to better exploration and selection. In the majority of the floor plans selected for positioning, some APs are found to impart similar information to localization, that is they are highly correlated w.r.t a given training context. Thus, there can be more than one global optimal selection of APs for that context. With different ambient conditions and devices, the context varies during testing. Hence, to retain generality irrespective of context, a feature-based ensemble approach is developed. The feature-based ensemble classifier is proposed based on the important feature subsets obtained by the use of the BPSO approach to perform positioning. In summary, the following are the contribution of the work;

1. BPSO based feature selection technique is proposed for smartphone based Indoor Positioning.

2. Based on the selected feature sets, a feature based ensemble model is designed. A neural network based meta model classifies the test instances based on the prediction outcomes of the base learners.

3. The proposed BPSO based feature selection approach is tested for accuracy against the collected real-life dataset and chosen benchmark sets. The proposed training pipeline has been tested for location independence.

The chapter is organized in the following manner. Section 4.1 gives an overview of the

BPSO approach. In Section 4.1, the AP selection mechanism using metaheuristic based approach is discussed followed by the detailing of the proposed feature based ensemble approach. The experimental set-up along with extensive experimentation and results are discussed in Section 4.2. The chapter concludes in Section 4.3.

4.1 Binary Particle Swarm Optimization overview

Swarm intelligence (SI) is a behavior of self-organized systems that is decentralized and collectively takes place. PSO is one such approach which is an *evolutionary* meta heuristic-based procedure. Binary PSO is a variant of naive PSO where every particle is a candidate solution to the selection problem. Every particle has a position vector to hold the current selection of WiFi AP. In the PSO approach, there are two update rules, the velocity update function responsible for the speed and direction of the particles and the particle's position update function. In BPSO, the update rule is a bit different as we are dealing with binary values. A generalized velocity update rule is given in Equation 4.1. Let $P_{i,j}^{(t)}$ be the vector for position or selection of the i^{th} particle for a generalized BPSO representation at iteration t. Let $vel_{i,j}^{(t)}$ be the velocity value of the j^{th} position of the i^{th} particle at iteration t which can be initialized to 0 or random values for each particle of the swarm. $Pbest_{i,j}$ and $Gbest$ are generalized forms of a particle's best selection and global best selection vectors. The velocity update is as follows.

$$\begin{aligned} vel_i^{(t)} = vel_i^{(t-1)} \times W + c1 \times r1 \times (Gbest^{(t-1)} - P_i^{(t-1)}) \\ + c2 \times r2 \times (Pbest_i^{(t-1)} - P_i^{(t-1)}) \end{aligned} \tag{4.1}$$

The position update rule for the particles is as follows.

$$sigmoid(vel_{i,j}^{(t)}) = \frac{1}{1 + e^{-vel_{i,j}^{(t)}}} \tag{4.2}$$

$$P_{i,j}^{(t)} = \begin{cases} 1, & \text{if rand() } <= sigmoid(vel_{i,j}^{(t)}) \\ 0, & \text{if rand() } > sigmoid(vel_{i,j}^{(t)}) \end{cases} \tag{4.3}$$

Like PSO, an estimation of *cost* or a *fitness* function is formulated to decide about the applicability of the selection/ position update rule (Equation 4.3) in BPSO for a better solution. The *rand*() function generates a pseudo-random number which is taken from a uniform distribution in the range of 0 to 1. Leaders are chosen among the particles which have a better fitness score. They indicate the likelihood of a good solution. The PSO approach has variants that are distinguished by its initialization and weight parameters. The initialization of the particle vectors greatly impacts the performance of the process. Initialization can be random or based upon some analysis like opposition based initialization [89], chaotic approach [90], and so on. Exploration and exploitation of the swarm can be controlled by the weight parameter along with the social and cognitive components. These are essential for finding the unknown global optimal solution.

The BPSO based feature selection algorithm is presented first, followed by the description of the feature based ensemble. The system model used in this work is as follows. The floorplan F is divided into virtual grids of equal-sized blocks represented as $L_{z,x,y}$, where (x, y) are the grid coordinates and z is the floor number. The radiomap vector contains the collected RSS values from all the registered APs on the floor. The radiomap vector R is denoted as $< rss_{u1}, rss_{u2}, ...rss_{uM}|L_{z,x,y} >$, where u represents the u^{th} row of the fingerprint vector; u is varied from 1 to N, that is the no. of instances. M is the number of detected APs and rss_{uv} denotes the RSSI fingerprint value of AP_v ($v \in 1...M$). The collected radiomap data is preprocessed, which is discussed vividly in Section 4.2.1. For instance, missing values are either interpolated or often replaced by poor signal strength values. The preprocessed radiomap R' is considered for feature selection containing N' instances or rows.

4.1.1 BPSO based Access Point Selection Mechanism

The signal strengths of the WiFi APs do not vary uniformly with the distance. The RSS values decrease as one moves away from an AP source. However, for weak signal strengths, this change in RSS value with distance is minimal. Thus, there could be some APs over the experimental region whose signal strengths are too weak to show any distance sensitivity. That is, information gain from these APs is almost negligible. The APs with a Standard Deviation (SD) of 0 are deselected from R as the RSS values from them do not contribute to any localization information. Thus M' denotes the count of the effective APs. Table 4.1 lists out the variables involved in the BPSO process along with their significance. In Figure 4.1, a block diagram is shown to present an overview of the vectors involved in the process. P represents the swarm object vector where each index of P pertains to a particle object, referenced as P_i. The particle count or the swarm population is referred to with variable $pSize$. Each particle is represented by a candidate solution, that is, the current selection of APs denoted in binary vector form as, $P_i.feature[j]$. Velocity vector of a particle, $P_i.vel[j]$ signifies the process of updation of a selection in order to attain optimality. $P_i.feature$ is a binary vector where a random initial selection is assigned; 1 denotes the selection of the WiFi AP, and 0 indicates that the AP is not selected. $P_i.Pbest$ gives the local best selection of APs by particle i, and $Gbest$ is the global best selection of APs. Here i is iterated from 1 to $pSize$ and j from 1 to M'.

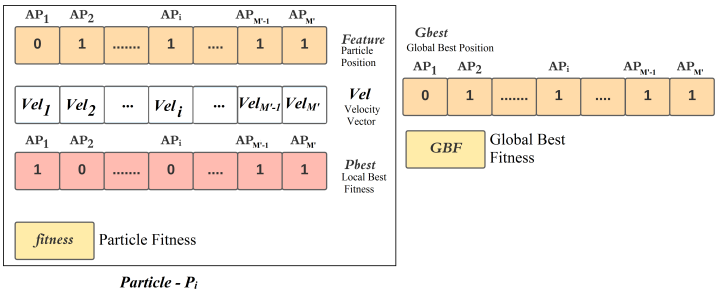


Figure 4.1: Block diagram depicting the vectors involved in the BPSO procedure

The proposed selection procedure is demonstrated using a flowchart given in Figure 4.2. For every run of the BPSO, the velocity calculation and the particle's selection update rule are iterated for $maxIter$ times or until convergence. The difference $(avgF - GBF) \leq \epsilon$ should hold for achieving convergence, where $avgF$ is the average fitness score, and

ϵ is appreciably small. The BPSO based feature selection procedure is carried out for k number of iterations to capture k combinations of feature sets. In the following subsection, the fitness function to calculate the score of the particles is discussed.

Table 4.1: BPSO Feature Selection Terminologies

Variables	***Significance***
M'	Number of APs after columns with SD=0 removed
N'	Number of instances in Preprocessed Radiomap
$R'_{u,v}$	Preprocessed Radio Map Database;$1 \leq u \leq N'$ and $1 \leq v \leq M'$
$pSize$	Swarm Size or population size
P_i	Particle object vector
$P_i.feature$	Current selection of APs (binary vector) by ith particle
$P_i.vel$	Velocity vector of Particle i
$Gbest$	Global best selection of AP (binary vector)
$P_i.Pbest$	Local best AP selection vector (binary vector) by particle i
GBF	Global best fitness
$C_{v1,v2}$	Pearson Correlation Coefficient between $v1$ and $v2$ AP
$bcount$	Feature Base Learner count
$c1$	Social acceleration coefficient,
$c2$	Cognitive acceleration coefficient
$maxIter$	Maximum iteration
$avgF$	Average fitness score of all the particles
ϵ	Threshold for convergence

4.1.1.1 Fitness Function

The inherent correlation within the APs is estimated with the Pearson Correlation Coefficient [91] to measure the association or relationship between the APs. To measure the fitness (Algorithm 2) of the selected set of APs, a 2D correlation coefficient vector or cost matrix C is evaluated. A particular cell $C_{v1,v2}$ gives the correlation coefficient between recorded RSSI of AP $v1$ and AP $v2$; $v1, v2 \leq M'$. The Cost Matrix C is estimated using

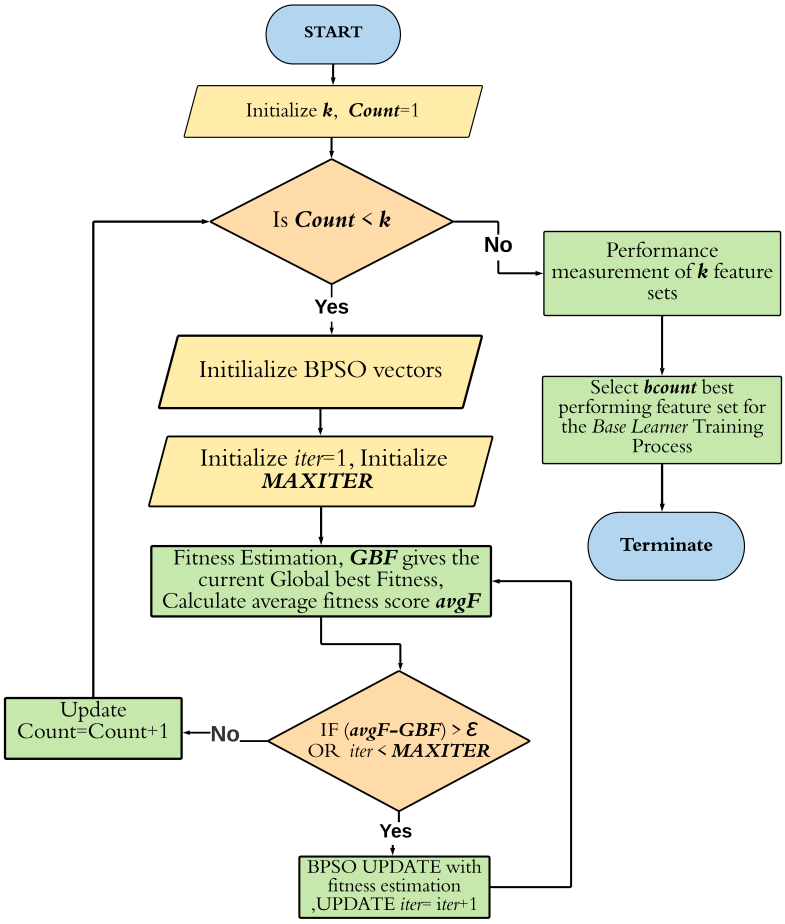


Figure 4.2: Access Point Selection Procedure using Binary PSO

the given Equation 4.4.

$$C_{v1,v2} = \frac{\sum_{l=1}^{N'}(R'_{l,v1} - \bar{R'_{l,v1}})(R'_{l,v2} - \bar{R'_{l,v2}})}{\sqrt{\sum_{l=1}^{N'}(R'_{l,v1} - \bar{R'_{l,v1}})^2 \sum_{l=1}^{N'}(R'_{l,v2} - \bar{R'_{l,v2}})^2}} \tag{4.4}$$

R'_{l,v_i} and R'_{l,v_j} are the RSS values from AP v_i and AP v_j, while on varying l select the different records on the floor map. The objective is to minimize the correlation, hence, the fitness function is given as follows.

$$\min \prod_{v_i=1}^{M'} \sum_{v_j=1}^{M'} C_{v_i,v_j} \tag{4.5}$$

This fitness function(4.5) aims to minimize the cumulative correlation between all pairs of access points in the system. The sums of the correlation coefficients for a given access point v_i with every other access point v_j is given by $\sum_{v_j=1}^{M'} C_{v_i,v_j}$. The outer product over

all access points v_i aggregates the correlations to obtain a holistic measure of dependency across the entire network. Minimizing this product ensures that the correlations between different access points are as low as possible, enhancing the distinctiveness of the RSS fingerprinting for localization.The objective of selecting this fitness metric is to improve the robustness and accuracy of RSS-based indoor localization by ensuring that each access point provides independent information about the user's location.

Algorithm 1 calculates the cost matrix C for a particular particle. The particle's current selection is passed to the function as parameter $SFeature$. The selected APs from $SFeature$ are first extracted to a collection (lines 6-11 of Algorithm 1). This collection is utilized to calculate the pairwise correlation of all the selected APs through lines 12-16 of Algorithm 1. The fitness is estimated using Algorithm 2. The fitness is the product of sum (line 3-9) of rows estimated from the cost matrix C for the respective particle.

Algorithm 1: costMatrix(R', $SFeature$,M')

```
   Result: Correlation Coefficient Matrix C_{v1,v2}
 1 SFeature_v - selected binary feature vector (v ≤ M'),
 2 R'- Preprocessed Radio Map Database
 3 temp ← 0
 4 Initialize dynamic vector column
 5 Initialize each location of vector C[M' x M'] to 0
 6 for v=1 to M' do
 7 |   if SFeature[v]==1 then
 8 |   |   temp ← temp + 1
 9 |   |   column[temp] = v
10 |   end
11 end
12 for each v1∈ column do
13 |   for each v2 ∈ column;v2≠ v1 do
14 |   |   Estimate C_{v1,v2} using Eq 4.4
15 |   end
16 end
17 return C, column
```

4.1.1.2 BPSO Update

The update function constitutes the velocity updation and AP selection update for each particle, the update rule is stated in Equation 4.6-4.7; where $i \leq pSize$ and $j \leq M'$

Algorithm 2: fitnessCalc($C, column$)

```
C: Cost matrix, column:Vector containing the selected APs
fitness ←1
for each v1 ∈ column do
    temp ← 0
    for each v2 ∈ column;v2≠ v1 do
        temp ← temp+ C_{v1,v2}
    end
    fitness ← fitness × temp
end
return fitness
```

$$P_i.vel = P_i.vel \times W + c1 \times r1 \times (Gbest - P_i.feature) + c2 \times r2 \times (P_i.Pbest - P_i.feature) \tag{4.6}$$

$$P_i.feature[j] = \begin{cases} 1 & \text{if } sigmoid(P_i.vel[j]) >= 0.5) \\ 0 & \text{if } sigmoid(P_i.vel[j]) < 0.5) \end{cases} \tag{4.7}$$

Algorithm 6 depicts the procedure of the BPSO update rule used in the feature selection procedure. The fitness score of particle i referenced as $P_i.fitness$ (local best fitness) is calculated using Algorithm 2. *Gbest* indicates the global best selection of WiFi AP set selected among the particles. The global best vector is updated whenever a better fitness score is found among the current selection. If the fitness score is less than the previously estimated fitness score $P_i.fitness$, the local best selection or particle best selection $P_i.Pbest$ is updated with the newly estimated feature combination, and the local best fitness score $P_i.fitness$ is also updated accordingly. The new velocity of the particle P_i is evaluated by substituting the old velocity of the previous iteration into the Equation 4.6. The current selection of the particle is evaluated by passing the updated velocity vector through the sigmoid function (Equation 4.7) and the comparison is done in Equation 4.7. During the process of velocity updation in every PSO procedure, difference between the previous best selection vector and the present selection vector is calculated both for the cognitive and

the social component. The cognitive component is calculated as $P_i.Pbest - P_i.feature$ and the social component is calculated as $Gbest - P_i.feature$.

For both the *cognitive component* and the *social component* the difference is estimated by subtracting 0's and 1's. So the possible values for the difference can be $0, 1$ and -1. For APs that are selected in both the vectors of $P_i.Pbest$ and $P_i.features$, the difference becomes 0. The change in the AP selection metric happens only when the difference is -1 & 1. If the AP is present in $P_i.features$(particles i's current selection) and not in $P_i.Pbest$((particles i's best selection) or $Gbest$(global best selection) the difference is -1. The velocity vector decreases towards the negative x-direction, hence the sigmoid function is pushed towards 0. Similarly, if an AP is not selected in P_i but is present in $Pbest$ or $Gbest$, the sigmoid function is pushed towards 1.

4.1.2 Discussion on Convergence

The convergence of BPSO depends upon various criteria. In standard BPSO, velocity contains both direction and speed; hence, it is a vector quantity. Thus, any updation in the particle's velocity brings changes in the speed and the selection/deselection of features. In [92], the authors have proved that if the PSO algorithm is convergent, the velocity of the particles moves towards zero or stays unchanged until the end of the iteration. However, for the current work involving BPSO, the velocity update rule is a little modified that involves the sigmoid function as stated in Equation 4.7. Hence, when the velocity reaches zero, $sigmoid(P_i.vel[j]) = 0.5$ for the AP j of the i^{th} particle.

Now, for $P_i.vel[j] > 0$, $sigmoid(P_i.vel[j]) > 0.5$. Thus, no bit flip occurs following Equation 4.7; i.e. no selection or de-selection of AP occurs. However, if $P_i.vel[j] < 0$, then, $P_i.feature[j]$ changes from 1 to 0. Thus, when $P_i.vel[j] \approx 0$, bit flip may occur depending on the direction of change of value. However, the weight W plays a crucial role here [93]. In Equation 4.6, W is set to 1 for this work. Consequently, following the stopping criterion $avgF - GBF < \epsilon$ as stated in the flowchart (Figure 4.2), once the Gbest stabilizes, convergence could be achieved as the velocity, though closer to 0, would remain unchanged.

Algorithm 2: bpsoUpdate(P)

$Gbest$:Vector storing Global Best Position, GBF:Global Best Fitness,
M':Total number of APs (SD=0 removed) , $P_i.Pbest$:Local best selection of AP by particle i, $P_i.feature$:Vector containing the APs selected by particle i, $P_i.vel$:Velocity vector of particle i, $P_i.fitness$:Fitness value of selected AP by particle i.
for *i=1 to pSize* **do**
  $cognitive \leftarrow (Gbest - P_i.feature) * c1 * r1$
  $social \leftarrow (P_i.Pbest - P_i.feature) * c2 * r2$
  $P_i.vel \leftarrow W * P_i.vel + cognitive + social$
  for *j=1 to M'* **do**
    $temp \leftarrow sigmoid(P_i.vel[j])$
    if $temp \leq 0.5$ **then**
      $P_i.feature[j] \leftarrow 0$
    else
      $P_i.feature[j] \leftarrow 1$
    end
  end
  $SFeature \leftarrow P_i.feature$
  $cMat, column = costMatrix(R', SFeature, pSize)$
  $tempFitness \leftarrow fitnessCalc(cMat, column)$
  if $tempFitness < P_i.fitness$ **then**
    $P_i.Pbest \leftarrow SFeature$
    $P_i.fitness \leftarrow tempFitness$
    if $tempFitness < GBF$ **then**
      $Gbest \leftarrow SFeature$
      $GBF \leftarrow tempFitness$
    end
  end
end

Space Complexity: In the AP selection procedure, every particle P_i has 3 memorizer vector associated with them namely: $P_i.vel$, $P_i.Pbest$ and $P_i.feature$. The dimension of each vector is of the order of number of APs which in this case is represented with M' (APs with $SD = 0$ removed). Hence, if there are $pSize$ number of particles then the total space complexity is $O(3 * pSize * M')$.

The BPSO based AP selection is a part of the offline training phase which is performed in the server.

4.1.3 Design of the proposed Feature based Ensemble Model

The BPSO is executed a number of times obtaining k combination of feature sets. The radiomap datasets are formed using data collected from multiple devices under different ambient conditions. Due to the difference in sensitivity of devices and other environmental factors, the signal strength obtained from the same set of APs might have variation (slight or significant), which adds an extra dimension to the optimization problem. Two different feature sets on the same floormap may impart an equivalent amount of information for localization, i.e., they lie on the Pareto-front. As the ambient conditions for training and testing may vary, it is not easy to ascertain which set of selected APs out of the k combinations would work better than the other for a given test condition. Hence, to retain the generality of the classification model, an ensemble of classifiers is built based on the different feature sets selected from the k combinations. A base classifier is tuned individually for each of these feature sets. In the end part of the pipeline, a neural network with two hidden layers used as a meta-model is proposed that predicts the outcome depending on the base classifier outcomes. The backpropagation algorithm here tunes the weights for the meta-model during training. Figure 4.3 depicts the overview of the feature-based ensemble model. It is based on the stacking mechanism. The proposed model is summarized in Algorithm 3.

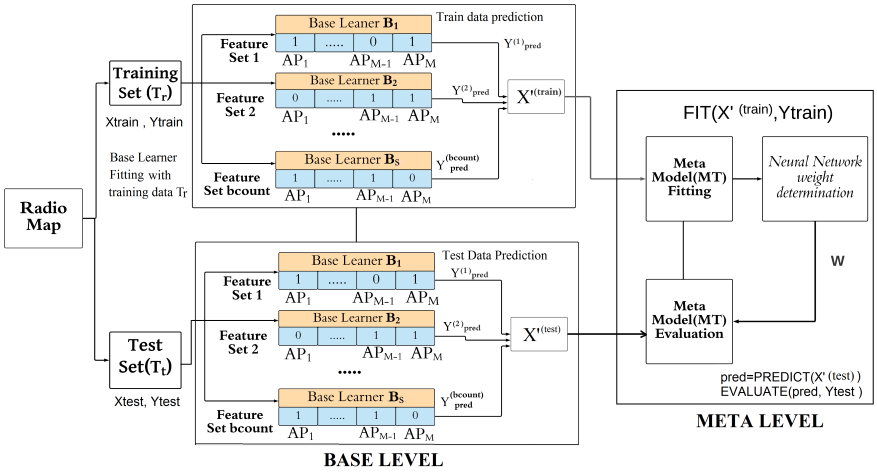


Figure 4.3: Framework of the Proposed Ensemble of Classifiers

The selected top performing feature sets are referenced using the vector $apSet$; where

$apSet$={ $apSet_1$,$apSet_2$...$apSet_{bcount}$ }. During the training phase, the base level model B_s is trained with training set, $T_r = (X_{train}, Y_{train})$; where $T_r \in R'$ and s is iterated from 1 to *bcount*. The output Y_{pred} from base learners are aggregated as X' which forms the feature vector for fitting the meta model. The meta model MT is trained with X' with its respective class label Y_{train}.

Algorithm 3: stackingEnsemble($apSet, R'$)

Input: R' Preprocessed radio map vector, $apSet[\,]$ Access point sets selected by BPSO,
bcount: number of selected subsets of APs
Output: Prediction outcome of the ensemble
Initialize Training and Test Set from Radio Map R'
$T_r \leftarrow (X_{train}, Y_{train})$
$T_t \leftarrow (X_{test}, Y_{test})$
for *s=1 to bcount* **do**
    $tempT_r \leftarrow T_r$
    Deselect APs $\notin apSet_s$ from $temp_{T_r}$
    Initialize Base learner B_s with the Training Algorithm
    Fit the Model B_s with $tempT_r$
end
Initialize empty vector Y_{pred}
for *s=1 to bcount* **do**
    Deselect APs $\notin apSet_s$ from X_{train}
    $Y_{pred}^{(s)} \leftarrow B_s(X_{train})$
    /* Get the predicted labels of X_{train} into vector Y_{pred} */
end
$X' \leftarrow Y_{pred}$ is the input feature to the Meta Model, MT
Initialize the architecture of the $MT \leftarrow \{layer_1, .., layer_d\}$
Learn the weights of the MT by fitting (X', Y_{train})
Evaluate the model by T_t

During the validation of the learning process, the validation set $T_t = (X_{val}, Y_{val})$ is passed through both base level (B_s) and the meta level MT. Validation of the model takes place by comparing the predicted output from the meta model with the actual class label Y_{val}.

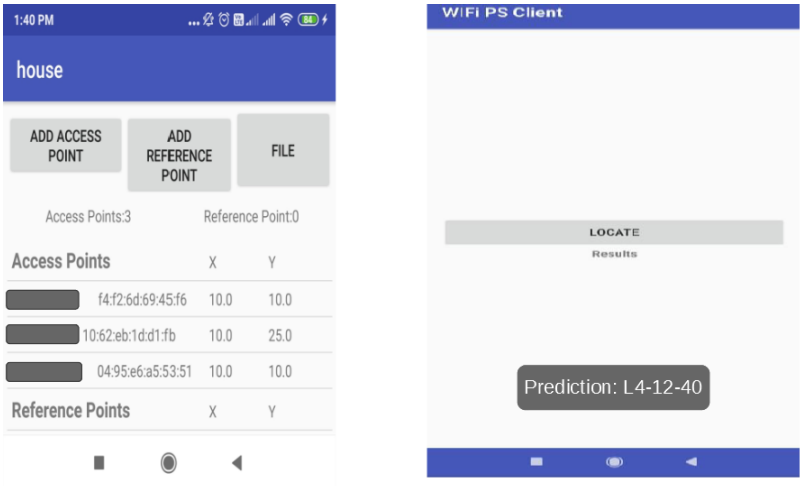
4.2 Experiments and Analysis

In this section, the proposed metaheuristic based AP selection procedure along with the ensemble model is validated against the collected dataset and two benchmark datasets, JUIndoorLoc [71] and UJIIndoorLoc [72] at different granularity.

4.2.1 Experimental Setup

The physical floor into reference coordinates (x, y), which is formed by dividing the floor into virtual grids of 1×1 sq.m.

During the data collection phase, an android application is developed in a manner so that the hotspots can be detected. The front end graphical user interfacne of the application is shown in Figure 4.4. Static registered Wi-Fi APs were considered for localization; the hotspots were detected and removed during the data preprocessing. The details about the dataset can be found in [94].



(a)Data Collection Application (b) Location Prediction Application

Figure 4.4: Graphical view of the fingerprint collection and prediction application

During the preprocessing, missing RSS records are replaced with -110 dBm, a very

low value indicating the APs are to be out of range. The dataset contains readings from four mobile devices (Samsung Galaxy Tab, Moto G, Redmi Note 4, and Google Pixel) collected for a period of 21 days covering a floor of a building of the university. A histogram is depicted in Figure 4.5, which shows an overview of the number of samples per label or grids in the radio map database. The depicted histogram is a combination of all the fingerprints collected using all four devices. It can be observed that the collected dataset has imbalanced class labels. The class imbalance has a direct consequence on the accuracy metric.

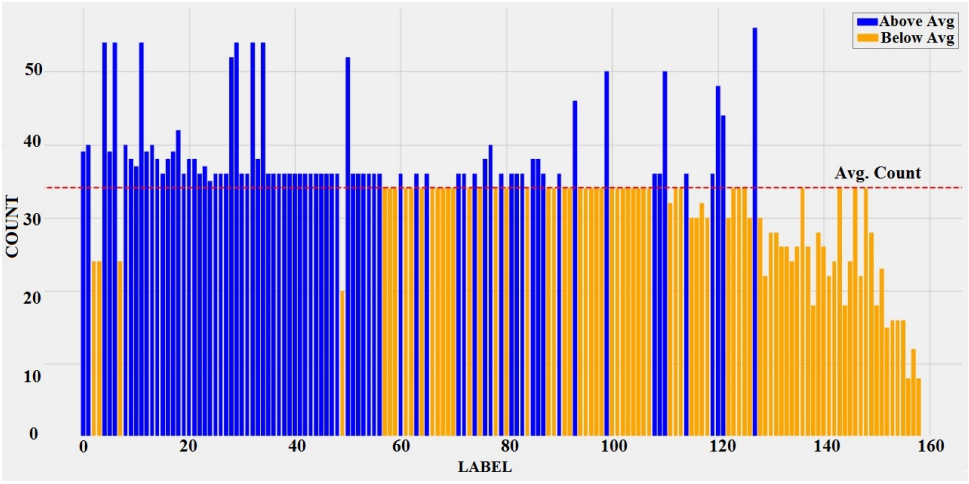


Figure 4.5: Histogram depicting the number of samples per label for the combined device dataset

To tackle the class imbalance problem, *Synthetic Minority Oversampling Technique* (SMOTE) [94, 95] is applied. The procedure iterates through the radio map database to find the minority class labels and generate synthetic samples by utilizing the nearest-neighbor procedure. SMOTE is by far a better approach over random oversampling of minority class samples for lower-dimensional data [96].

Table 4.2: Essential parameters of the fingerprint datasets for experimentation

Dataset	***AP Count***	***Label Count***	***Sample Count***
D1	105	152	548
D2	105	152	1257
D3	105	152	1184
D4	105	152	2442
Combined	105	152	5431
JUIndoorLoc Combined (Floor-4)	172	514	5290
UJIIndoorLoc (Building 1)	520	235	5197

Table 4.2 depicts the salient points about the datasets on which experimentations are performed. The selected set of classifiers on which the performance of the BPSO based feature selection approach is carried out are: k-nearest neighbor(kNN), Decision Tree, Support Vector Machine(SVM), Gaussian Naive Bayes and Artificial Neural Network(ANN).

The experiments have been carried out using the feature-based ensemble approach to explore the potency of the proposed approach by altering the base learners with different classifiers. The meta classifier is a two-layer ([*layer* 1: 70, *layer* 2: 100]) neural network model. The feature-based trained base learners are altered and experimented with kNN, Decision Tree, and ANN. For the experiments conducted with ANN as base learners, a three-layer architecture ([*layer* 1: 70, *layer* 2: 60, *layer* 3: 100]) is used. The weights are inferred through the backpropagation procedure.

The subsequent subsections present the various experiments conducted using collected as well as benchmark datasets with error and performance analysis discussed in Section 4.2.5.

4.2.2 Performance Analysis of BPSO on Collected Radio Map

In this section, the empirical results obtained using the BPSO approach on the collected dataset is reported. The parameter setting of the experiments has been consolidated in Table 4.3. The first experiment indicates that a stable solution is found out through the proposed feature selection procedure as shown in Figure 4.6. Thus, the maximum iteration value taken for the subsequent experiments is 100.

It is essential to investigate the effect of the number of important APs on the localization performance across different state-of-the-art classifiers. From Figure 4.7, it can be observed that with around 30 important APs, the localization performance almost stabilizes for most of the classifiers. For *SVM* classifier, it has been observed that with the use of *polynomial kernel* the accuracy of the classifier increases highly instead of using a *radial bias function kernel*.

Table 4.3: Default parameter setting for the BPSO based feature selection algorithm

BPSO Parameter	***Setting***
Swarm Size(pSize)	50
Maximum Iteration(maxIter)	100
Acceleration variables(c1,c2)	2
Initial Particle Feature	random selection
Velocity initialization	(-2,2)
Weight(W)	1

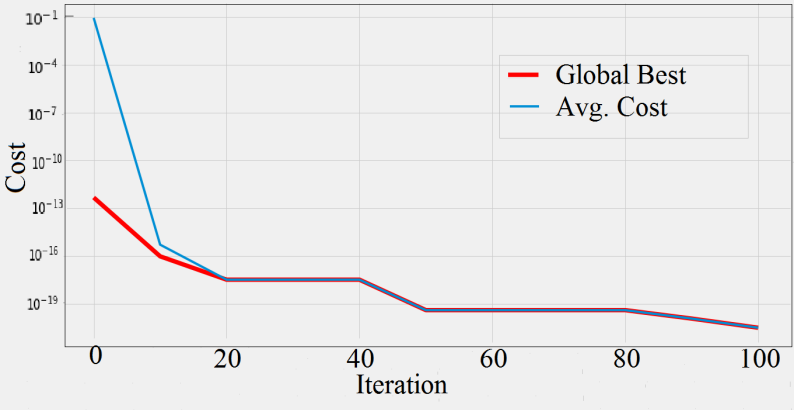


Figure 4.6: Convergence of the BPSO based feature selection approach with the mean fitness

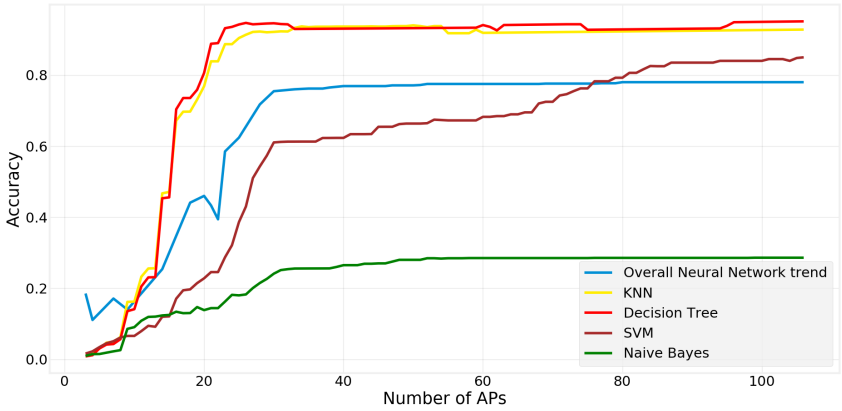


Figure 4.7: The effect of features on localization accuracy

The *Combined* set contains data from all the devices; hence, the data distribution causes the Gaussian Naive Bayes classifier to give less accuracy.

In Table 4.4, the 10-fold cross-validation accuracy has been reported using the selected set of features after the application of the BPSO approach. It can be observed that SVM

and Decision Tree classifiers perform better on the datasets. In order to detail the effect of the feature selection approach on the classifier performance, an experiment is conducted. The results are shown in Figure 4.8 that compares the accuracy values before and after the application of the AP selection process. A rise in classification accuracy for some classifiers has been observed on application of the AP selection procedure. In contrast, for some, the accuracy values remain comparable. Dataset $D1$ is the most imbalanced one among the radiomap datasets. A plot depicting the effects of SMOTE on the BPSO based AP selection is provided in Figure 4.9. The plot shows that the proposed approach can identify the important APs for datasets having both (im)balanced class samples.

Table 4.4: Cross Validation Accuracy after BPSO based feature selection approach

Dataset	***KNN***	***Naïve Bayes***	***SVM***	***Decision Tree***
D1	34.03(±3.7)	68.68(±2)	72.8(±2.9)	73.25(±1.7)
D2	86.6(±3.4)	91.4(±3.8)	84.65(±3.7)	90.8(2.7)
D3	74.23(±2.7)	68.9(±1.2)	92.46(±3.8)	92.08(±3.8)
D4	81.56(±1.4)	30.06(±3)	91.2(±3.3)	91.65(±3.3)
Combined	90.06(±0.22)	18.65(±2.4)	90.72(±2.2)	89.9(±0.223)

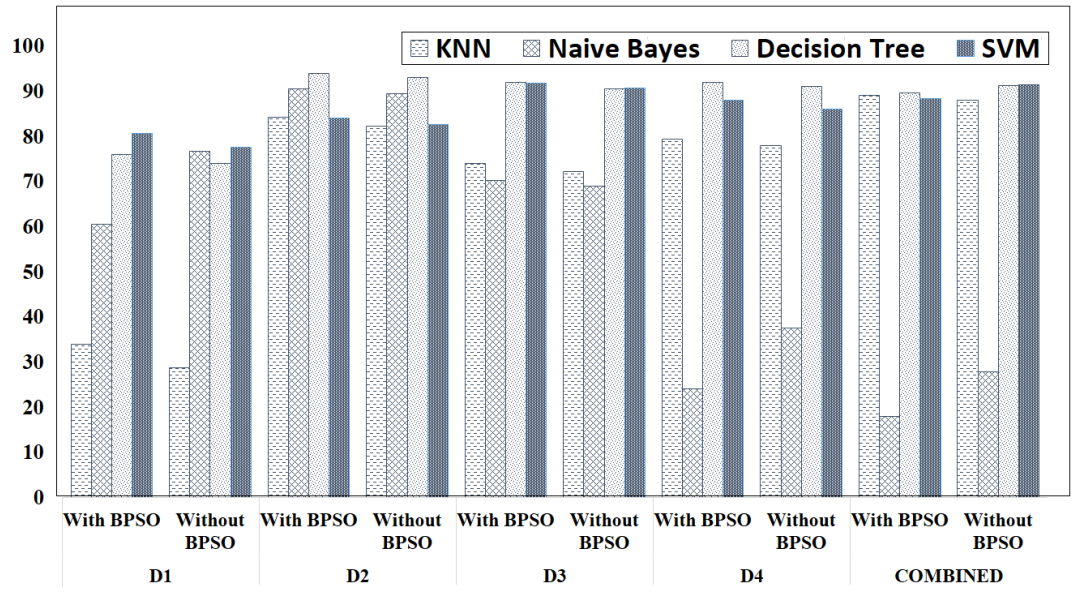


Figure 4.8: Comparison of classification accuracy among selected classifiers before and after applying BPSO based feature selection approach

Figure 4.9: Accuracy on $D1$ Dataset with and without Smote Approach after applying BPSO

4.2.3 Performance Analysis of the Proposed Ensemble Approach

In Table 4.5, a comparative accuracy table among all the selected classifiers (kNN, Naive Bayes, Decision Tree, SVM) with the proposed ensemble approach is presented. It has been observed that even with an imbalanced dataset, the ensemble of feature vectors can retain appreciable localization performance across different classifiers. Performance improves as the dataset contains more fingerprints from different devices (for the *Combined* dataset). A comparative analysis using feature-based majority voting with the proposed training pipeline is presented in Table 4.6. Due to the varying WiFi sensitivity of different devices, the optimal feature sets may not impart similar classification information for a given device. So, the meta classifier, which prioritizes some of the base learners, is found to improve the proposed ensemble's accuracy compared to majority voting.

Table 4.5: Accuracy comparison of the proposed feature ensemble approach (Without SMOTE)

Dataset	*Neural Network*	*KNN*	*Naive Bayes*	*Decision Tree*	*SVM*	*Neural Network Feature Ensemble*	*Decision Tree Feature Ensemble*	*kNN Feature Ensemble*
D1	40.8	28.48	62.4	74.54	70.9	76.53	79.14	45.43
D2	68.9	84.4	90.7	90.89	82.75	88.23	91.51	84.88
D3	70.14	73.23	72.3	91.18	90.46	89.15	92.11	88.73
D4	56.68	79.39	37.51	91.95	88.4	90.81	93.45	88.54
Combined	72.81	90.05	27.9	89.6	91.78	94.28	96.99	94.17

Table 4.6: Comparison of the proposed training pipeline with feature based majority voting approach in terms of localization accuracy

(a) Base Learner Accuracy of Proposed Ensemble Model - Neural Network as Base Learner						
Dataset	***BL 1***	***BL2***	***BL3***	***BL4***	***Majority Voting***	***Proposed Ensemble***
D1	36.92	30.3	40.8	38.48	69.089	74.35
D2	57.56	58.09	64.72	67.9	78.32	87.61
D3	60	70.14	62.54	63.1	80.91	88.14
D4	43.93	55.12	54.71	56.68	75.36	89.9
Combined	70.84	72.81	67.89	60.1	89.65	94.5
(b) Base Learner Accuracy of Proposed Ensemble Model - Decision Tree as Base Learner						
Dataset	***BL 1***	***BL2***	***BL3***	***BL4***	***Majority Voting***	***Proposed Ensemble***
D1	68.2	68.4	68.4	69.09	71.51	76.36
D2	93.8	94.2	93.33	93.36	94.42	97.16
D3	81.4	89.5	90.14	89.29	90.36	91.26
D4	90.17	91.26	91.25	90.07	92.9	94.4
Combined	94.41	97.17	94.22	93.79	96.56	97.11
(c) Base Learner Accuracy of Proposed Ensemble Model - kNN as Base Learner						
Dataset	***BL 1***	***BL2***	***BL3***	***BL4***	***Majority Voting***	***Proposed Ensemble***
D1	28.48	31.51	31.45	27.87	35.65	42.54
D2	83.5	82.7	81.6	83.5	83.41	85.9
D3	72.11	74.36	71.54	71.26	74.64	87.04
D4	78.4	76.53	74.62	75.9	78.3	90.17
Combined	83.91	87.8	85.45	84.77	88.27	94.10

Table 4.7: Comparison of classification accuracy for the proposed feature ensemble approach after applying SMOTE

Dataset	***KNN***	***Naive Bayes***	***Decision Tree***	***SVM***	***Neural Network Feature Ensemble***	***Decision Tree Feature Ensemble***	***kNN Feature Ensemble***
D1	86.84	81.3	95.34	92.24	92.4	95.89	92.33
D2	97.96	97.01	96.91	97.32	95.23	99.89	97.28
D3	91.23	79.4	98.08	95.67	93.15	98.77	97.12
D4	91.45	30.1	97.98	96.14	95.81	98.47	96.54
Combined	94.2	32.4	98.46	96.38	96.28	99.25	98.35

In Table 4.7, the accuracy results after the application of SMOTE is reported. It has been observed that the accuracy values for the kNN classifier have significantly improved. The kNN classifier becomes biased with the minority class samples. This bias is removed on the application of SMOTE. The application of SMOTE does not change the label-specific mean values but decreases the variability of data and induces correlation between the samples. It can be observed not just for the kNN classifier but for the majority of the classifiers; the accuracy has increased. It has also been observed that the Decision Tree used for training the base learners followed by ANN-based meta learner produced the maximum accuracy. The proposed ensemble approach with ANN as base learners has seen minor accuracy modification(Table 4.7), but the accuracy remained consistent with

the datasets.

One of the main objectives of the model is to make it robust and adaptive to changes in the environment and sensitivity of various devices. The following experiments show an accuracy bar plot (Figure 4.10) where the training has been done using two different devices $D3$ and $D4$ collected under different ambient conditions, and the testing is performed using data collected from device $D2$. It can be observed that the proposed ensemble approach is giving appreciable accuracy and can capture different contextual contexts about the environment.

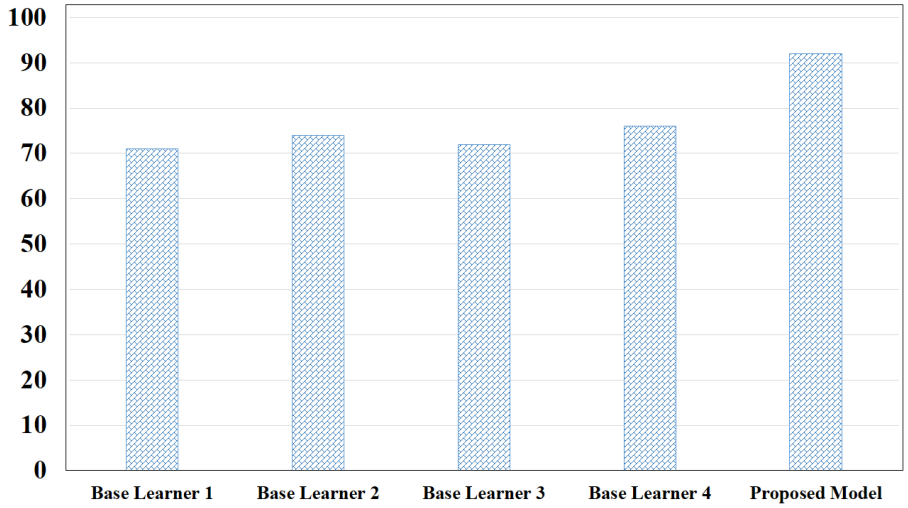


Figure 4.10: Classification accuracy of proposed ensemble (neural network used as base learner), model training performed using device $D3, D4$ and testing done using device $D2$

4.2.4 Analysis on Benchmark Datasets

As mentioned in the earlier part of the section, the UJIIndoorLoc [72], and JUIndoorLoc [71] as the standard benchmark datasets are considered. During experimentation, only building 1 was considered for evaluation in UJIIndoorLoc dataset. For the 2nd Floor of building 1 of UJIIndoorLoc Dataset [72] the accuracy measurements from the classifiers were recorded as shown in Table 4.8(a). It can be observed that the proposed AP selection mechanism is found to select 92 APs out of 520 APs. Notice there is no significant reduction in the accuracy values even though the number of APs has been reduced from 520 to 92. SMOTE used in the pre-processing phase has significantly increased the accuracy values.

Table 4.8: Performance evaluation of the proposed BPSO based feature selection approach on benchmark datasets

(a) Accuracy of different classifiers on UJIIndoorLoc Building 1 Dataset with and without the proposed BPSO based feature selection approach

Approach	***Number of Access Points***	***KNN***	***Decision Tree***	***SVM***	***Naive Bayes***
Without PSO	520	65	70.23	26.17	61.8
With PSO	92	61	69.18	45.3	54.17
With SMOTE (k_neighbors=5) and PSO	92	86.31	88.10	67.14	66.22

(b) Classification accuracy of different classifiers on JUIndoorLoc *Combined* dataset with and without BPSO based feature selection approach

Approach	***Number of Access Points***	***KNN***	***Decision Tree***	***SVM***	***Naive Bayes***
Without PSO	172	80.65	92.56	93.38	77.9
With PSO	57	79.59	94.13	93.25	72.63
With SMOTE (k_neighbors=5) and PSO	57	97.97	97.34	96.69	82.3

The JUIndoorLoc [71] dataset has over 172 APs. On application of the proposed feature selection mechanism, the number of features selected by the BPSO approach is 57. It can be observed from Table 4.8(b) that there is no significant decrease in the accuracy values, and there is a slight increase in accuracy in the Decision Tree-based approach. The SMOTE procedure considered with *neighbors* parameter set to 5 during the pre-processing phase has significantly impacted the training procedure and has contributed to an increase in the accuracy value.

4.2.5 Error and Performance Analysis

The error analysis is carried out by estimating the deviation of the predicted location point from the actual location point. The error deviation metric is evaluated in terms of MAE as defined in chapter 2. Experimentation to investigate Mean Absolute Error(MAE) has been carried out. The results are plotted in Figure 4.11. It can be observed that the MAE is highest for Linear SVM with Polynomial Kernel. The graph (Figure 4.11) is plotted for all the classifiers both with and without the AP selection approach. After the application of the proposed AP selection approach, the MAE has decreased in the case of almost all the selected classifiers. Thus, the error metric has improved significantly. One of the notable decreases can be observed in $D4$ dataset using the Decision Tree classifier where the error has decreased from 4.85m to 3.67m. It can also be noted that the MAE for the Neural Network based feature ensemble lies in the range of $1.5 - 2.5$m, which is giving a better result than the rest of the selected classifiers. A simple path traced on two particular walks is shown in Figure 4.12, showing the actual and predicted values of the

labels.

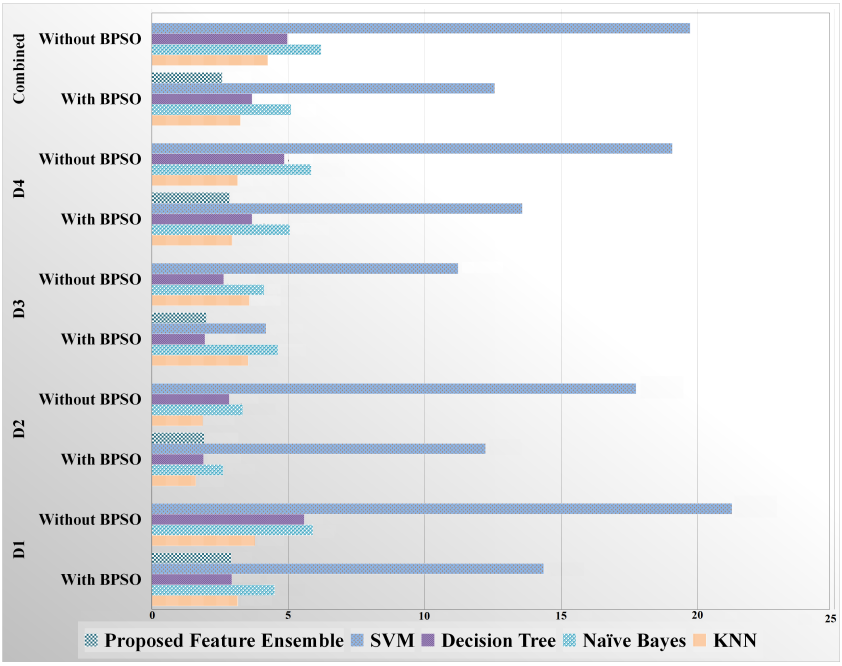


Figure 4.11: Comparison of the proposed feature ensemble model based on MAE (estimated in meters from actual grid)

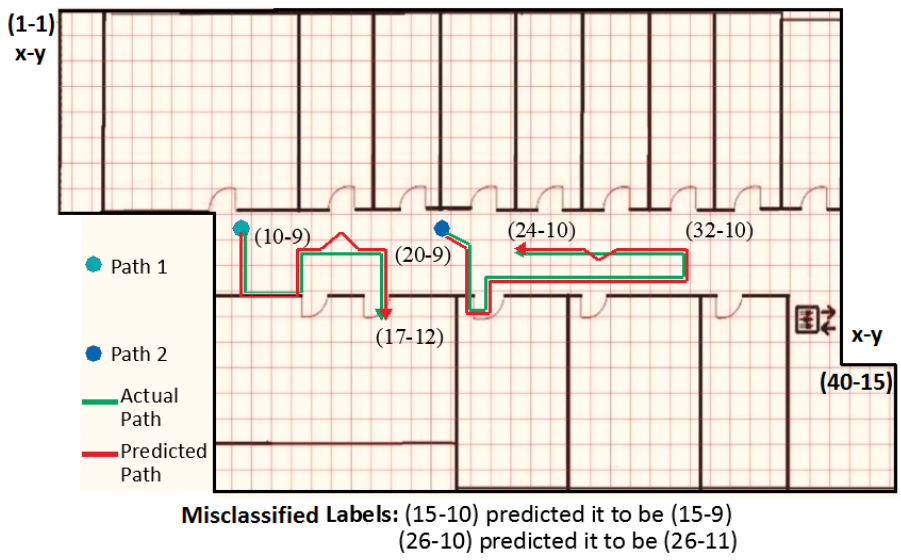


Figure 4.12: Prediction of the labels as a traced path with the proposed ensemble model

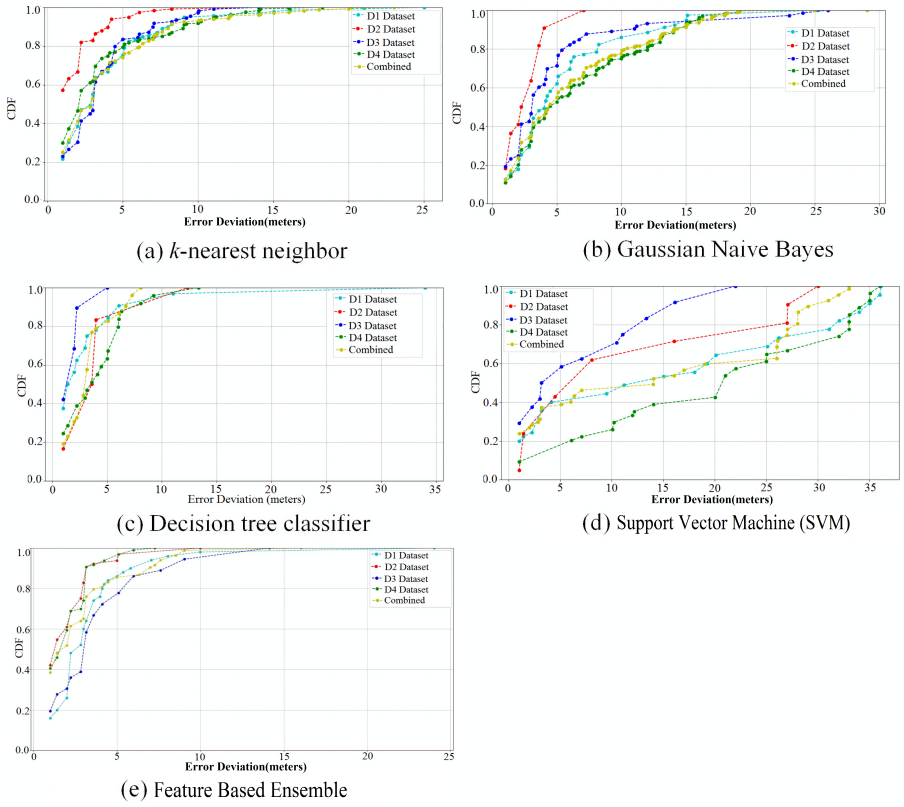


Figure 4.13: Cumulative Distribution Function of positioning errors (in meters)

The Cumulative Distribution Function (CDF) against the error deviations is plotted in Figure 4.13. The CDF gives the probability that the positioning error takes on a value less than or equal to i meters deviation. For the proposed feature ensemble pipeline with ANN as the base learner, the majority of the positioning errors are found to be lying within 3m. The MAE is found to be around 2.68m for the proposed ensemble approach. The CDF drawn shows that more than 85% of the deviations are within $1m - 3m$ for the misclassified instances.

Experimentation by varying the number of base learners and their architecture have been carried out. A snapshot of 3 sets of combinations of base and meta-model can be observed in Table 4.9. It can be observed that on decreasing the base learner count from 4 to 2, the accuracy on the *Combined* device dataset has decreased by 3%. However, the error deviation has increased by 0.24m. On varying the architecture from a $(70, 60, 100)$ combination of hidden layers to $(100, 120, 150)$ there was no significant rise in accuracy

metric, although the MAE decreased by 0.287m. It can be observed in Table 4.7 that the Decision Tree works best for training the feature-based ensemble model. However, a significant decrease in the error metric has been observed when ANN is used as the base classifier.

Table 4.9: Effects of varying the feature based ensemble architecture

Metric	*Combination 1*	*Combination 2*	*Combination 3*
Base Learner Count	4	2	4
Base Learner Architecture	(70,60,100)	(70,60,100)	(100,120,150)
Meta Learner Architecture	(70,100)	(70,100)	(70,100)
Accuracy	96.28	93.15	97.35
Epochs	25	25	40
MAE	2.88m	3.125m	2.539m

A comparison (Figure 4.14) of the proposed feature-based ensemble with some of the popular feature selection approaches like Select-K-Best, PCA, Extra Tree. The plot for the experiment presented is done with ANN as the base classifier for the training process. The graph is plotted on an imbalanced dataset $D1$ and *Combined* device dataset. It has been observed that the proposed approach is performing better in both the selected datasets.

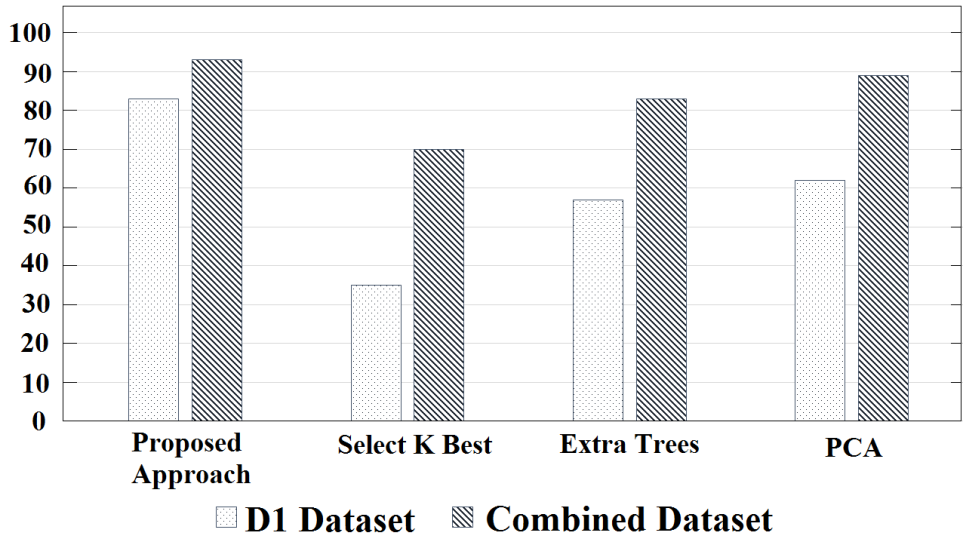


Figure 4.14: Accuracy comparison on application of state-of-the art feature selection approach with the proposed approach (in meters)

A recent publication on clustering based multiple AP selection (MAPS) approach proposed by Huang et al. [37] has been compared with this work. The MAPS approach considers the set of APs for which the RSS values are more than a certain threshold across

all the labels. The results are reported through Table 4.10. It can be observed that the Mean Absoulte error (MAE) is significantly better in this case. It has been observed that 80% of the errors are lying within $2m$ deviation from the actual label. The reason behind that is, the proposed approach emphasizes more on the relationship between the APs. Emphasizing the relationship rather than the information gain helps, the proposed approach overcome device heterogeneity by capturing different pareto-optimal AP sets. These sets are used in creating the feature-based base learners. The comparison of both the approaches has been done on the same radiomap dataset during experimentation for better clarity.

Table 4.10: Performance comparison between two approaches

Approach	Dataset Used	Accuracy	MAE
MAPS [37]	*combined*	93.3	5.63
Proposed Ensemble (neural network as base learner)	*combined*	94.5	2.88

4.3 Summary

The work presents a BPSO based feature selection approach for the AP selection problem to enhance the reliability and make the system adaptive to the dynamic environment.

In indoor localization, there can be no AP that is important throughout the entire experimental region. Thus, depending on the floor plan, some APs cover some regions better, while closely situated APs could be highly correlated and may impart similar knowledge to overall localization performance. However, this correlation factor depends on the ambient conditions and context, such as the WiFi sensitivity of smartphones commonly vary from one device to another.

As the ambient conditions for collecting the test data instances is unknown, it is hard to distinguish between the AP sets having comparable performance as chosen by the various runs of the BPSO based feature selection procedure.Thus, to develop a reliable and adaptive system, it is important to retain the classification model's generality. So, a feature-based ensemble model is proposed in this paper. A brief discussion about the problem definition, selection procedure, and training procedure is presented. Interestingly, it has been found that the BPSO based feature selection algorithm captures the sets

of APs distributed throughout the floor and hence, their range vividly covers the floor map.

The proposed approach has some limitations such as it may fail to generalize well to unseen data or different problem domains, as the it may not adapt effectively to new data distributions or patterns. In the following chapter exploration of the feature extraction and representation of radiomap dataset into lower dimensional encoding with use of Deeplearning approaches has been carried out.

Published Journal:

1. **Panja, A.K.**, Karim, S.F., Neogy, S. and Chowdhury, C., 2022. A novel feature based ensemble learning model for indoor localization of smartphone users. Engineering Applications of Artificial Intelligence, 107, p.104538. *(IF : 8)*

Published Conference:

1. **Panja, A.K.**, Chowdhury, C., Roy, P., Mallick, S., Mondal, S., Paul, S. and Neogy, S., 2021. Designing a framework for real-time wifi-based indoor positioning. In Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020 (pp. 71-82). Springer Singapore.

Chapter 5

Deep Learning Approach for Automatic Feature Engineering

Selecting important APs throughout the experimental region is one aspect of dimensionality reduction problem in the context of indoor localization problem. This is performed through meta-heuristic approaches in the previous chapters. As an AP cannot impart equal knowledge for location prediction for all ambient conditions across the floorplan, ensemble learning models are presented to retain the generality of the solution. In the previous chapter, a neural network based meta classifier have been formulated to capture the importance of each representative context. In this chapter, an attempt to encode the effect of heterogeneous context through deep neural network based models have been carried out. The aim is to transform the dimensionality reduction problem to finding latent representation of the feature space. The concept of instance hardness is also incorporated in order to notify the model about the important data instances.

Deep Learning approaches are becoming popular in the domain of positioning. Deep learning methods are expected to learn a model of the RSS based positioning relationship by properly encoding complex environment factors into its model parameters. Feature extraction is an important aspect in deep learning techniques. Convolutional neural networks(CNN) [49] are widely used in extracting the hierarchical representation of input data. Autoencoder are another form of learning through latent space representation.

Many of past researches have explored the use of Autoencoders in feature extraction process. Xing et al. in [50] have proposed an approach on stacked denoise autoencoding on Hyperspectral images [51]. The high level and sparse level features have been extracted through ReLU activation in the Autoencoding process. The classification process have been carried out using Logistic regression. Meng et al. in [53] proposed a Autoencoder based feature extraction that considers the attributes as well as the relationship into consideration. Kunang et al. in [54] proposed the use of Autoencoding in Intrusion Detection System. The classification is carried out using SVM classifier. The authors claim that their proposed dimensionality reduction is able to retain the relation of information that have been compressed.

In this work, the Convolutional Autoencoding(CAE) process is utilized to extract the features for model training. Temporal locality is one of the important aspect in radiomap dataset. It refers to the measurement of RSS taken at adjacent timestamps that are likely to be correlated. A two-channel representation of input data is formed where one channel holds the normalized radiomap vector and on the other channel holds a k-disagreeing score pertaining to each normalized instances. The autoencoding process is used to represent the 2-channel input dataset into a lower dimensional representation. The model training is carried out using deep 1-D CNN. The Convolutional block can be thought of as a filter that learns to capture important characteristics of the RSS signal. As the input signal is convolved with each layer's filter, the resulting output is a transformed signal that highlights specific patterns and features. The first layer of a 1D CNN can be thought to identify basic signal characteristics such as signal strength and frequency, while subsequent layers might learn to identify more complex patterns such as the presence of walls or other obstacles that can affect the RSS signal. The summarized form of the main contribution is as follows.

- A framework that performs dimensionality reduction through automated feature engineering of Convolutional Deep Learning Model while capturing the temporal dependency of the RSS fingerprints is proposed.
- A two channel representation of input data is formed using the k-disagreeing neighbor

score and normalized RSS dataset.

- The latent space representation is carried out through CAE process. The embedded representation is further used for model training.
- Extensive experimentation have been performed using and performance analysis on benchmark datasets on indoor localization have been carried out.

The chapter is organized in the following manner. First, the basic preliminaries are presented in Section 5.1. Section 5.2 presents the proposed approach and describes the CAE process and the classification model. The Experimental results is discussed in Section 5.3.

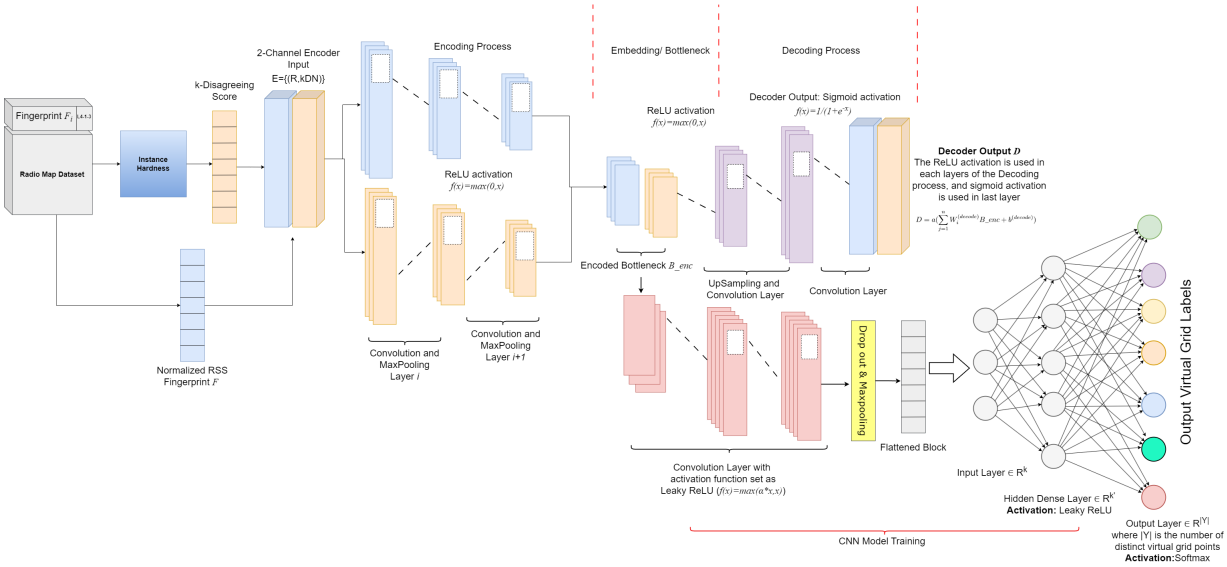


Figure 5.1: The illustration of proposed architecture using 2-Channel 1D Convolutional Neural Network. The 2-channel bottleneck is used for training the Classification Model.

5.1 Preliminary

This section defines the problem definition and input variable mapping to the target problem statement.

5.1.1 Problem Formulation

The radiomap vector $R_{m,n}$ is a 2D vector comprising of fingerprint instances as defined in the previous chapter for a floormap F. The goal of the classification task is to learn a

function $f : R \rightarrow Y$ that maps the input RSS values to the corresponding grid label.

A 1D CNN architecture is proposed here that takes in a 2-channel input, where each channel represents the RSS values obtained from the APs at different time intervals. The input is first passed through a 2-channel Convolutional Autoencoder (CAE) to extract meaningful features, and the bottleneck of the CAE is treated as the transformed feature set for location prediction as detailed in the following subsections.

5.1.2 Normalization

The RSS fingerprints for an ambient context are needed to be normalized so that the internal representation of data could become independent of the ambient variations but mainly to capture the location sensitivity. Applying Standard Scaler(Equation 5.1) normalization to a WiFi RSS (Received Signal Strength) dataset $R_{m,n}$ involves transforming the dataset such that each feature (representing a signal strength measurement) have a mean of 0 and a standard deviation of 1. This ensures that the signal strength measurements are on a common scale, which can prevent overfitting.

$$R' = \frac{R - \mu}{\sigma} \tag{5.1}$$

5.1.3 Hardness Measure

All training instances pertaining to a class may not be equally important for classification. The instances lying at the class centroid and near the boundaries are the ones that contribute most to prediction process. However, in reality in the data collection process, it is natural to have outliers that lie very closely to the boundary instances of a class. Instance hardness is a measure of the importance of an instance that could be indicated to the autoencoder. Thus, the autoencoder would learn about the noisy instances and the probable outliers and thus the dimensionality reduced transformation could be made more precise.

Instance hardness is the measure of how frequently a certain instance is misclassified

by a classification system. The k-Disagreeing neighbor(kDN) kDN_i is the measure of ratio of k-neighboring instances of fingerint F_i that do not share the class boundary.

$$kDN_i = \frac{|F_j|F_j \in kNN(F_i) \wedge Y_i \neq Y_j|}{k} \tag{5.2}$$

The k nearest neighbor instances is measured using the function $kNN(F_i)$ that returns a subset of instances nearer to instance F_i. Y_i is the grid label from where the fingerprint instance F_i is collected. kDN_i stores the k-disagreeing score pertaining each fingerprint instances. The degree of inconsistency in the RSS values for location Y_i can be measured by the kDN score. A high kDN score indicates that there are many nearby data points with different labels, suggesting that the RSS values at location i may be inconsistent or it can be a probable outlier.

5.1.4 Input Channel Representation

A two channel representation have been designed to capture different levels of features and include instance importance.The two channel input vector E (Equation 5.3) is composed of normalized RSS component(*Channel* 1) and the k-disagreeing measure(*Channel* 2). The kDN score is evaluated such that it captures the importance of each instances. By including this information in a separate channel of the CNN, the network can potentially learn to use this information to adjust its weights and biases in a way that is more adaptive to instances with higher or lower hardness scores. In *Channel* 2 the kDN measure is replicated column wise to give same weight to the kDN measure in each column and the normalized RSS values.

$$E = \{(R'^{(N)}, kDN_i)_i\}_{i=1}^{|enc\_rss|} \tag{5.3}$$

5.2 Proposed Approach

In this section, the overall proposed architecture of the training pipeline is presented. The section discusses the Encoding phase using 2-Channel Autoencoder and the subsequent

location prediction phase. Figure 6.3 shows the overall architecture.

5.2.1 Convolutional Autoencoding(CAE) Process

Autoencoding process comprises of encoding and decoding layer. A combination of convolutional layers are used during CAE. The input to the encoding process is a two channel input vector E (Equation 5.3) consisting of normalized RSS component and the k-disagreeing score kDN_i (Equation 5.2). The encoding layers comprises of 1D convolution and Maxpooling layer. The embedded bottleneck is represented in Equations 5.4,5.5. The encoding process is carried out separately for the input two channels. The output of the encoding process is finally concatenated.

$$B_1 = a(\sum_{i=1}^{n} W_i^{(1)} F_i + b^{(1)}) \tag{5.4}$$

$$B_2 = a(\sum_{i=1}^{n} W_i^{(2)} C_i + b^{(2)}) \tag{5.5}$$

$$B\_enc = \{(B_1, B_2)\} \tag{5.6}$$

The concatenated encoded bottleneck B_enc (Equation 5.6) is fed into the decoding layer (Equation 5.7) as the input. The activation function a during the encoding and decoding phase is the Rectified Linear Unit(ReLU)(Equation 5.8). The Leaky ReLU which is a modified version of ReLU is utilised, that allows a small non-zero gradient when the input is negative. High level features are extracted from the input data and introduced into the convolutional layers using the Leaky ReLU activation. The Leaky ReLU is utilised to map the higher level features to the original input space during the decoding phase.

$$D = a(\sum_{i=1}^{n} W_i^{(decode)} B\_enc + b^{(decode)}) \tag{5.7}$$

$$LeakyReLU(x) = \begin{cases} x, & \text{if } x < 0 \\ 0.01x, & \text{otherwise} \end{cases} \tag{5.8}$$

For evaluating how well the input and output of the autoencoder fit together, the Mean Squared Error (MSE) is used. By modifying the weights of the convolutional layers and the decoder layer, the autoencoding method aims to reduce the MSE. The process works by compressing and reconstructing the input data using two sets of convolutional layers in parallel. Equation 5.9 gives the MSE error; here $i \leq n$ is iterated through all the datapoints n and $n\_channel$ represents the number of input channels.

$$MSE = \sum_{i=1}^{n} \sum_{j=1}^{n\_channel} (E_{ij} - D_{ij})^2 \tag{5.9}$$

The goal is to minimize this reconstruction loss, which can be achieved through back-propagation hence, gradient descent. Specifically, the gradients of the loss function can be computed with respect to the weights and biases of the encoder and decoder functions, and update them iteratively to minimize the loss.

The objective is to reduce the loss function $L(B\_enc, D) = MSE$ which is stated as below in Equation 5.10.

$$\min_{\theta} L(E, D)) \tag{5.10}$$

The stochastic gradient descent is used to update the parameters of the encoding and decoding process in the following manner(Equation 5.11).

$$\theta_i = \theta_i - \alpha \frac{\partial L}{\partial \theta_i} \tag{5.11}$$

Here θ_i is the i-th parameter of the encoder or decoder function, α is the learning rate, and $\frac{\partial L}{\partial \theta_i}$ is the partial derivative of the loss function with respect to θ_i.

The gradient is evaluated using the chain rule. To compute the gradients of the loss function with respect to the output of the decoder, the compressed representation, and the input data are stated as follows(Equation 5.12,5.13,5.14).

$$\frac{\partial L}{\partial D} = \frac{2}{n}(E - D) \tag{5.12}$$

$$\frac{\partial L}{\partial B\_enc} = \frac{\partial L}{\partial D}\frac{\partial D}{\partial B\_enc} = \frac{2}{n}(E-D)\frac{\partial g}{\partial B\_enc} \tag{5.13}$$

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial B\_enc}\frac{\partial B\_enc}{\partial \theta_i} = \frac{2}{n}(E-D)\frac{\partial g}{\partial B\_enc}\frac{\partial f}{\partial \theta_i} \tag{5.14}$$

where $\frac{\partial g}{\partial B\_enc}$ and $\frac{\partial f}{\partial \theta_i}$ can be obtained using the backpropagation algorithm.

In practice, mini-batch gradient descent is typically used, where the gradients are computed on a small batch of training samples are computed and the parameters are updated based on the average gradient are updated. This allows for efficient training of the autoencoder on large datasets.

The autoencoder training involves minimizing the reconstruction loss between the input and output data using backpropagation. The goal is to find the optimal parameters of the encoder and decoder functions that preserve the important features of the input data in the compressed representation and thus, this bottleneck layer could signify a compact feature representation of the input signal.

5.2.2 Prediction Model Architecture

The prediction model which is the Autoencoded Compounded Convolutional Neural Network(AECCNN) process is represented as variable WM which is built on the pretext of instance importance. Introducing the kDN measure along with the normalized RSS ensures the manipulation of the weights of learning model with respect to the disagreeing neighbor score. The prediction model for the proposed approach consists of compounded 1D convolutional layers. The input to the model building is the encoded bottleneck B_enc of the Autoencoder layer.

$$WM.fit(B\_enc, Y) \tag{5.15}$$

The tth feature map of the l-th layer $FM_t \in \mathbb{R}$ is given in Equation 5.16; where FM_t is the t-th element of the output tensor, b_t is the bias term for the t-th output feature, s is the stride, f_size is the filter size, $n\_channel$ is the number of input channels, and $W_{j,i,k}$ is the weight of the filter at position j in the filter kernel, for input channel i, and output

feature t.

$$FM_t = A(b_t + \sum_{j=1}^{f\_size} \sum_{i=1}^{n\_channel} FM_{s\cdot(t-1)+j,i} \cdot W_{j,i,t}) \tag{5.16}$$

The activation function is applied element-wise to the output. The encoded 2 channel B_enc after passing through multiple convolutional and pooling layers, the resulting feature maps are flattened into a one-dimensional vector and fed into one or more dense layers. The dense layers are fully connected layers, the convolutional layers. The number of neurons in the dense layers is greater than number of convolutional features ($n\_dense >> n\_conv$) and the activation function used is usually a non-linear function such as ReLU or sigmoid. The dense layers serve to learn complex, non-linear relationships between the extracted features and the grid labels.

5.3 Experimental Result

The section discusses the floormap and the radiomap datasets on which the experiments were conducted. Comparative error deviation plots have been showcased with other DNN models and benchmarks datasets. A comparison of the proposed approach with state-of-the-art Feature transformation and selection approach have also been visualized.

5.3.1 Experimental Data

During the experimentation the collected *combined* dataset. For effectively assessing the effectiveness of the proposed approach beyond the collected dataset, three benchmark datasets have been considered, namely, JUIndoorLoc [71], UJIIndoorLoc [72] and Shopping Mall dataset [97]. The sample count and feature count of all the datasets are represented in table 5.1.

Table 5.1: Overview of sample count and feature count in the experimental datasets

Dataset	Instance Count	AP Count
Combined(D1,D2,D3,D4)	*5428*	*108*
JUIndoorLoc	*6639*	*172*
UJIIndoorLoc	*9493*	*520*
Shopping Mall	*3293*	*55*

5.3.2 Result Analysis

In this section, experiments are conducted for the *Collected-combined* dataset. Two performance metric which is the localization accuracy and the Mean Absolute Error(MAE) are considered. The subsequent subsection presents the performance comparison with other Deep Learning Models followed by experimentation on benchmark datasets.

5.3.2.1 Localization Accuracy and Error Evaluation

An experiment was conducted to find the training vs validation accuracy on *Collected-combined* dataset using the 1-D-AECCNN with 2 channel input. The results are plotted in Figure 5.2. The model architectural parameters are presented in table 5.2. The plot indicates the effectiveness of parameter tuning and reduced probability of overfitting as the validation accuracy and training accuracy are found to be comparable.

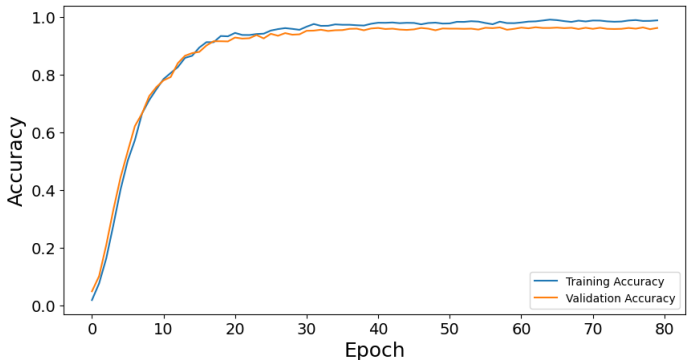


Figure 5.2: Training Accuracy vs Validation Accuracy Plot of 1D-AECCNN

Table 5.2: Overview of Parameter settings 1D-AECCNN.

Parameters	***AutoEncoding Block***	***Encoded Training Block***
Layers with Filter Size	*Encoding:[Conv1D,Maxpooling], [Conv1D,Concatenate]* *Decoding:[Upscaling,Conv1D, Conv1D,Conv1D]*	*[Conv1D,Conv1D,Conv1D,Conv1D, Dropout(0.5),Maxpooling1D, Dropout(0.5)] [Dense(256), Dense(1024)]*
Optimizer	*Adam*	*Adam*
Epochs	*10*	*60*
Filter Size	*[64][64],[64,64,108]*	*[32,64,128,224]*
Activation Function	*ReLU(Conv1D blocks), sigmoid(Decoder output)*	*LeakyReLU(Conv1D and dense blocks), Softmax(output layer)*
Learning rate	*0.01*	*0.001*
Kernel Size	*3*	*3*
Padding	*same*	*same*

Table 5.3 gives a comparative analysis between different ANN and selected CNN models. Three different dense models have been used with the following layer combination:*[128,64,32],[256,128,64,32] and [256,1024].* It can be observed from the table that 1D-CCNN with 2 channel representation is giving the best result. It can also be observed that 1D-AECCNN is performing at par with 1D-CCNN(2 channel). Apart from accuracy, the MAE metric plays an important part in evaluating the performance of the model. It can be observed that, on using 1D-CCNN(2channel) the error deviation is 2.43m which is quite acceptable for localization applications. Thus, incorporation of 2 channels is found to improve the localization performance as kDN provides an effective insight into the nature of the instance. Interestingly, incorporation of CAE, the dimensionality gets reduced without much reduction in accuracy.

A study on the effects of altering the layer configuration of CAE of the proposed AECCNN model have been carried out. The performance of 6 such configuration have been evaluated whose details are reported in Table 5.4. Figure 5.3 gives the error deviation in meters from all the 6 CAE architectures. It can be observed that a more precise latent space representation is obtained with filter configuration set to 64 with LeakyReLU as the activation function. The hidden embedding matrix generated by the Autoencoder is the learned representation of the input data in the latent space. The designed Autoencoding process is carried out in a way that the input features of the Autoencoder is directly dependent on the embedding dimensions. Experiments carried out by selectively increasing the size of the filter, updating the activation parameter and also modifying the dropout parameter. It can be observed from the figure that AECCNN-M1 with filter size set to 64 is giving the best result with MAE of $2.37m$. The CDF was measured as it signifies statistical stability/significance of the results.

Table 5.3: Accuracy comparison between ANN and 1D-CNN architectures

Metric	**Dense M1**	**Dense M2**	**Dense M3**	**1D-CCNN (1 Channel)**	**1D-CCNN (2 Channel)**	**1D-AECCNN (2 Channel)**
Accuracy	*89.26*	*90.15*	*91.25*	*94.43*	*97.12*	*96.56*
MAE(metre)	*3.16*	*3.05*	*3.006*	*2.89*	*2.43*	*2.55*
Dense Layer Neurons	*128,64,32*	*256,128,64,32*	*256,1024*	*256,1024*	*256,1024*	*256,1024*
Activation	*ReLU*	*ReLU*	*LeakyReLU*	*LeakyReLU*	*LeakyReLU*	*LeakyReLU*
Optimizer	*Adam*	*Adam*	*Adam*	*Adam*	*Adam*	*Adam*

Table 5.4: Different layer configuration of Convolutional Autoencoder(CAE) used in experimentation

AE Model	Encoder Channel 1 Layers	Encoder Channel 2 Layers	Decoding Layers	MAE (meters)
Autoencoder -M1	*Conv1(filter=64,kernel=3,stride=1), BN-1(0.99),LeakyReLU-1(0.0015), Dropout-1(0.1), Maxpooling-1(2)*	*Conv1(filter=64,kernel=3,stride=1), BN-1(0.99),LeakyReLU-1(0.0025), Dropout-1(0.1), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=64,kernel=3,stride=1), BN(0.99),LeakyRELU(0.0015),Dropout(0.1), Conv1(filter=64,kernel=3,stride=1), BN(0.99), LeakyReLU(0.0015), Dropout(0.1)*	*2.37*
Autoencoder -M2	*Conv1(filter=128,kernel=3,stride=1), BN-1(0.99),LeakyReLU-1(0.0075), Dropout-1(0.15), Maxpooling-1(2)*	*Conv1(filter=128,kernel=3,stride=1), BN-1(0.99),LeakyReLU-1(0.0075), Dropout-1(0.15), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=128,kernel=3,stride=1), BN(0.99),LeakyRELU(0.0075),Dropout(0.15), Conv2(filter=128,kernel=3,stride=1), BN(0.99), LeakyReLU(0.0075), Dropout(0.15)*	*3.6886*
Autoencoder -M3	*Conv1(filter=256,kernel=5,stride=1), BN-1(0.75), LeakyReLU-1(0.015), Dropout-1(0.2), Maxpooling-1(2)*	*Conv1(filter=256,kernel=5,stride=1), BN-1(0.75),LeakyReLU-1(0.015), Dropout-1(0.2), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=256,kernel=5,stride=1), BN(0.75), LeakyRELU(0.015),Dropout(0.2), Conv1(filter=256,kernel=5,stride=1), BN(0.75), LeakyReLU(0.015), Dropout(0.2)*	*4.2238*
Autoencoder -M4	*Conv1(filter=32,kernel=5,stride=1), BN-1(0.75),LeakyReLU-1(0.15), Dropout-1(0.1), Maxpooling-1(2)*	*Conv1(filter=32,kernel=5,stride=1), BN-1(0.75),LeakyReLU-1(0.15), Dropout-1(0.1), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=32,kernel=5,stride=1), BN(0.75), LeakyRELU(0.15),Dropout(0.1), Conv1(filter=32,kernel=5,stride=1), BN(0.75), LeakyReLU(0.15), Dropout(0.1)*	*4.6446*
Autoencoder -M5	*Conv1(filter=128,kernel=1,stride=1), BN-1(0.99),LeakyReLU-1(0.025), Dropout-1(0.1), Maxpooling-1(2)*	*Conv1(filter=128,kernel=1,stride=1), BN-1(0.99),LeakyReLU-1(0.025), Dropout-1(0.1), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=128,kernel=1,stride=1), BN(0.75), LeakyRELU(0.025),Dropout(0.1), Conv1(filter=128,kernel=1,stride=1), BN(0.75), LeakyReLU(0.025), Dropout(0.1)*	*5.6372*
Autoencoder -M6	*Conv1(filter=512,kernel=7,stride=1), BN-1(0.99),LeakyReLU-1(0.025), Dropout-1(0.1), Maxpooling-1(2)*	*Conv1(filter=512,kernel=7,stride=1), BN-1(0.99),LeakyReLU-1(0.025), Dropout-1(0.1), Maxpooling-1(2)*	*Upsample(2),Conv1(filter=512,kernel=7,stride=1), BN(0.75), LeakyRELU(0.025),Dropout(0.1), Conv1(filter=512,kernel=7,stride=1), BN(0.75), LeakyReLU(0.025), Dropout(0.1)*	*2.65*

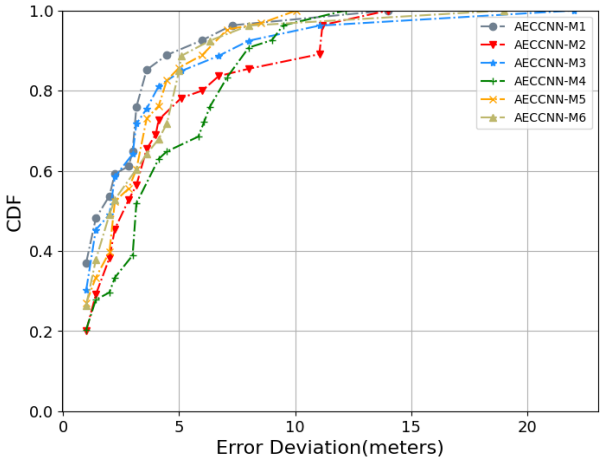


Figure 5.3: Error deviation CDF plot of different CAE architectures. The model training architecture of all the CAE is same as Table 5.2.

5.3.2.2 Comparison with other DNN Frameworks

The proposed model AECCNN representation of 2-channel input and training from the bottleneck output of the Autoencoding process have been tested against state-of-the-art Deep Neural Network models(DNN). The considered DNN models are AlexNet [98], ZFNet [99] and a residual model [100]. The AlexNet is a convolutional neural network (CNN) architecture consisting of eight layers, including five convolutional layers and three fully connected layers. ZFNet architecture is an extension of AlexNet, which is characterized by its specific architectural details, such as increased filter sizes and strides in the initial layers and additional deconvolutional layers. The ResNet architecture is kept same as the

proposed CNN architecture with 3 dense layers and 4 convolutional layers. The *Residual blocks* are made to ascertain the exact mapping from the input to the output, by acquiring the knowledge of the residual, or the discrepancy, between the two. All the DNN models have been iterated for 60 epochs and have been tested using both 1 and 2 channel input.

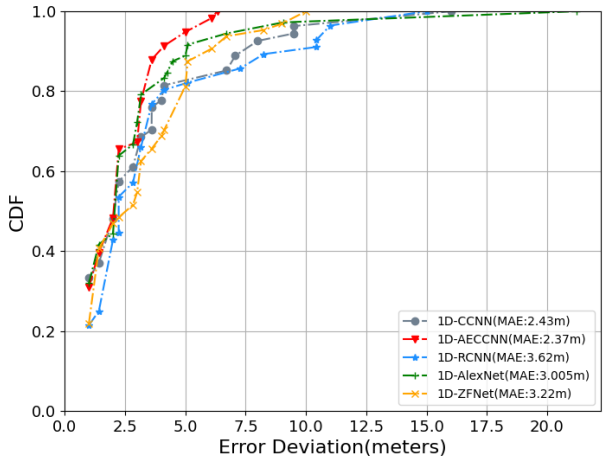


Figure 5.4: Error Deviation CDF plot during testing phase from 5 selected Deep Learning (2-channel) architectures on Collected-combined dataset. The error deviation is calculated in metre level.

Table 5.5 gives the accuracy and precision metric evaluated on the experimentation conducted on Collected-combined dataset. It can be observed that in all the cases the 2-channel input (Normalized RSS, kDN) is performing better than 1-channel input. A CDF plot(Figure 5.4) on the error deviation is evaluated using considered DNN models. The CDF visualization is carried out on 2-channel input. It can be see that 1D-AECCNN and 1-D ZFNet have majority of their misclassified instances within 5 meter deviation.

Table 5.5: Accuracy Comparison between selected set of model architectures on Collected-combined Dataset. The models have been iterated for 60 epochs.

Model	Channel	Accuracy	Precision
1D-CCNN	*1*	*94.43*	*95*
1D-CCNN	*2*	*97.12*	*97*
1D-AECCNN	***2***	***96.56***	***97***
1D-RCNN	*1*	*96.56*	*97*
1D-RCNN	*2*	*95.4*	*95*
1D-Alexnet	*1*	*92.4*	*93*
1D-Alexnet	*2*	*94.41*	*95.1*
1D-ZFNet	*1*	*93.2*	*94.5*
1D-ZFNet	*2*	*94.3*	*95*

Table 5.6: Comparison of Accuracy and MAE with benchmark dataset-Shopping Mall, UJIIndoorLoc and JUIndoorLoc

	Shopping Mall			JUIndoorLoc			UJIIndoorLoc		
Metric	1D-CCNN (1Channel)	1D-CCNN (2Channel)	1D-AECCNN (2Channel)	1D-CCNN (1Channel)	1D-CCNN (2 Channel)	1D-AECCNN (2 Channel)	1D-CCNN (1Channel)	1D-CCNN (2 Channel)	1D-AECCNN (2 Channel)
Accuracy	89.46	92.65	93.25	88.88	92.88	92.75	63.23	66.47	66.25
MAE	12.95	11.56	11.58	3.82	2.95	2.65	13.85	12.6	12.85

5.3.2.3 Comparison with Benchmark Dataset

The UJIIndoorLoc, JUIndoorLoc and Shopping Mall (published by Microsoft research) have been considered for performance evaluation. The 2 channel model 1D-AECCNN and 1D-CCNN have been used for this experimentation. The accuracy and MAE are reported in the table 5.6. It can be observed that the error deviation on JUIndoorLoc dataset is the least with meter level deviation ranging from 2.65-2.95m. Through experimentation it have been observed that the performance of 2-channel is better than 1-channel architecture.

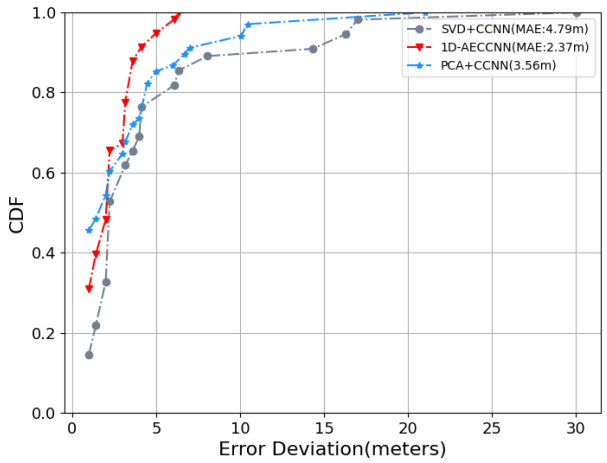


Figure 5.5: Comparison of proposed approach error deviation CDF plot with SVD and PCA approach.

5.3.2.4 Comparative study of dimensionality reduction methods

In this section, a comparison of performance of the proposed approach 1D-AECCNN is compared with two commonly used dimensionality reduction approaches like Principal component analysis (PCA) [101] and Singular value decomposition(SVD) [102]. The classification accuracy of both the approaches are evaluated for the collected dataset. PCA works by capturing the orthogonal components of the fingerprints and captures the maximum variance. SVD works on the principle of matrix factorization containing three

components: U, $\sum$, and V, where U and V are orthogonal matrices and $\sum$ is a diagonal matrix of singular values. During experimentation the Explained Variance Ratio(EVR) attribute in PCA or SVD model was evaluated. The EVR indicated how much information is captured by each component or vector relative to the total amount of variance in the collected RSS dataset. It was observed that selecting components in the range of 40-50 yield the best performance for both the approaches. With training performed using the proposed Compounded CNN(CCNN) model, it was observed that PCA+CCNN and SVD+CCNN yield accuracy of 91.4% and 92.3%, respectively. Figure 5.5 gives the insight into the error deviation. It can be clearly observed that PCA+CCNN had an error deviation of 3.56m and SVD+CCNN with 4.79m which clearly indicates that the proposed dimensionality reduction approach is performing better both with respect to accuracy as well as error deviation.

5.4 Summary

In this work, the effects of dimensionality reduction on radiomap dataset through multiple channel input have been studied. The proposed CAE architecture is able to effectively reduce the dimensionality and the subsequent CNN model is capable of performing localization without a decrease in accuracy metric. The CAE model AECCNN-M1 have achieved the maximum reduction in reconstruction error against all the experimented architectures.

The performance of the CNN model is affected by whether the kDN score is present in the input data. By adding the kDN score as an additional input channel, the model may train to recognise the relationship between the kDN score and the device position. The device location estimation may be subject to more uncertainty when the kDN score is high, which denotes that the RSS data from several APs is inconsistent. The fingerprint could be an outlier or have an inconsistent RSS measurement. The model can learn to assign less weight to the uncertain data pieces and produce more accurate location predictions by integrating the kDN score. It is clearly observable from the output that kDN measure used in one channel is capable of reducing the error deviation by 0.8-1.2m. Furthermore, the accuracy of localization model have also increased with the use of 2-channel input.

The reduced feature map output from the autoencoding block is performing at par with the normal 2-channel architecture. The same holds by altering the model architecture with other DNN models.

The output benchmark datasets JUIndoorLoc, UJIIndoorLoc and Shopping Mall dataset have been used to evaluate the performance. It can be observed that the proposed pipeline is effective in reducing both the error deviation and accuracy metric as compared to 1D-CCNN(1 Channel).

While feature selection and extraction as discussed in this chapter and the preceding chapters are crucial steps in dimensionality reduction analysis, but the process falls short in capturing the intricate nuances of indoor localization using WiFi access points. Hence the subsequent chapter necessitates to explore the domain of the instance selection and instance based learning.

Communicated Journal:

1. **Panja, A.K.**, Biswas, S., Neogy, S., Chowdhury, C. Dimensionality Reduction through Multiple Convolutional channels for RSS based Indoor Localization. Submitted to IEEE Transaction on Artificial Intelligence.

Chapter 6

Meta-heuristic based Instance Selection Approach

The previous chapters delved into the intricate world of feature selection and transformation. Another pivotal aspect of the work is to decrease the influence of noise and ensuring localization in constrained environment with limited storage. This is catered by understanding instance importance and selection. Thus, the aim of this work is to investigate the problem of instance selection from the perspectives of indoor localization. During the Fingerprint collection phase, redundant and noisy instances also get included in the training set. Selection of proper *fingerprint instances* is important for the performance of the model. Furthermore, the size of the dataset and number of data samples play an important role during edge level training process. For better sustainability and real time performance, localization algorithms are needed to be executed closer to the device (such as, the edge) rather than at a cloud server connected over the Internet. This calls for deployment of a lightweight model at the edge requiring a good number of well-distributed samples that better defines the class labels. Hence, instance selection [103] is a plausible solution during the pre-processing phase of the *fingerprint* dataset.

Tomelinks [104], *Near Miss* [105], and *Condensed Nearest Neighbor (CONN)* algorithms [106] are some of the most advanced instance selection techniques. *Clustering* based approaches are also widely explored in the domain of instance reduction. Ougia-

roglou et al. in [58] proposed a reduction method through a homogeneous cluster. Another work that discusses retaining class border points and selectively removing points closer to a cluster's center is proposed in [107]. Though the problem is very relevant for developing edge level solutions for WiFi based indoor localization, it is hardly explored in the literature. This forms the motivation of the work.

In the domain of the smartphone RSS-based indoor positioning, the instance selection problem gets a new dimension as such approaches involve the consideration of environmental context as well as device context. This is crucial as instances collected during two different time frames at one location point might exhibit varying characteristics pertaining to their WiFi RSS fingerprint. A t-SNE plot can help in visualizing the radiomap dataset. As evident from the t-SNE plot visualization(Figure 6.1) the class boundaries may vary depending on different ambient context. Thus, in the selection process, it is important to

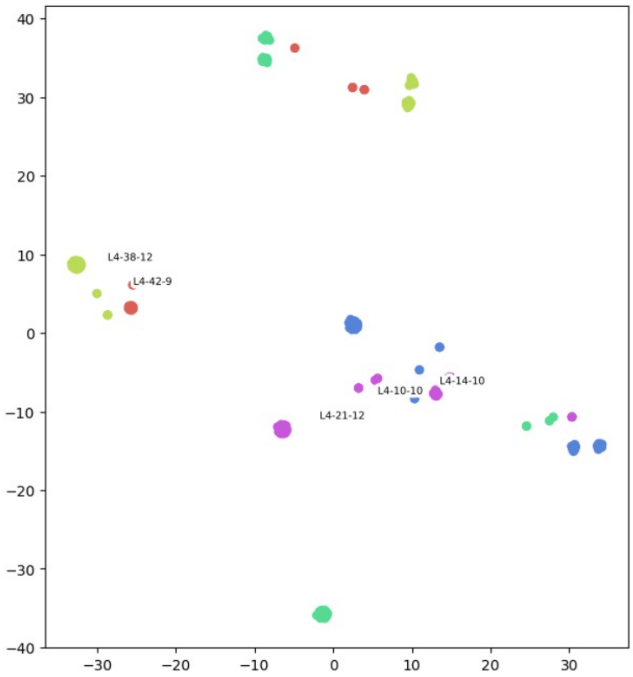


Figure 6.1: t-SNE plot visualization on Collected combined dataset

retain such instances pertaining to varying context. *Metaheuristic* [108] based approach can search over a large set of practical solutions to find near-optimal solutions which in this case is the selection of proper instances that aptly defines the class labels. In this

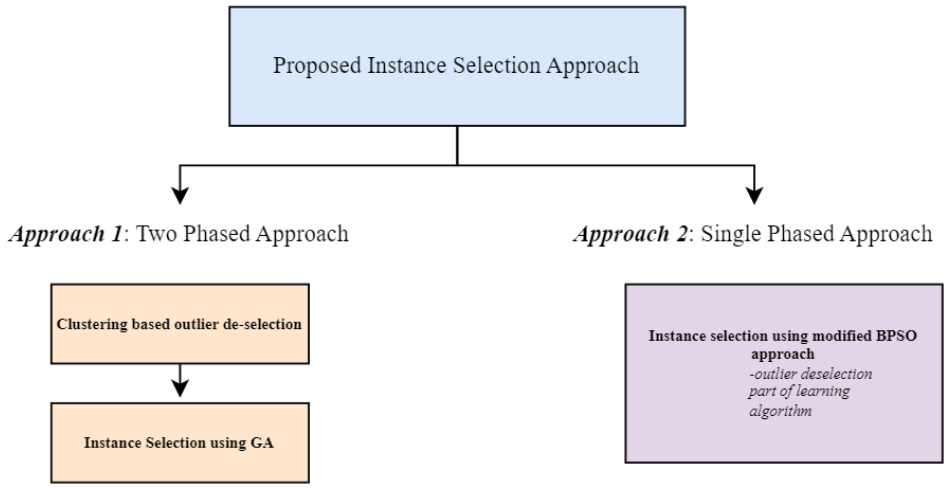


Figure 6.2: Summary of the approaches showcased

chapter, the instance selection problem proposed has been carried out using metaheuristic based approach. A two phased and a single phased approach has been proposed. For the two phased approach, in the first phase class centroids are detected first, followed by a KDTree based deterministic approach to reduce the outliers. The stochastic selection and replacement of instances are done to reduce the redundancy and noisy instances. In the second phase a meta-heuristic approach has been adopted to further optimize the class boundaries by focussing on instance hardness. Here, genetic algorithm is chosen as the meta-heuristic approach and for evaluating instance hardness the kDN measure is considered.

Later in the single phased approach a hybrid modelling of BPSO and *k-differentiating neighbor*(kDN) assessment for the instance selection process have been formulated. The proposed approach aims at selection of instances as well as detection and de-selection of an outlier at each local level. Here, outlier deselection has been inserted in the exploration phase of BPSO that utilizes the concept of kDN.

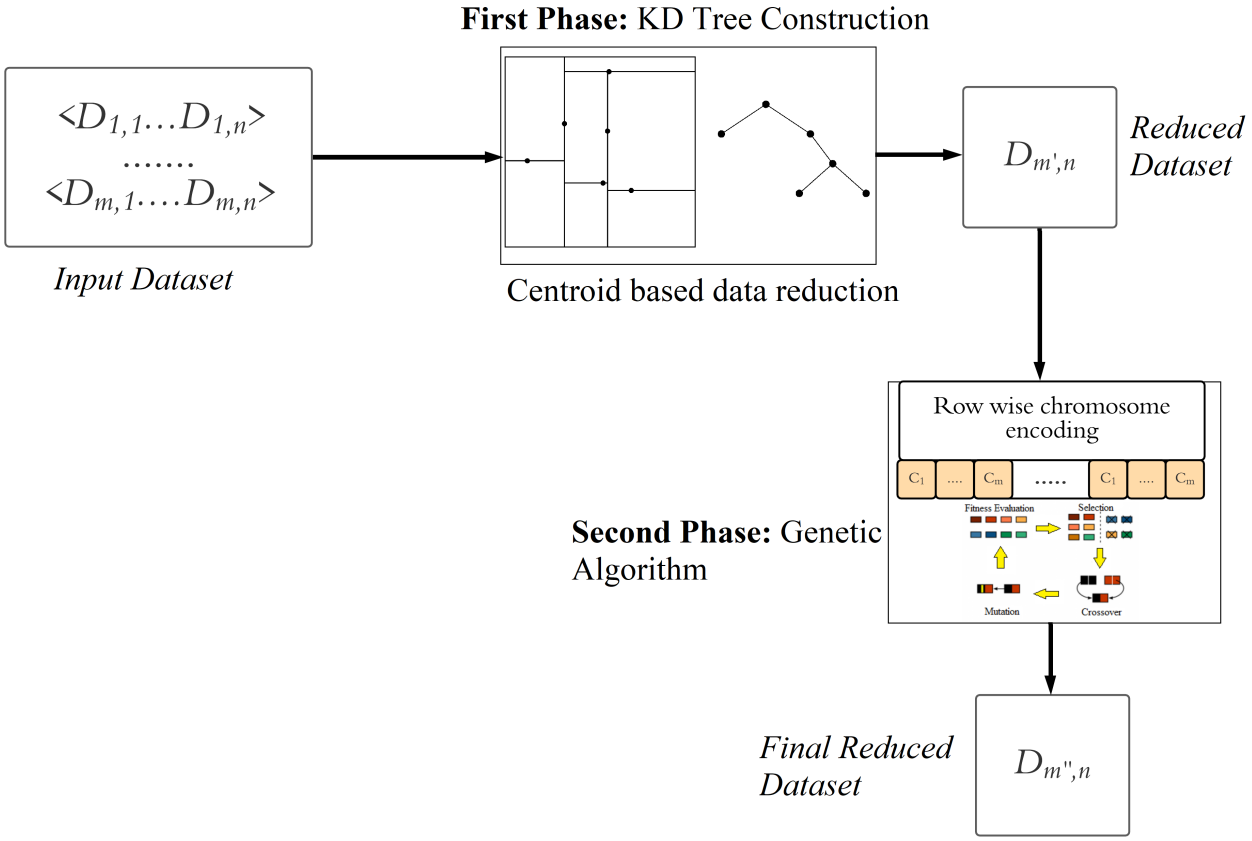


Figure 6.3: Framework of two phase instance selection pipeline

6.1 Approach 1:Two Phase Approach

The proposed instance selection procedure takes place in two phases. In the first phase, a centroid-based clustering approach has been designed that takes care of the data distribution per activity class. The next part of the pipeline is designed by applying the Genetic Algorithm. Supervised learning-based classification is incorporated to measure the effectiveness of the approach. To ascertain the statistical significance of the outcome, cross-validation has been performed. The overall framework is shown in Figure 6.3.

6.1.1 First Phase Reduction Pipeline

The objective of this phase is to mainly identify the outliers using the centroid-based method. The procedure starts by constructing a KD-Tree on the entire dataset. KD-tree is a space-partitioning tree that uses Depth-First Search (DFS) tree traversal algorithm

followed by backtracking. In the very process of KD-tree construction, hierarchical indexes of hyper-rectangles are constructed which are an axis-aligned partition of the space constructed using the dataset. KD-tree works by partitioning point sets recursively along with different features or dimensions. Every node in the tree is defined by a plane along with one of the dimensions that partitions the dataset.

The generated children from the parent nodes are regions that are again divided into equal partitions, using planes along other different dimensions. The searching process takes place along one of the nearest and non nearest planes and through DFS and backtracking methods.

Definition 1. *An Instance $I_i = \{D_i, A_i\}$ is represented using tuple D_i and class label A_i; where $1 \leq i \leq m$ and n features which is encoded as a subset of $\mathbb{R}^n$. A KD-Tree is formalized by the function $KDT(D, m, n)$; with median splitting having height $O(log(n))$.*

Definition 2. *A query $q_{I_r,k}$; where $r \in m$ is a nearest neighbour query that maps a given attribute/data value D_r of instance I_r to k nearest data tuples; $k \in \mathbb{N}(k \geq 1)$. The output t is a set that is estimated by a distance function d as follows:*

$$t = \arg\min_{r \leq m} d(q_{I_r,k}, D_r) \tag{6.1}$$

t is a set containing k closest distant points to tuple D_r.

Definition 3. *A point replacement function $ReplacePT(t, D_x)$ replaces the tuples in t with tuple D_x; $x \leq m$.*

The replacement procedure(Algorithm 4) starts by selection of D_r random points. From the tree, a query $q_{I_r,k}$ is evaluated(line no 9); where k nearer points to point D_r is searched and selected. A centroid c of all the $k+1$ points is evaluated (line no 10). In line no 11, point D_x from the evaluated nearest neighbor set close to the centroid c is estimated. All the $k+1$ points are removed and replaced by a single point D_x(line no 12). The replacement is iteratively carried out until the threshold condition ($(mAcc - tempAcc)$

$\leq \beta$) holds. $mAcc$ is the i-fold cross-validation accuracy, $tempAcc$ is the cross-validation accuracy of the intermediate reduced dataset and β is the small threshold value. The iteration continues and in each iteration, a new KD-tree is constructed unless comparable cross-validation accuracy has been obtained to indicate sustainable performance. Effectively, all the smaller and larger cluster distributions are retained through this mechanism. The inter-class and intraclass clusters can also be identified through this procedure as it does not assume one cluster per activity class.

Algorithm 4: nearestCentroid(D, m,n)

Result: Reduced Dataset $D'_{m',n}$

$D_{m,n}$ - Input Dataset of dimension m x n
$mAcc \leftarrow modelAccuracy(D)$
$tempAcc \leftarrow mAcc$
$D' \leftarrow D$
Initialize threshold t
Construct KDTree $KDT(D', m, n)$
while $(mAcc - tempAcc) \leq \beta$ **do**
    D'_r=rand(D');select random instance
    $t \leftarrow q_{I_r,k}$; selecting k points nearer to D'_r into a vector t
    $c \leftarrow \sum_{i=1}^{k} \frac{t_i}{k}$; evaluate center element
    Select point D_x such that $d(c, D_x) < d(c, \{t\} - D_x)$
    $ReplacePT(t, D_x)$; Replace the k points with the point closer to centroid c and update D'.
    $m' \leftarrow m' - k$
    $tempAcc \leftarrow modelAccuracy(D')$
end
return D', m'

6.1.2 End phase pipeline using point-wise GA approach

In this section, discussion on how GA is designed for selecting the best instances to further optimize the total dataset for training and execution in constrained environments.

6.1.2.1 Proposed GA based selection

The objective of this phase is to achieve undersampling by accepting a minimal reduction of accuracy. The dataset has already been undersampled using the centroid method which constitutes the first phase of the pipeline. The undersampled dataset is passed through the proposed GA-based approach to select a global set of input instances for which the

accuracy is comparable to the accuracy attained when the model is trained using the whole dataset. The reason behind using genetic algorithm later in the pipeline is to achieve better prediction. This is done by strategically exploring the solution space by tailoring the fitness function to the problem at hand and introducing a factor of randomness through mutation and crossover probabilities.

Table 6.1: Terminologies used in defining the proposed two phase tuple selection approach

Notation	Description
$D'_{m',n}$	*Dataset after first phase of pipeline*
m'	*Total no of instances; with respect to chromosome encoding it represents the number of genes*
$pSize$	*Chromosome Population size*
Ch_i	*Chromosome Object i*
n	*No of shuffled training set*
r	*Fraction of population selected for crossover.*
$maxGen$	*Maximum no of times GA is iterated*
$genC$	*Current generation of the procedure*

6.1.2.2 Encoding Scheme

Let $D'_{m',n}$ be the undersampled dataset received from the first part of the pipeline. The objective is to select the global best set of the dataset for which the difference of accuracy is within the threshold limit. As the number of instances is m' thus the search space is $2^{m'}$. The search space can be represented by m' genes into a binary chromosome vector. Each gene represents the selection of the instance. The chromosome vector is represented with the variable $Ch_i.sel_{m'}$ where $1 < i \leq pSize$ which is the population size or the number of chromosomes.

6.1.2.3 Fitness Function and Selection

The chromosomes hold the selected set of tuples. The fitness function is evaluated, and the chromosomes which satisfy the fitness function are considered for crossover. The fitness function for the proposed approach is the 10-fold cross-validation accuracy of the SVM classifier. The SVM classifier is applied to the training pattern. The hypothesis function h is defined in Equation 6.2 as follows:

$$h(n_i) = \begin{cases} +1 & \text{if } w \times n_i + b \geq 0 \\ -1 & \text{if } w \times n_i + b < 0 \end{cases} \tag{6.2}$$

Let α_i be the accuracy obtained from the ith shuffled training set where $1 < i < k$. The fitness function(Equation 6.3) is defined as follows.

$$f(Ch_i) = -\frac{\sum_{i=1}^{k} \alpha_i}{k} \tag{6.3}$$

The negative k-fold cross validation accuracy of SVM on the current selection gives the fitness value. The higher the value of the fitness the better selection of instances is done.

In the selection procedure, the current pool of chromosomes is considered. The selection procedure is done by evaluating the fitness of the current selection, and selecting the chromosomes or the best parents in the population for the mating procedure to create the new generation. The selection is carried out through the roulette wheel [80] approach. The roulette wheel selection method is used for selecting all the individuals for the next generation. A roulette wheel has been constructed from the relative fitness of each individual. The roulette wheel [80] allows the fittest selection of instances to get selected; hence, the fittest chromosomes have the higher probability of getting selected. The probability involved in chromosome selection; $\rho(Ch_i)$ is defined in equation 6.4.

$$\rho(Ch_i) = \frac{f(Ch_i)}{\sum_{i=1}^{pSize} f(Ch_i)} \tag{6.4}$$

6.1.2.4 Crossover and Mutation

The fraction of the population that satisfies the fitness criteria is passed for the crossover and mutation operations. The chromosomes are probabilistically selected and let r be the fraction of the population ($0 < r < 1$) that are passed for the crossover. In the very process of crossover the selected instances of the data set are swapped among the chromosome pairs; yielding new solution pairs for the next generation. Experimentation with both single-point and two-point crossover for the creation of a new generation has been carried

out. The mutation operation controls the diversity factor of the population pool. The encoded instances of the dataset after the crossover operations are mutated according to the mutation probability. For the proposed approach, 1-bit flip has been adopted. A randomly selected instance of a chromosome is selected or deselected after mutation operation.

With the crossover and mutation probabilities set, and the end phase reduction procedure (figure 6.4) is executed for multiple generations. m'' instances in the range of 1 to $2^{m'}$ are obtained at the end which gives the best undersampled dataset. A visualization of the mutation and crossover operation have been presented in Figure 6.5. The block diagram demonstrates how a single point crossover and mutation between two chromosomes aids in the formation of a new candidate solution.

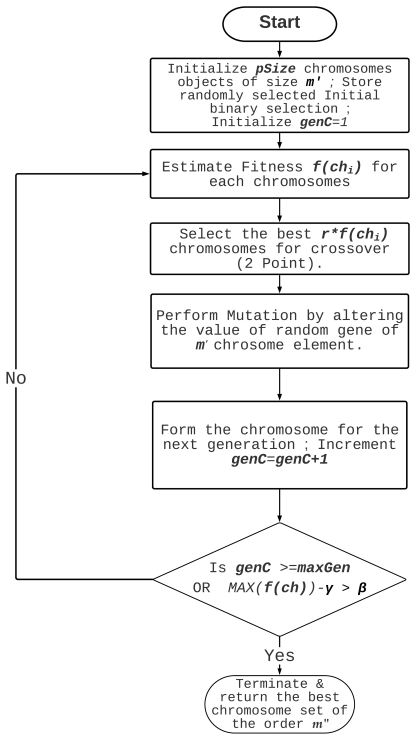


Figure 6.4: Endphase reduction procedure using GA approach

6.1.3 Computational Analysis

The present section discusses the computational complexity and convergence of the GA procedure. The total computation time $T(P)$ can be defined as follows.

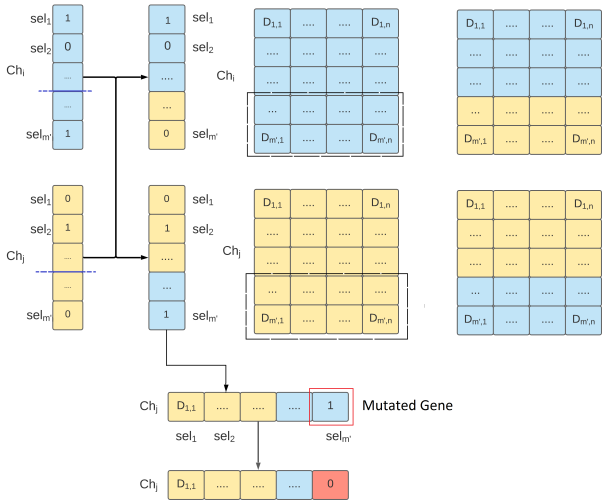


Figure 6.5: Crossover and mutation overview

$$T(P) = T(NN) + T(GAR) \tag{6.5}$$

Here, $T(NN)$ is the computational complexity of the nearest neighbor replacement approach. The objective of the point replacement is to reduce the redundant instances from the pool of samples without any reduction in the kNN accuracy metric. The complexity of nearest centroid consists of building the KD-Tree which requires $\mathcal{O}(m \log^2 m)$ complexity [109]; where m is the number of instances and complexity of point replacement method is designated as $T(PR)$. The complexity of fetching the nearest neighbor from KD-Tree is $\mathcal{O}(\log m)$. $T(NN)$ can be calculated as follows.

$$T(NN) = \mathcal{O}(m \log^2 m) + T(PR) \tag{6.6}$$

The complexity of the iterative point replacement, $T(PR)$ can be expanded as follows.

$$\begin{aligned} T(PR) = \log m + \log(m - k + 1) + \log(m - 2k + 2) + \\ .. + \log(m - ik + i) \end{aligned} \tag{6.7}$$

The process terminates when the following is true.

$$m - ik + i = 0 \tag{6.8}$$

$$k = \frac{m+i}{i} \Rightarrow \mathcal{O}(m) \tag{6.9}$$

$$\begin{aligned} \therefore T(NN) &= \mathcal{O}(m \log^2 m) + O(m) \\ &= \mathcal{O}(m \log^2 m) \end{aligned} \tag{6.10}$$

The complexity of the GA approach i.e. $T(GAR)$ depends highly on the size of the chromosome, the size of the population, and the cost function. The cost or the fitness function of the GA procedure is the negative of the cross-validation accuracy. The GA approach is loosely dependent on the theory of evolution. The GA procedure's knowledge for the solution space is very vast. Thus, it is acceptable to agree on a near-optimal solution. Furthermore, randomness is ingrained in the GA procedure with the selection, crossover, and mutation operation so that it does not get stuck to a local optima. In this very process of instance selection, the size of the chromosome variable depends on the number of instances received from the first phase of the pipeline. Thus, the complexity depends on the number of instances, that is the dimension of the problem considered here. However, GA does not perform an exhaustive search. The first phase of the pipeline is responsible for reducing redundancy without affecting the classification accuracy. Initializing GA with a good initial population, such as the outcome of Phase 1, and taking advantage of the exploration through the crossover and mutation, effectively reduces the complexity to an approximate linear phase.

The process of GA-based instance selection is a part of the data preprocessing phase performed during the off-line training. Thus, the complexity of the procedure will not affect any real-time processing or prediction phase.

6.2 Approach 2: Single Phase Approach using BPSO

In this section, the meta-heuristic based instance selection approach is carried out using BPSO based approach. The variables used in the work are listed in Table 6.2. The floorplan P is defined as the combination of the feature set AP=$\{ap_1,.. \ ap_n\}$ that is, the RSS from registered set of AP sources and the well defined virtual grids or labels Y. As defined in the previous chapters, let F be the fingerprint vector denoted using a 2D vector with its assigned label denoted as $< f_{v,1}, ..., f_{v,n}|Y_v >$; where $f_{v,1}$ is the RSS from AP_1

and Y_v is the label of the fingerprint instance F_v.

Table 6.2: Terminologies used in defining the proposed single phase tuple selection approach

Variables	***Significance***	***Variables***	***Significance***
P	Selected floorplan upon which localization is to be carried out	$percR$	Percentage reduction of fingerprint instances
n	Count of registered set of access points	ec	Extra cost imposed on difference of accuracy score metric
AP	Registered set of WiFi access vector $<ap_1,..,ap_n>$	α_i	Classification accuracy of fold i
F	2D fingerprint radiomap set $<f_{v,1},..,f_{v,n}— Yv>$	$meanD_i$	ith instances mean distance to its nearest neighbor set
Y_v	Virtual grid label of vth fingerprint instance	δ	Acceptable mean distance threshold percentage
noP	Swarm Size	β	Percentile mean distance evaluated
S_i	Particle object variable where i<=noP	W	Weight parameter of the BPSO based selection approach
$S_i.f$	ith particle selection/deselection vector	$c1$	Social acceleration coefficient
$S_i.v$	ith particle velocity vector	$c2$	Cognitive acceleration coefficient
$S_i.shape$	No of instances selected by ith particle of the swarm	$r1$	stochastic adjustment weight of social component
$S_i.best$	ith particle's best selection of fingerprint instances	$r2$	stochastic adjustment weight of cognitive component
$S_i.fitness$	ith particle's evaluated fitness	$w1$	weight associated with class rate fitness metric
$Gbest$	Global best selection of fingerprint instances	$w2$	weight associated with percentage reduction fitness metric
$globalF$	Global best evaluated fitness	$w3$	weight associated with class rate fitness metric
cr	crossvalidation accuracy of a classifier	$score$	k-disagreeing neighbor score of an instance

6.2.1 Encoding

The goal of any *instance selection* method is to reduce the superfluous instances from the actual dataset. The particle vector is denoted by the object variable S; where S_i represents the i^{th} particle object variable. The encoding of particle objects are done by mapping each row of the fingerprint instance to position vector S_i; where $1 \leq \text{i} \leq noP$. Here, $S_i.f[j]$ represents the selection/de-selection of fingerprint instance j by the i^{th} particle; 1 represents the selection of the instance while 0 represents the de-selection of it. $S_i.shape$ represents the number of fingerprints selected by particle S_i.

The velocity component $S_i.v[j]$ represents the velocity or the update pertaining to the next state update by particle i. Every particle has its best selection of fingerprint instance given as $S_i.best$. The global best solution is given by the vector $Gbest$ with a global fitness variable $globalF$. The procedure is summarized in Figure 6.6.

6.2.2 Cost Function

The fitness is evaluated using the cost function that considers three important parameters-class rate cr, percentage reduction $percR$ and extra cost ec. Let α_i be the classification accuracy for fold i out of k-fold cross validation of a selected classifier *clf* trained on particle $S_i.f$. The class rate cr is evaluated using k-fold cross validation accuracy of a selected classifier *clf*.

$$cr = \frac{\sum_{i=1}^{k} \alpha_i^{(clf)}}{k} \tag{6.11}$$

The percentage reduction $percR$ (Equation 6.12) evaluates the fraction of instances reduced.

$$percR = \frac{m - S_i.shape}{m} \tag{6.12}$$

Here, m is the total number of instances and $S_i.shape$ gives the number of selected fingerprint by particle S_i. An extra cost ec is estimated that gives the fraction of reduction in the accuracy metric from the original score. Equation 6.13 gives the extra cost metric that needs to be reduced; the variable $orgScore$ represents the cross validation accuracy on the whole dataset without the application of selection approach with classifier *clf*.

$$ec = \frac{orgScore^{(clf)} - cr^{(clf)}}{orgScore^{(clf)}} \tag{6.13}$$

Objective lies in maximizing the summed value with assigned weight quantities.

$$\max_{0 \leq w1,w2,w3 \leq 1} w1 \times cr + w2 \times percR - w3 \times ec \tag{6.14}$$

6.2.3 Selection procedure and update rule

The particle S_i's selection are distributed in the search space with the objective of reaching the global optimal solution or selection; i.e. $Gbest$. The update rule has been modified to incorporate instance toggling that leads to de-selection of probable outliers for the proposed BPSO based algorithm. Thus the update rule consists of two parts as stated below.

- Selection Update Rule
- Probable outlier de-selection through instance toggling.

6.2.3.1 Update Rule 1: Selection Update Rule

The two update rules that govern the selection procedure are the *velocity* update rule and fingerprint instance *selection* update rule (Equation 6.15, 6.16, 6.17). The calculated velocity value of particle i at iteration t is evaluated by substituting the velocity value at iteration $t-1$.

The present subsection discusses on the selection update rule while subsequent subsection briefly describes the toggling update. At first the evaluated velocity is passed through a sigmoid function in order to carry out the selection or de-selection of a fingerprint instance. If the velocity value $S_i.v[j]$ reaches 0 then the *sigmoid* function is equal to 0.5. Thus, selection of the instance occurs depending on the direction of the change of value. If $S_i.v[j] > 0$ then the sigmoid value is more than 0.5.

$$\begin{aligned} S_i.v^{(t+1)} = W \times S_i.v^{(t)} + c1 \times r1 \times (global^{(t)} - S_i.f^{(t)}) \\ + c2 \times r2 \times (S_i.best^{(t)} - S_i.f^{(t)}) \end{aligned} \tag{6.15}$$

$$sigmoid(S_i.v[j]^{(t+1)}) = \frac{1}{1 + e^{-S_i.v[j]^{(t+1)}}} \tag{6.16}$$

$$S_i.f[j]^{(t+1)} = \begin{cases} 1 & \text{if } sigmoid(S_i.v[j]^{(t)}) >= 0.5) \\ 0 & \text{if } sigmoid(S_i.v[j]^{(t)}) < 0.5) \end{cases} \tag{6.17}$$

The velocity evaluation of particle i's jth instance; i.e. $S_i.v[j]$ is estimated by substituting previous velocity value times a weight estimate W. The difference $(global - S_i.f)$ and $(S_i.best - S_i.f)$ enable the particle to move towards a present best selection and finally, towards a global best selection. The change in the selection/de-selection of an instance for particle i occurs when $S_i.f[j]$=1 but the global best vector $global[j]$=0 or particle i

best selection $S_i.best[j]$=0. Such a situation causes the velocity gradient to be towards a negative direction. The velocity component $S_i.f[j]$ is passed through a sigmoid evaluation which pushes the selection $S_i.f[j]$ of the instance towards 0. The selection of the instance occurs when $global[j]$ or $S_i.best[j]$ is equal to 1 but $S_i.f[j]$=0 for the previous iteration t.

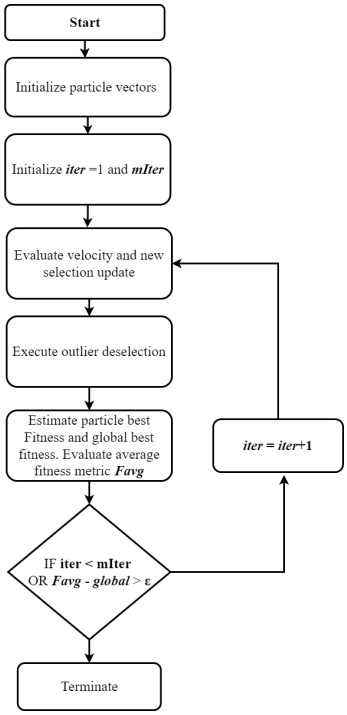


Figure 6.6: Instance selection procedure flowchart

Algorithm 5: costFunction(S_i)

1 w_1, w_2, w_3: Weights

2 $pr \leftarrow \frac{m - S_i.shape}{m}$

3 $cr \leftarrow CVAccuracy(S_i)$

4 $ec \leftarrow \frac{orgScore - cr}{orgScore}$

5 $fitness \leftarrow w_1 * cr + w_2 * pr - w3 * ec$

6 return $fitness$

6.2.3.2 Update Rule 2:Instance toggling and outlier de-selection

Each of the particle S_i is a candidate solution to the problem of global best selection of instances. In the proposed approach a method to selectively detect and de-select outliers in the flow of the process has been discussed. The procedure follows by evaluating the

Algorithm 6: bpsoUpdate(S)

```
1  Gbest:Global Best finger print instance selection vector, globalF:Global Best Fitness,
2  m:Total number instances , S_i.best:Local best fingerprint selection vector by particle i,
     S_i.f: Current Selection of Instance by particle i, S_i.v:Velocity vector of particle i,
     S_i.fitness:Fitness value of selected AP by particle i.
3  for i=1 to noP do
4      //velocity update
5      cg ← c1 * r1 * (global − S_i.f
6      sc ← c2 * r2 * (S_i.best − S_i.f)
7      S_i.v ← W * S_i.v + cognitive + social
8      for j=1 to m do
9          select ← sigmoid(S_i.v[j])
10         if select ≤ 0.5 then
11             S_i.f[j] ← 0
12         else
13             S_i.f[j] ← 1
14         end
15     end
16     S_i.f ← outlierDeselection(S_i.f, Y, m)
17     fitness ← selectionFitness(S_i.f)
18     if fitness > S_i.fitness then
19         S_i.best ← S_i.f
20         S_i.fitness ← fitness
21         if fitness > globalF then
22             Gbest ← S_i.f
23             globalF ← fitness
24         end
25     end
26 end
```

nearest neighbors of a selected instance. A KD-Tree [109] is constructed to evaluate the nearest neighbors of a selected fingerprint instance.

Definition 1. F_v defines a fingerprint tuple $< f_1, ..f_n >$ with its associated class label or grid position Y_v. A KD-Tree $KDT(F, m, n)$ is constructed with median splitting having height $O(log(n))$.

Definition 2. A Query Q_{k,F_v}; where $v \in m$ is a nearest neighbour query that returns a 1-D vector of k fingerprint instances' indexes; where $k \in \mathbb{N}(k \geq 1)$. The output vector nf is evaluated using a distance function d as follows.

$$nf = \arg\min_{v \leq m} d(Q_{k,F_v}, F_v) \tag{6.18}$$

nf contains the closest fingerprint instance's indexes to the tuple F_v.

Algorithm 7: outlierDeselection(S_i, Y, m)

```
kdt ← KDT(S_i, m, n)
for j=1 to m do
    if j ≠ 0 then
        Evaluate query Q_{k,F_j} from kdt and store in np_j
        score,meanD ← Q^{(KDN)}_{np_j,j}
        /*the query to return the count of k-disagreeing neighbour and mean
          average distance.*/
        if score/k ≤ δ AND meanD ≥ β then
            if rand() ¡ mProb then
                S_i.f[j]=0
                /* outlier detection and de-selection */
            end
        end
    end
end
```

Definition 3. $Q^{(KDN)}_{nf,F_v}$ is a query defined to get the count of number of fingerprint instances that do not share the same class as the instance F_v. The result of k disagreeing query is defined as follows.

$$Q^{(KDN)}_{nf,F_v} = |x \in nf \wedge Y_x \neq Y_v| \tag{6.19}$$

The mean distance of each instance from its k-neighbors is evaluated into a vector, $meanD_i (1 \leq i \leq m)$. The K-Disagreeing measure score of a fingerprint instance F_v results into the fraction of nearest neighbor instances $score (0 \leq score \leq 1)$ which do not share the same class boundary.A point which has a higher percentage of its k-neighbors agreeing with its class can be considered, i.e. for which the score is less than a certain value δ; where $\delta < 0.5$. If the mean distance for this point is low (tending towards 0), this means the point might be located at/near a cluster of that particular class. If the mean distance is higher than a certain threshold and *score* value is less than a certain δ then the selected instance can be a probable outlier (Table 6.3). This has been visualized in Figure 6.7.

The percentile evaluation of the function $meanDistance(F_v)$ (Equation 6.20) from its nearest neighbor set is evaluated. The percentile evaluation gives the rank of percentage scores of mean distance in the frequency domain where the limit is 0 to π_t; where t=100

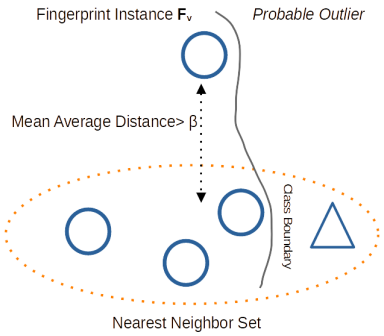


Figure 6.7: Probable Outlier visualization with respect to KDN score and mean average distance

gives the $100th$ percentile of mean distance distribution.

$$\beta = \int_0^{\pi_t} meanDistance(F_v)\, dF_v \tag{6.20}$$

Algorithm 7 demonstrates the outlier de-selection procedure. The mean distance threshold β and a kDN score threshold δ are used for the outlier de-selection procedure. For a selected particle S_i if line 7 and 8 of the Algorithm 7 satisfies for instance j then the instance is de-selected by assigning 0 to $S_i.f[j]$.

Table 6.3: Inference drawn on the fingerprint instance with respect to k-disagreeing score and mean average distance to its nearest neighbor

	k-Disagreeing Score			
***Mean Distance* >β**	**<0.3**	**>0.3 and <=0.5**	**>0.5 and <=0.75**	***>0.75 and <=1.00***
False	*Same Class*	*Indecisive*	*Boundary Element*	*Boundary Element*
True	*Probable outlier*	*Indecisive*	*Boundary Element*	Instance closer to its clusters center

The flowchart discussing the steps involved in the selection process is demonstrated in Figure 6.6. The instance selection procedure is iterated for *iter* times for which the BPSO update rule is carried out. On every iteration the selected instances $S_i.f$ by particle i are passed through the outlier deselection method (Algorithm 7). If an instance j is detected to be an outlier following Algorithm 7; the value of $S_i.f[j]$ is set to 0 with a randomized probability estimate. The particle's best fitness $S_i.fitness$ and global best fitness $globalF$ are evaluated using Algorithm 5. The difference of global best fitness $globalF$ and the average of fitness score $Favg$ should be less than a certain threshold value ϵ in order to achieve convergence.

6.3 Experimental Analysis

This section discusses the experimental setup and the results to validate the performance of the proposed instance selection approach.

6.3.1 Experimental Dataset

The experiments were conducted on the own collected *combined* dataset. More insight into the collected dataset can be found in [94]. In order to effectively evaluate the performance of the proposed approach three benchmark datasets JUIndoorLoc [71], UJIIndoorLoc [72] and Shopping Mall (part of Microsoft Research) [97] have been considered. Table 6.4 gives an overview of the sample count of the considered datasets used for experimentation.

Table 6.4: Overview of sample count in the experimental datasets

Dataset	Instance Count	Features
D1	548	105
Combined (D1,D2,D3,D4)	5428	105
JUIndoorLoc	*6639*	105
UJIIndoorLoc	*9493*	520
Shopping Mall	*3283*	55

Table 6.5: Parameter setting during the conducted experiments

Parameter	Setting	Parameter	Setting
Maximum iteration($mIter$)	100	kDN score threshold(δ)	0.3
Particle swarm size(noP)	50	Deselection Probability ($mProb$)	0.05
Social Component($c1$)	2	Decision Tree (Tree Depth)	80
Cognitive Component($c2$)	2	SVM(Kernel)	polynomial
Particle initial instance selection($S_i.f$)	random	kNN(no of neighbors)	3
BPSO weight parameter(W)	1		
Percentile Mean Distance(β)	π_{85}	Neural Network(Layers)	(70,70,100)

6.3.2 Evaluation Metric

The experiments are conducted after proper preprocessing and parameter setting. Table 6.8 gives the parameter setting used during the presented experimental analysis. The results are presented in the subsequent section. Following are the metric considered during the experimentation.

- **Instance Count:** The instance count metric evaluates to the count of the number of tuples selected from the input dataset.

- **Accuracy:** The accuracy of the machine learning model gives the number of correctly classified instance with respect to localized grid point by the total number of classification. The presented accuracy are done using 70% of the data in the training instance and 30% in the testing.

- **Hardness score:** The frequency of a particular instance being misclassified by a classification method is measured by instance hardness. It enables a deeper analysis of the performance of the learning algorithms [110]. The k-disagreeing neighbor is evaluated as the ratio of k-neighboring instances of a selected instance F_i that do not share the class boundary as in Equation 6.21.

$$kDN(F_i) = \frac{|F'_j|F'_j \in kNN(F_j) \wedge Y_j \neq Y'_j|}{k} \tag{6.21}$$

The $kNN(F_j)$ estimates the k nearest neighbor instances, Y_j is the location label from where the fingerprint instance F_j is collected.

A fingerprint is regarded as hard if they are frequently misclassified by majority of the learning algorithms. The fingerprint hardness measure $FHD(F_i, Y_i)$ is estimated using Equation 6.22.

$$FHD(F_i) = 1 - p(Y_i|F_i, h) \tag{6.22}$$

Here, h is the hypothesis function with which F_i is mapped to Y_i. A reasonable outlier is one with a greater hardness score, and occurrences away from the mean that is difficult to classify by any classification model. Smith et al. in [111] proposed instance level analysis for defining the hard instances as follows.

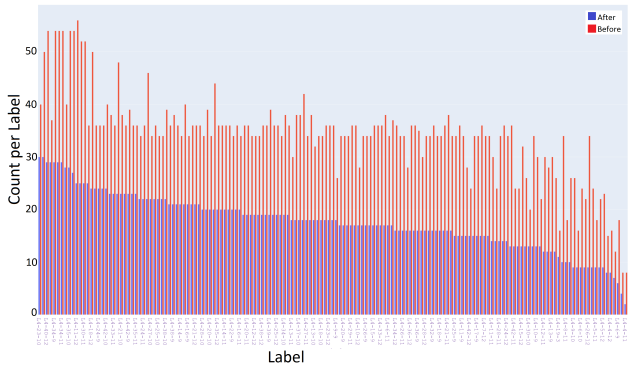
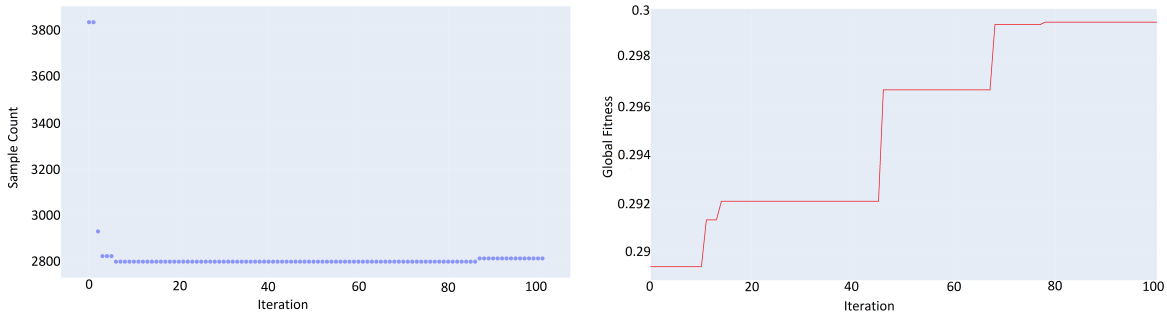


Figure 6.8: Dataset visualization before and after the application of BPSO based selection procedure



(a) Variation of sample count at each iteration step during the BPSO selection process

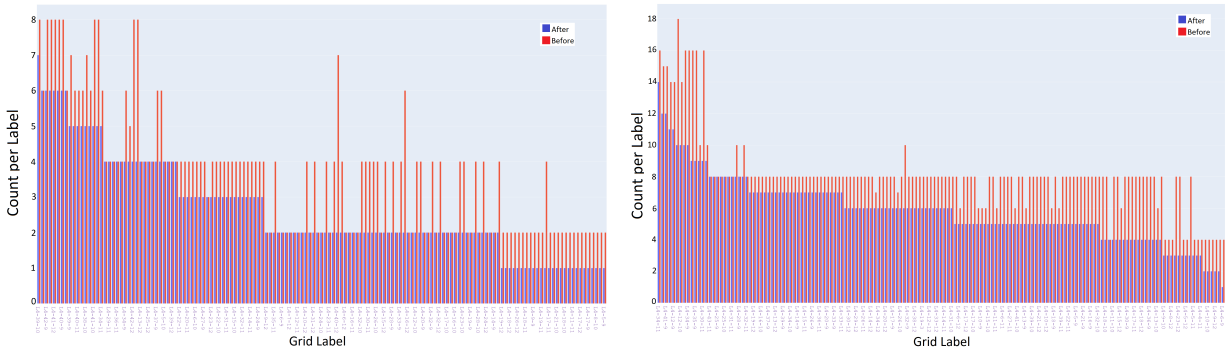
(b) Global fitness variation at each iteration step during the BPSO selection process

Figure 6.9: Fitness and sample count analysis at each iteration step of the BPSO based selection process

- *Class Likelihood*: Likelihood measures the probability that a particular fingerprint F_i belongs to a virtual grid Y_i.
- *Class Likelihood Difference*: The likelihood difference is evaluated by taking the difference between the maximum likelihood and class likelihood.
- *Disjunct class percentage*: Percentage of fingerprints in the disjunct of an instance fingerprint that shares the same virtual grid. The disjunct is evaluated by constructing a decision tree classifier.
- *Percentage of features in overlapped area*: Measures the percentage of access points pertaining to the features of instance F_i in the overlapping region of labels.
- *Local Set Cardinality*:Local set cardinality of an instance F_i, gives the set of instances whose distance to F_i is closer than than F_i and F_j's nearest neighbor; where $Y_i = Y_j$.

- *Local Set Radius*: Considers a radius of local set of fingerprint instance to F_i.
- *Pruned Tree Depth*: The depth of the leaf node evaluated that predicts F_i in a pruned decision tree classifier which is normalized by the maximum depth of the tree.
- *Un-pruned Tree Depth*: The depth of the leaf node evaluated that predicts F_i in a un-pruned decision tree classifier which is normalized by the maximum depth of the tree.

- **Error Deviation:** The Euclidean error estimate with MAE evaluation as defined in the previous chapters(Chapter 2,3), has been utilized.
- **Cumulative Distribution Function:** For statistical analysis, the Cumulative Distribution Function(CDF) is estimated which is evaluated on the error metric the equation for which is already defined in Chapter 3.



(a) Sample Count per label before and after the application of the proposed approach on D1 dataset

(b) Sample Count per label before and after the application of the proposed appraoch on D2 dataset

Figure 6.10: Dataset visualization before and after the application of proposed selection approach on collected imbalanced dataset- D1 and D2

6.3.3 Results of two phase Approach

In this section, the performance of the GA based approach has been analyzed for the collected radiomap dataset and the JUIndoorLoc dataset. The result showcased are from the nearest neighbor and GA based instance selection pipeline.

An experiment to evaluate the performance of the model is given by the accuracy metric visualization. The accuracy before and after the application of the instance selection approach is plotted in Figure 6.11. It can be observed that for the *collected* combined

dataset there is a minor drop in the accuracy metric with almost more than 40% reduction in sample count.

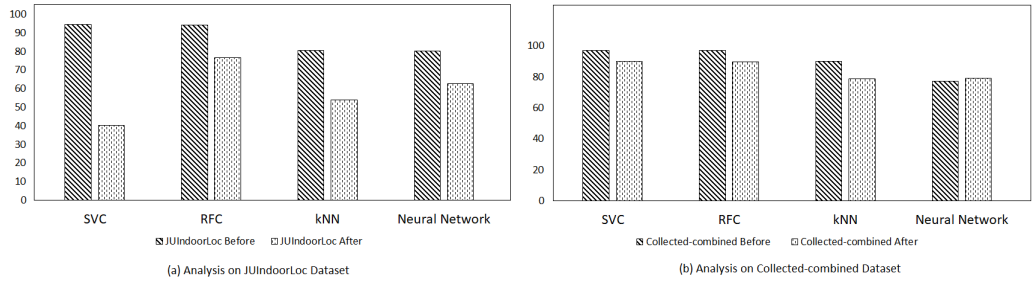


Figure 6.11: Accuracy analysis of the two phase hybrid tuple selection pipeline using nearest neighbor and GA approach

This leads us to the next experiment that showcases the error deviation in Figure 6.12, which is anther important metric to evaluate the performance of the tuple selection pipeline. In almost majority of the case it can be seen that the deviation error is decreasing for almost all of the classification process. This ensures that reduction pipeline is mitigating the effect of outliers from the radiomap dataset while retaining the border fingerprint instances.

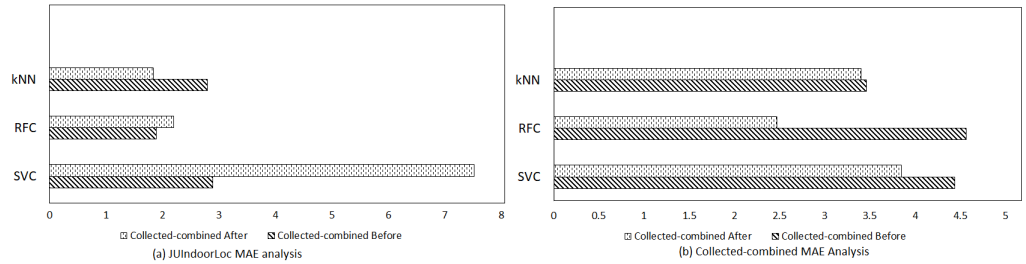


Figure 6.12: Error Deviation analysis of the two phase hybrid tuple selection pipeline using nearest neighbor and GA approach

To explore the effectiveness of the selection pipeline on other sensing modalities, experimentation on benchmark datasets in the Human Activity Recognition(HAR) domain is carried out. This is done as the dataset of HAR is usually opted for navigation problem which well aligns with the localization problem. The UCI HAR dataset contains 7352 samples, while the WISDM dataset contains $1,098,207$ samples. Both the datasets have 6 activity classes whose details are summarized in Table 6.6 and Table 6.7. The experiments are conducted and tested using a train/test split where 70% data is put into the training

set and 30% is separated for testing.

Table 6.6: UCI HAR dataset sample distribution

Label	Count	Percentage(%)
Walking	1226	*16.6*
Walking Upstairs	*1073*	*14.5*
Walking Downstairs	*986*	*13.4*
Sitting	*1286*	*17.4*
Standing	*1374*	*18.6*
Laying	*1407*	*19.1*

Table 6.7: WISDM dataset sample distribution

Label	Count	Percentage(%)
Walking	424,400	*38.6*
Jogging	*342,177*	*31.2*
Upstairs	*122,869*	*11.2*
Downstairs	*100,427*	*9.1*
Sitting	*59,939*	*5.5*
Standing	*48,395*	*4.4*

The experiments conducted are parameter tuned which are reported in Table 6.8. For the first phase pipeline in the point replacement method, the k parameter which is the number of points selected for replacement is set to 3. In the GA-Based instance selection, the number of chromosomes is set to 20 with single bit flip, mutation probability set to 0.2 and single point crossover fraction set to 0.4. These parameters are evaluated based on the conduction of experiments. The setting for the classification models is carried out through Grid Search [112].

Table 6.8: Parameter setting for the conducted experiments

Parameter	Setting	Parameter	Setting
'k' - nearest Centroid	*3*	*kNN(no of neighbors)*	*3*
Chromosome Population 'pSize'	*20*	*RF- Tree Depth*	*110*
Shuffled training set 'n'	*5*	*RF- No of Trees*	*100*
Crossover fraction 'r'	*0.4, Single Point*	*SVM(Kernel)*	*polynomial*
Maximum generation 'maxGen'	*100*	*SVM (Gamma)*	*scale*
Mutation Probability 'mProb'	*0.2, Single Bit Flip*	*Neural Network (Layers)*	*<100,120,80>*

In Table 6.9 a count of the number of samples before and after the reduction pipeline. For the UCI HAR dataset, a total of 10299 datapoints is passed onto the centroid-based method, which leads to a reduction of dataset size to 9399. Before the application of the pipeline, the accuracy achieved with SVM is 90.06 %. Following the proposed pipeline

further reduces the size to a size of nearly 5196 data points with an increase in accuracy value.

Table 6.9: Sample count and accuracy estimation at each sub-phase of the instance selection pipeline

Dataset	Without Pipeline		Only Centroid Method		Only GA Method		Proposed Selection (Centroid+GA)	
	Datapoints	Accuracy	Datapoints	Accuracy	Datapoints	Accuracy	Datapoints	Accuracy
UCI	*10299*	*90.6*	*9399*	*90.9*	*5232*	*91.7*	*5196*	*93.45*
WISDM	*9000*	*69.8*	*8100*	*70.6*	*4067*	*72.2*	*4015*	*72.8*

A study by altering the fitness classifiers has been carried out in Figure 6.13. The convergence plot is shown, where it can be observed that during the GA procedure, all the three classifiers(SVM,RFC and kNN) used as the fitness metric have converged after 20 generations in case of UCI-HAR dataset. It can also be observed that SVM selected as fitness function performed better in UCI HAR dataset. While kNN selected as fitness function produced better reduced tuples with the WISDM dataset. The convergence on WISDM dataset has been observed after 30 generations.

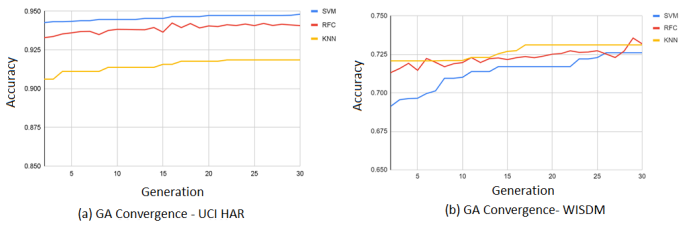


Figure 6.13: Variation of accuracy of the GA procedure on varying the fitness classifiers

6.3.4 Results of Single phase Approach

This section presents the results obtained from the collected combined radiomap dataset and three benchmark datasets with the single phase approach using BPSO. First, the impact of different smartphone configurations and execution environment (workstation or Raspberry Pi) have been analyzed w.r.t the collected dataset. This is followed by an analysis of the effectiveness of the proposed approach for different benchmark datasets that are collected over various experimental regions (2 universities and one shopping mall). The experimental results of both the two phase approach as well as the single phase approach is presented in the subsequent sections.

6.3.4.1 Analysis on Collected Radiomap Dataset

The visualization of the magnitude of the reduction obtained after executing the proposed selection approach on the collected *Combined* radiomap dataset is presented in Figure 6.8. For the experiment, the k-nearest neighbor classifier in the fitness calculation method. The sample per label before and after the application of the proposed selection can be seen with respect to each class label.

Table 6.10: Comparison of cross validation accuracy before and after the application of the proposed selection approach

Dataset	**kNN**		**SVM**		**Decision Tree**	
	Before	**After**	**Before**	**After**	**Before**	**After**
D1	40.2(± 0.8)	39.7(± 0.4)	72.8(± 2.8)	70.3(± 1.4)	71.25(± 1.7)	69.46(± 1.2)
D2	82.5(± 0.7)	68.6(± 1.2)	87.33(± 0.7)	84.33(± 0.6)	90.93(± 1.2)	85.3(± 0.8)
Combined	83.24(± 2.1)	84.8(± 1.2)	94.43(± 1.1)	93.5(± 1.6)	93.8(± 0.9)	92.5(± 1.4)

Table 6.11: Accuracy comparison between collected combined dataset and benchmark set-UJIIndoorLoc, JUIndoorLoc and China Shopping Mall part of Microsoft Research. The results are presented before and after applying the proposed approach

Classifier	***Decision Tree***		***SVM***		***kNN***		***Neural Network***	
	Before	***After***	***Before***	***After***	***Before***	***After***	***Before***	***After***
Combined	94.8	92.39	95.43	93.3	84.53	83.8	63.2	61.35
JUIndoorLoc [71]	98.14	97.58	98.54	98.02	95.53	92.02	60.16	58.45
UJIIndoorLoc [72]	53.52	52.16	66.74	63.51	59.51	57.85	55.67	52.89
Shopping Mall [97]	92.1	85.34	92.6	90.52	91.35	86.94	84.9	83.34

It is important to investigate the stability of the BPSO based algorithm. So, the variation of selected instances and global fitness metric of particles with respect to iteration has been visualized in Figure 6.9(a) and Figure 6.9(b), respectively. The plot is extended to 100 iterations and k-NN is used in the cost estimation function. It can be observed that the procedure has stabilized as the particles converge toward a near optimal selection.
A study of the effect of the proposed selection approach on imbalanced dataset. $D1$ & $D2$ are imbalanced datasets upon which the experiments were conducted. Figure 6.10 gives a histogram of sample count before and after the application of the selection approach. Table 6.10 presents the accuracy comparison presented with $D1$, $D2$, and $Combined$ dataset.

10-fold cross validation accuracy values are reported so that reproducibility of the results could be better ensured. It can be observed from the table and the figure that the proposed selection approach is able to give stable performance both with respect to instance reduction as well as accuracy metric.

The training of the collected combined dataset has also been analyzed in lightweight edge level device such as, Raspberry Pi 3. Figure 6.14 gives an overview of the utilization. It can be observed that the training time has decreased due to the decrease in the number of selected instances. The effect will be more significant with the increase in the magnitude of data samples in the original dataset. It has also been observed that the memory requirement has decreased by 5%-7%. This indicates the effectiveness of the proposed instance selection approach for ensuring real-time indoor localization where the location prediction can be executed on a local environment or an edge.

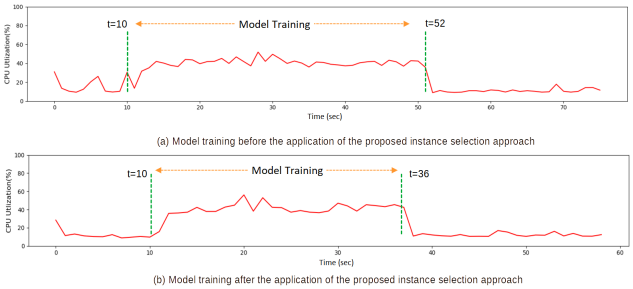


Figure 6.14: CPU utilization analysis on Raspberry Pi-3 using SVM Classifier(kernel=polynomial) on collected combined dataset

6.3.4.2 Analysis on benchmark datasets

Accuracy evaluation on benchmark datasets- JUIndoorLoc, UJIIndoorLoc and Shopping Mall dataset in China with (training(70)/testing(30)) percentage split is carried out using selected set of classifiers- Decision Tree [113], SVM [114], kNN [115] and Neural Network [116]. Table 6.11 gives the accuracy measurement on selected set of classification model. The reported accuracy values of the model validation is done for both before and after the application of proposed selection strategy. It can be observed from the table that the proposed approach is able to achieve the selection with minimal decrease in the accuracy metric.

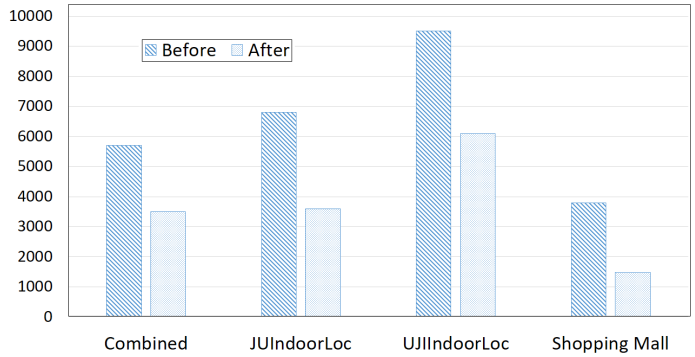


Figure 6.15: Instance count before and after the application of selection procedure(Dataset- collected Combined device dataset, JUIndoorLoc, UJIIndoorLoc and Shopping Mall).

Figure 6.15 reports the count of the number of instances before and after the application of BPSO based selection approach. A bar plot on the collected combined dataset and benchmark set-UJIIndoorLoc, JUIndoorLoc and Shopping Mall is showcased. From the figure, it can be observed that the proposed approach is able to reduce the number of tuples by more than 40% yet retaining appreciable classification accuracy for different types of indoor places reflected by the datasets.

6.3.4.3 Fingerprint Hardness Analysis

The particle fitness evaluation involves the use of a classification model upon which two metrics are evaluated, namely-*class rate* and *extra cost*. Experiments involving altering the classification model used is carried out. A plot for hardness analysis is presented in Figure 6.16. The experimentation was done by altering the classifier used during the evaluation of the fitness. In Figure 6.16(a), the hardness analysis was carried out on the whole combined device dataset. k-NN, Decision Tree, and SVM classifiers are used for classification to obtain the accuracy considered as the fitness metric for which the subsequent kDN analysis is reported in Figure 6.16(b), (c), and (d). It can be observe from the figure that after the application of the proposed approach the score metric has become more distributed indicating that the data distribution has changed. Increase in the kDN score lying in range $0.8 - 1.0$ indicates that the border points are retained.

Other meta features are evaluated for each instances that is performed on the collected combined radiomap set. Figure 6.18 showcases the histogram visualization before the application of selection approach while Figure 6.19 presents the visualization after the

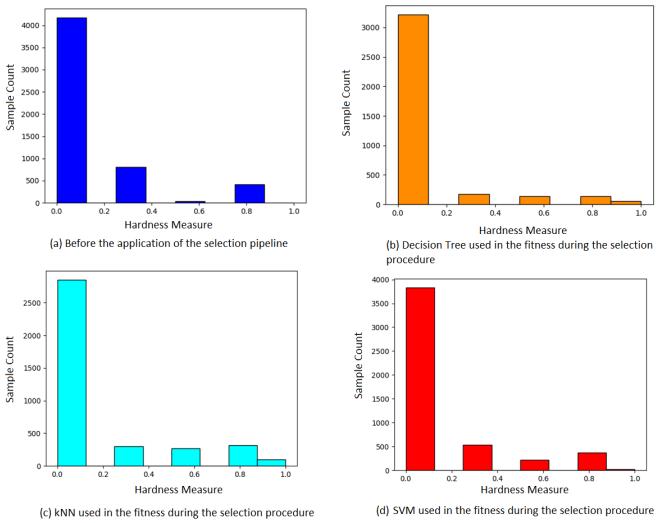


Figure 6.16: Hardness analysis with distribution of k-differentiating neighbor on the collected combined device dataset.

application of the approach. It can be observed that the local set cardinality has become well balanced. Fingerprints that are easier to classify are the ones found in dense areas surrounded by instances of their same label, thus, larger local sets are obtained. The normalized radius for each instance has decreased after the application of the approach, a clear indication that the overall hardness has been perturbed. Retaining the border points might lead to such a decrease; as border points are important for defining the class boundary and are harder to classify.

6.3.4.4 Error analysis

For indoor localization, it is important to measure the distance error when the location prediction goes wrong. Figure 6.17 presents the error deviation in meters from the actual grid point. In Figure 6.17(a), decision tree is used for the fitness calculation and the localization has been done using kNN, decision tree and SVM. For the subsequent figures Figure 6.17(b) and (c), kNN and SVM are used for calculating the fitness metric. It can be observed that both kNN and decision tree used in the cost function has performed at par. The error deviation has reduced from $4m$ to $3.2m$ after the application of BPSO based selection approach.

A CDF plot is estimated in Figure 6.20. The figures show case the meter level error

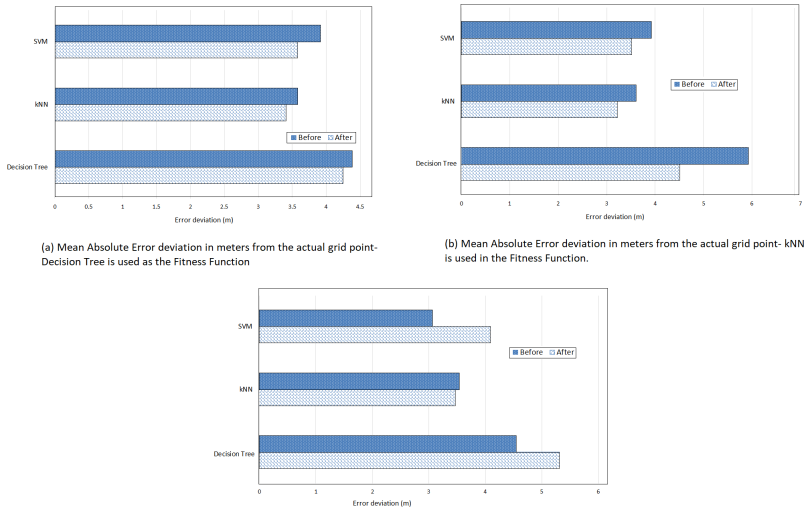


Figure 6.17: Error Deviation visualization from actual grid points for the combined device dataset by altering the classifier in the fitness metric of the proposed instance selection approach. Decision Tree classifier is used in the Training process.

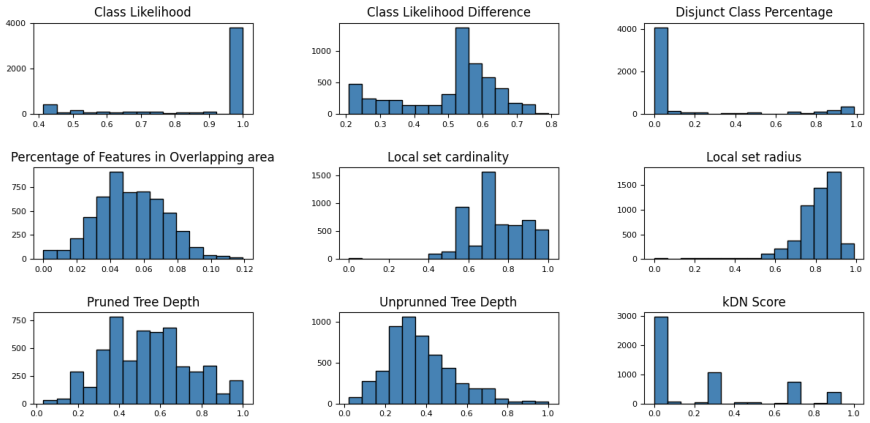


Figure 6.18: Hardness analysis before the application of proposed selection approach on collected combined device dataset.

deviation before and after the application of the proposed approach. Decision tree classifier has been used as the localization model. Figure 6.20(a),(b),(c) presents the distribution visualization after the application of the selection process by altering the cost metric of the particles. It can be seen that majority of the misclassified instances are lying within $5m$ meter deviation. On application of the selection approach the deviation from the actual grid point has improved significantly.

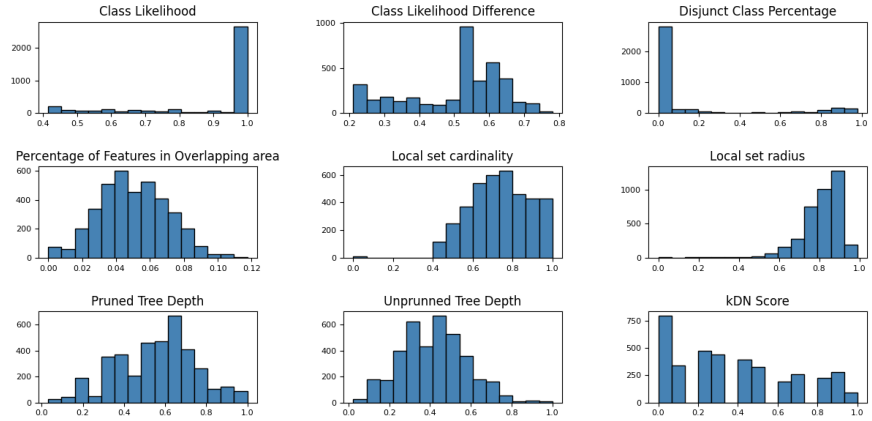


Figure 6.19: Hardness analysis after the application of proposed selection approach on collected combined device dataset.

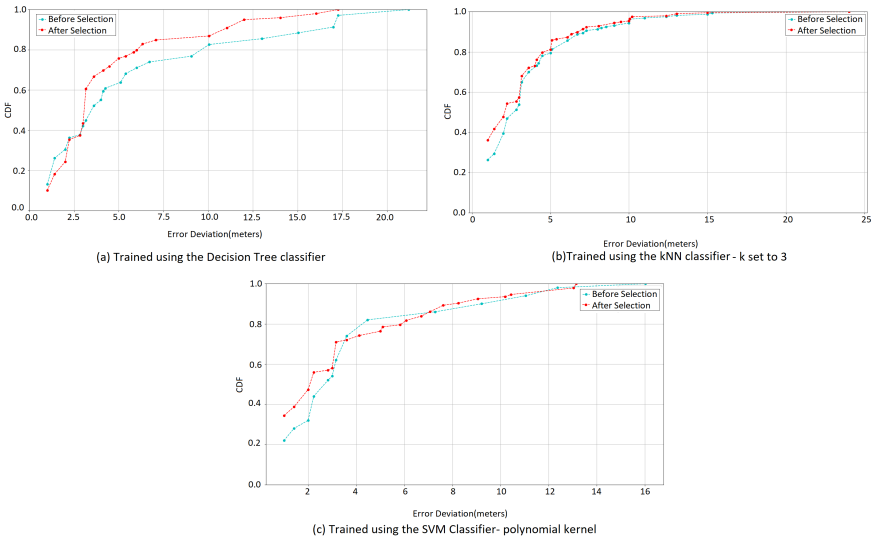


Figure 6.20: CDF visualization of combined device dataset by altering the classifier in the fitness metric of the proposed instance selection approach

6.3.4.5 Performance comparison with state-of-the-art undersampling approaches

The proposed selection approach has been compared with state-of-the-art undersampling approaches namely, Condensed Nearest Neighbor(CONN) [56] and Near Miss [117]. The CONN approach tries to find the subset of tuples for which the performance of the model does not degrade. The Near miss undersampling approach works by examining the class distribution and removes samples from the larger class at random.

The experimentations are conducted using the benchmark dataset JUIndoorLoc and the collected combined radiomap dataset. Table 6.12 presents a comparative analysis of accu-

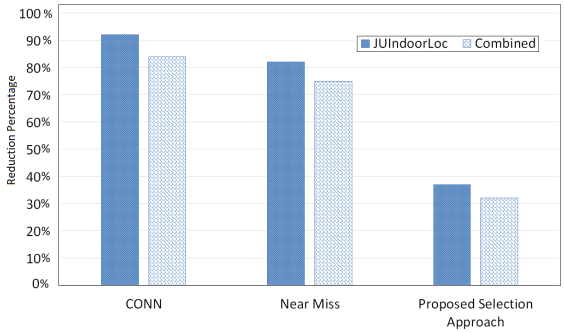


Figure 6.21: Percentage of sample reduction analysis between state-of-the-art underspampling approaches with the proposed approach- JUIndoorLoc and Combined dataset is used for the analysis

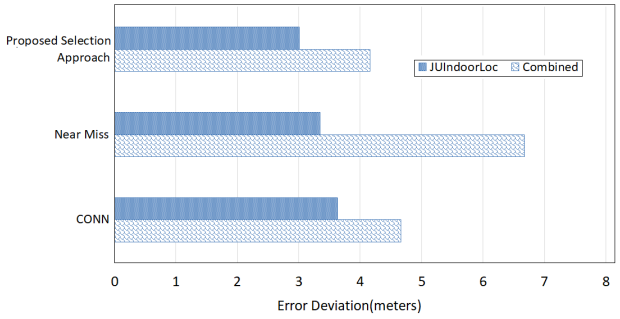


Figure 6.22: Comparative Error deviation analysis between state-of-the-art underspampling approaches with the proposed approach- JUIndoorLoc and Combined dataset is used for the analysis

racy values on both the datasets. kNN classifier is used in the cost metric of the proposed BPSO based selection approach. The comparison of accuracy values shows that the proposed selection approach is performing better than both of the undersampling approaches. Although the near miss undersampling is performing better in case of the collected combined set for which the sample per class are well balanced as compared to JUIndoorLoc dataset. Figure 6.21 presents the percentage of data reduction in a bar plot. It can be observed that CONN has reduced the data samples by 85% while the accuracy metric is also reduced by 30%. The error deviation is presented in Figure 6.22. The proposed selection strategy is found to perform better for both the JUIndooLoc and Combined device datasets, as shown in the figure as it retains the important data points through consideration of instance hardness.

Table 6.12: Comparative analysis of accuracy parameter with state-of-the-art undersampling approaches. For the proposed selection approach kNN is used as the fitness metric.

(a) Comparative Analysis of accuracy parameter on collect Combined Dataset			
Classifier	**CONN**	***Near Miss***	***Proposed Selection Approach***
Decision Tree	*51.4*	90.9	92.4
kNN	*32.5*	82.2	83.8
SVM	*57.8*	92.3	93.3
(b) Comparative Analysis of accuracy parameter on collect JUIndoorLoc Dataset			
Classifier	***CONN***	***Near Miss***	***Proposed Selection Approach***
Decision Tree	42.4	89.8	97.19
kNN	24.8	81.7	93.84
SVM	51.8	92.7	98.02

6.4 Summary

An incorrectly labelled fingerprint in the context of indoor localization frequently perturbs the prediction deviation from the actual grid point. Therefore, a significant reduction of such fingerprints can significantly lower the deviation and increase the localization system's effectiveness. Additionally, the process of edge-level training benefits from the decrease and balancing of dataset sample sizes.

In this chapter, the meta-heuristic based instance selection approach has been explored in two ways. The GA based approach involves a hybrid tuple selection pipeline that works on selecting the instances with the objective of preserving the accuracy metric. In BPSO based selection, the fitness metric is manifold and the inclusion of KDN measure analyzes the selected instances for probable outlier. It can be observe from the findings that the proposed two-phased and single-phased approach, both are capable of achieving appreciable instance reduction without significant drop in the accuracy metric. It can also be observed that on application of the reduction pipeline the improvement of the error deviation from its actual grid point is quite significant. This indicates a clear de-selection of the outliers during the reduction process. Both the approaches have been validated and comparison on benchmark dataset has been carried out.

Published Journal:

1. **Panja, A.K.**, Rayala, A., Agarwala, A., Neogy, S. and Chowdhury, C., 2023. A

hybrid tuple selection pipeline for smartphone based Human Activity Recognition. Expert Systems with Applications, 217, p.119536. *(IF : 8.5)*

Communicated Journal:

1. **Panja, A.K.**, Karim, F., Neogy, S., Chowdhury, C. A Novel Exploratory Instance Selection Approach for RSS based Indoor Localization Datasets. Submitted to Expert Systems with Application, Elsevier *(IF: 8.5)*

Chapter 7

Conclusion and Future Scope

Due to its widespread use in industries like security, healthcare, and retail, indoor localization has emerged as a crucial area of research in recent years. Due to the prevalence of Wi-Fi networks in indoor environments, the use of Wi-Fi signals for indoor localization has grown in popularity. Achieving accurate and robust device localization in indoor environments poses a significant challenge due to the issues, such as - multipath propagation, signal attenuation, and signal interference. In the previous chapters we have proposed solutions tackling the problems of device heterogeneity, contextual context, important access point selection, outlier removal. Approaches such as wrapper-based and filter-based in both the aspect of features which in this case are the access points and instances were discussed and proposed.

The final thoughts and a review of the works cited in this thesis are presented in this chapter. The results are examined for their overall relevance. Future extensions of this study are also recommended in a few other directions.

7.1 Summary

A summarized form of the work, its associated finding and the inferences drawn are presented as below:

- Meta-heuristic & Deep learning approaches in the domain of dimensionality reduction has been carried out in the domain of Wi-Fi based Indoor Positioning.

- Dimensionality reduction has been explored both in the feature space as well as the instance space. User dataset has been collected using 4 smartphone devices upon which the experimentation and analysis has been carried out. The data collection for manual site survey is carried out through an android application and a client side application has been built for localization.

- In feature selection, meta-heuristic based approaches has been emphasized considering the accuracy metric as well as the intrinsic characteristics within the AP sets. Through experimentation on feature selection problem, it has been observed that the problem of AP selection is a multimodal problem. Hence, there lies more than one subset of APs with similar performance metric. To capture the varying environmental context and train the learning model using the multiple subsets of APs, a feature-based ensemble learning is proposed.

- In the instance space, meta-heuristic selection approaches are proposed that select the important representative subset of the original data preserving class boundaries. Work has also been focussed in detection and de-selecting the noisy instances that are likely to be misclassified.

- All results have statistical stability which has been tested using multiple benchmark datasets taken from different experimental setups in university, shopping mall, etc. Testing on related sensing modalities such as activity dataset has also been carried out.

- The objective of the work is to contribute to sustainable localization. Sustainable localization can only be achieved through training in constrained edge environment. Hence, testing on Raspberry Pi has been carried out for practical training in edge devices and the CPU and memory usages has been validated.

7.2 Summary of Contributions

In the present work each chapter is structured to build on the information presented in the preceding chapters, providing a holistic understanding of dimensionality reduction in

indoor localization and outlining avenues for future research in this field. The chapters are summarized in the following manner.

- In chapter 3 we have introduced Wrapper Based AP selection with the use of Genetic Algorithm(GA). A meta-heuristic feature association based training pipeline is proposed in this chapter. Throughout the paper we have experimented and carried out study on the multi-modal effect of the Wi-Fi RSS. The test on device heterogeneity with training/testing was carried out on different devices. An accuracy of more than 90% was achieved which verifies the ensemble model's stability towards device sensitivity.

- Chapter 4 extends the work laid out in chapter 3. In this chapter we have proposed a Filter Based AP selection by exploring binary Particle Swarm Optimization. A feature-based stacking ensemble model is demonstrated. We have focussed more on the robustness aspect of the localization process. The proposed AP selection process was able to reduce the number of APs by more than 50%, and the feature-based ensemble was able to achieve an accuracy of more than 96%. Exhibiting enhanced performance on benchmark datasets, our model compares favorably with recent published works in the field.

- Chapter 5 explores dimensionality reduction through deep learning approach. In this chapter we have studied the boundary instances and their importance in localization process by introducing k-disagreeing neighbor score. Two Channel feature representation by inclusion of disagreeing score and feature extraction using Convolutional Autoencoding(CAE) process have been proposed. The training process has been carried out on a proposed CNN architecture on the latent space representation of the CAE. The proposed approach is performing better than the modern DNN architectures both with respect to accuracy(98%) as well as error deviation($2.43m$).

- The problem on instance selection is targeted in Chapter 6, which is another aspect of dimensionality reduction. We have investigated the applicability of the proposed idea to other smartphone sensing application beyond ILS such as Human Activity Recognition(HAR). The extension of work on k-disagreeing score and other instance

hardness metric has been carried out in this chapter. We have proposed modified meta-heuristic approaches that can effectively detect and remove outliers or irrelevant data instances, thereby enhancing the model's generalization ability and prediction performance. An observed reduction in the error deviation metric from $4m$ to $3.2m$ for the collected combined dataset states the efficacy of the instance selection process and the reduction in outliers.

7.3 Future Research Directions

The area of indoor localization, particularly its dimensionality reduction, presents a plethora of opportunities for further exploration and improvement. Through experimentation and the subsequent hurdles followed in carrying out the research, few direction in research requires further exploration.

7.3.1 Seamless Indoor and Outdoor Localization through sensor fusion

To achieve seamless positioning, it is essential to detect the different environmental context and its associated proper switching from indoor to outdoor navigation applications. This can only be achieved through sensor fusion. To create a more complete and accurate picture of the environment for smartphone based localization, combining Bluetooth, WiFi, and inertial data from mobile devices is essential.

In order to create a seamless positioning paradigm, it is customary to understand indoor, semi-indoor and deep-indoor environment. One of the approaches involves distinctly dividing the indoor and outdoor transition into finer granular grid points near the transition areas. Utilizing machine learning to predict the context can be plausible solution.

Context:

$$Context = Model(Acc, Mag, Light, WiFi, GPS)$$

In the aspect of seamless navigation fusion of WiFi RSS with inertial sensing can be a plausible solution. Probabilistic approaches such as Extended Kalman Filter (EKF) or Bayesian Filters can be adopted for estimating the next state/coordinate on the floor.

7.3.2 Exploring Generative Models to tackle the problem of manual site survey

In the domain of Wi-Fi RSS based ILS, the reliance on labor-intensive site surveys persists as a bottleneck, hindering scalability and adaptability. Hence, the endeavour to transcend the limitations of manual site surveys presents an intriguing trajectory for future research. Generative Adversarial Networks (GANs) can be a promising avenue for advancement to minimize the process of manual surveys.

GANs, in the realm of machine learning, offers a compelling framework for generating synthetic data [118, 119] that mimics the underlying distribution of real-world data. In the context of WiFi RSS-based indoor localization, GANs can be a plausible solution to manual site survey by creating synthetic RSS fingerprints that mirror the intricate patterns and nuances observed in actual environments. A simple representation of the application of GAN can be as follows. The radiomap dataset $R =< rss_{i1}, ..., rss_{i,m}, Y_i >$ as defined in the previous chapters, where F_i represents WiFi RSS fingerprints $F_i = [f_{i1}, f_{i2}, \ldots, f_{im}]$ and Y_i represents the corresponding class label (floor grids). The two important component on GAN are stated below.

Generator G:

$$\text{Synthetic fingerprint: } F = G(z)$$

Discriminator D:

$$\text{Discriminator output: } D(F_i) \text{ for real data}, \quad D(\hat{F}) \text{ for synthetic data}$$

GANs [120] leverages its inherent capacity to learn complex high-dimensional structures, the learning process can be represented as follows:

Generator Learning G: The objective of the generator is to minimize the discrepancy between the synthetic fingerprints generated by generator and the real fingerprints in the dataset.

$$\min_{G} \mathcal{L}_G(F_i, \hat{F}_i)$$

2. Discriminator Learning D: The discriminator works with the objective on maximizing the ability to differentiate between the real and fake fingerprints.

$$\max_{D} \mathcal{L}_D(D(F_i), D(\hat{\mathbf{F}}_\mathbf{i}))$$

This representation encapsulates the GAN framework for generating synthetic fingerprints from the radiomap dataset for indoor localization using WiFi RSS data.

7.3.3 Exploring sequential learning through Transformer Model in the domain of Indoor Localization

The problem with classification of discrete grid points in ILS using machine learning is that the temporal aspect is not considered. In the past, few of the works have been carried out in the domain of sequential learning. To record temporal patterns in RSS sequences Zhang et al. [121] proposed a DNN architecture with Hidden Markov Model(HMM). CNN architectures [122] have also been explored in the very domain of sequential learning. Khassanov et al. in [123] published a sequential ILS dataset containing 290 trajectories over a $9000m^2$ area. In the pursuit of enhancing indoor localization methodologies utilizing WiFi Access Point (AP) data, the potential integration of Transformer Models for sequential learning presents an intriguing avenue for future research.

Transformers have demonstrated remarkable success in various sequential data tasks, particularly in natural language processing and time-series analysis. Leveraging the inherent sequential nature of WiFi signal data in Indoor Localization, the adoption of Transformer-based models may offer a unique opportunity to capture intricate temporal dependencies and spatial patterns present in these signals. Hence, exploring transformer model in the domain of indoor localization can be a good future scope of research.

Bibliography

[1] Yu Liu and Shijun Tian. Research of indoor gps signals acquisition algorithm. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4. IEEE, 2008.

[2] Paolo Barsocchi, Stefano Chessa, Alessio Micheli, and Claudio Gallicchio. Forecast-driven enhancement of received signal strength (rss)-based localization systems. *ISPRS International Journal of Geo-Information*, 2(4):978–995, 2013.

[3] Yuan Zhuang, Jun Yang, You Li, Longning Qi, and Naser El-Sheimy. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors*, 16(5):596, 2016.

[4] Finn Zhan Chen. Indoor proximity-based advertising using bluetooth beacons.

[5] Merlin Samuel, Noufal Nazeem, P Sreevals, Resmi Ramachandran, and P Careena. Smart indoor navigation and proximity advertising with android application using ble technology. *Materials Today: Proceedings*, 43:3799–3803, 2021.

[6] Aditi Adhikari, Vincent W Zheng, Hong Cao, Miao Lin, Yuan Fang, and Kevin Chen-Chuan Chang. Intelligshop: enabling intelligent shopping in malls through location-based augmented reality. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1604–1607. IEEE, 2015.

[7] Carman Ka Man Lee, CM Ip, Taezoon Park, and SY Chung. A bluetooth location-based indoor positioning system for asset tracking in warehouse. In *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1408–1412. IEEE, 2019.

[8] SJ Hayward, Kate van Lopik, Christopher Hinde, and AA West. A survey of indoor location technologies, techniques and applications in industry. *Internet of Things*, page 100608, 2022.

[9] Eun Yi Kim. Wheelchair navigation system for disabled and elderly people. *Sensors*, 16(11):1806, 2016.

[10] George Dimas, Dimitris E Diamantis, Panagiotis Kalozoumis, and Dimitris K Iakovidis. Uncertainty-aware visual perception system for outdoor navigation of the visually challenged. *Sensors*, 20(8):2385, 2020.

[11] Soo-Cheol Kim, Young-Sik Jeong, and Sang-Oh Park. Rfid-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and ubiquitous computing*, 17:1699–1707, 2013.

[12] Hongtai Cheng, Heping Chen, and Yong Liu. Topological indoor localization and navigation for autonomous mobile robot. *IEEE Transactions on Automation Science and Engineering*, 12(2):729–738, 2014.

[13] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, volume 2, pages 775–784. Ieee, 2000.

[14] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 205–218, 2005.

[15] Maxim Shchekotov. Indoor localization method based on wi-fi trilateration technique. In *Proceeding of the 16th conference of fruct association*, pages 177–179, 2014.

[16] Mohd Ezanee Rusli, Mohammad Ali, Norziana Jamil, and Marina Md Din. An improved indoor positioning algorithm based on rssi-trilateration technique for internet

of things (iot). In *2016 International Conference on Computer and Communication Engineering (ICCCE)*, pages 72–77. IEEE, 2016.

[17] Zengshan Tian, Zhongchun Wang, Ze Li, and Mu Zhou. Rtil: A real-time indoor localization system by using angle of arrival of commodity wifi signal. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2019.

[18] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. Usenix, 2013.

[19] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184, 2010.

[20] Imran Ashraf, Soojung Hur, and Yongwan Park. Magio: Magnetic field strength based indoor-outdoor detection with a commercial smartphone. *Micromachines*, 9(10):534, 2018.

[21] Pothuri Surendra Varma and Veena Anand. Random forest learning based indoor localization as an iot service for smart buildings. *Wireless Personal Communications*, 117:3209–3227, 2021.

[22] Wenzhe Zhang, Lei Wang, Zhenquan Qin, Xueshu Zheng, Liang Sun, Naigao Jin, and Lei Shu. Inbs: An improved naive bayes simple learning approach for accurate indoor localization. In *2014 IEEE International Conference on Communications (ICC)*, pages 148–153. IEEE, 2014.

[23] Ahmed H Salamah, Mohamed Tamazin, Maha A Sharkas, and Mohamed Khedr. An enhanced wifi indoor localization system based on machine learning. In *2016 International conference on indoor positioning and indoor navigation (IPIN)*, pages 1–8. IEEE, 2016.

[24] YuFeng, JiangMinghua, LiangJing, QinXiao, HuMing, PengTao, and HuXinrong. An improved indoor localization of wifibased on support vector machines. *International Journal of Future Generation Communication and Networking*, 7(5):191–206, 2014.

[25] Lummanee Chanama and Olarn Wongwirat. A comparison of decision tree based techniques for indoor positioning system. In *2018 international conference on information networking (ICOIN)*, pages 732–737. IEEE, 2018.

[26] Esrafil Jedari, Zheng Wu, Rashid Rashidzadeh, and Mehrdad Saif. Wi-fi based indoor location positioning employing random forest classifier. In *2015 international conference on indoor positioning and indoor navigation (IPIN)*, pages 1–5. IEEE, 2015.

[27] Abebe Belay Adege, Yirga Yayeh, Getaneh Berie, Hsin-piao Lin, Lei Yen, and Yun Ruei Li. Indoor localization using k-nearest neighbor and artificial neural network back propagation algorithms. In *2018 27th Wireless and Optical Communication Conference (WOCC)*, pages 1–2. IEEE, 2018.

[28] Abhishek Goswami, Luis E Ortiz, and Samir R Das. Wigem: A learning-based approach for indoor localization. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, pages 1–12, 2011.

[29] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. Deepfi: Deep learning for indoor fingerprinting using channel state information. In *2015 IEEE wireless communications and networking conference (WCNC)*, pages 1666–1671. IEEE, 2015.

[30] Moustafa Abbas, Moustafa Elhamshary, Hamada Rizk, Marwan Torki, and Moustafa Youssef. Wideep: Wifi-based accurate and robust indoor localization system using deep learning. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, pages 1–10. IEEE, 2019.

[31] Mozi Chen, Kezhong Liu, Jie Ma, Xuming Zeng, Zheng Dong, Guangmo Tong, and Cong Liu. Moloc: Unsupervised fingerprint roaming for device-free indoor localization in a mobile ship environment. *ieee internet of things journal*, 7(12):11851–11862, 2020.

[32] Elina Laitinen and Elena Simona Lohan. On the choice of access point selection criterion and other position estimation characteristics for wlan-based indoor positioning. *Sensors*, 16(5):737, 2016.

[33] Hongyu Shi. A new weighted centroid localization algorithm based on rssi. In *2012 IEEE International Conference on Information and Automation*, pages 137–141. IEEE, 2012.

[34] Moustafa A Youssef, Ashok Agrawala, and A Udaya Shankar. Wlan location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, pages 143–150. IEEE, 2003.

[35] Pei Jiang, Yunzhou Zhang, Wenyan Fu, Huiyu Liu, and Xiaolin Su. Indoor mobile localization based on wi-fi fingerprint's important access point. *International Journal of Distributed Sensor Networks*, 11(4):429104, 2015.

[36] Yiqiang Chen, Qiang Yang, Jie Yin, and Xiaoyong Chai. Power-efficient access-point selection for indoor location estimation. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):877–888, 2006.

[37] Pengyu Huang, Haojie Zhao, Wei Liu, and Dingde Jiang. Maps: Indoor localization algorithm based on multiple ap selection. *Mobile Networks and Applications*, pages 1–8, 2020.

[38] Zhongyuan Wang, Zijian Wang, Li Fan, and Zhihao Yu. A hybrid wi-fi fingerprint-based localization scheme achieved by combining fisher score and stacked sparse autoencoder algorithms. *Mobile Information Systems*, 2020, 2020.

[39] Wei Meng, Wendong Xiao, Wei Ni, and Lihua Xie. Secure and robust wi-fi fingerprinting indoor localization. pages 1–7, 2011.

[40] Junhai Luo and Liang Fu. A smartphone indoor localization algorithm based on wlan location fingerprinting with feature extraction and clustering. *Sensors*, 17(6):1339, 2017.

[41] Lingwen Zhang, Yishun Li, Yajun Gu, and Wenkao Yang. An efficient machine learning approach for indoor localization. *China Communications*, 14(11):141–150, 2017.

[42] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[43] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[44] Siwei Feng and Marco F Duarte. Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation. *Neurocomputing*, 312:310–323, 2018.

[45] Qingyuan Zhao, Sheng Zhang, Xingchuan Liu, and Xiaokang Lin. An effective preprocessing scheme for wlan-based fingerprint positioning systems. In *2010 IEEE 12th International Conference on Communication Technology*, pages 592–595. IEEE, 2010.

[46] Priya Roy, Mausam Kundu, and Chandreyee Chowdhury. Indoor localization using stable set of wireless access points subject to varying granularity levels. In *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 491–496. IEEE, 2019.

[47] Tsung-Nan Lin, Shih-Hau Fang, Wei-Han Tseng, Chung-Wei Lee, and Jeng-Wei Hsieh. A group-discrimination-based access point selection for wlan fingerprinting localization. *IEEE Transactions on Vehicular Technology*, 63(8):3967–3976, 2014.

[48] Yen-Kai Cheng, Hsin-Jui Chou, and Ronald Y Chang. Machine-learning indoor localization with access point selection and signal strength reconstruction. pages 1–5, 2016.

[49] Song Xu, Wusheng Chou, and Hongyi Dong. A robust indoor localization system integrating visual localization aided by cnn-based image retrieval with monte carlo localization. *Sensors*, 19(2):249, 2019.

[50] Chen Xing, Li Ma, and Xiaoquan Yang. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *Journal of Sensors*, 2016, 2016.

[51] David Landgrebe. Hyperspectral image data analysis. *IEEE Signal processing magazine*, 19(1):17–28, 2002.

[52] Pengyu Huang, Haojie Zhao, Wei Liu, and Dingde Jiang. Maps: Indoor localization algorithm based on multiple ap selection. *Mobile Networks and Applications*, 26:649–656, 2021.

[53] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J Kennedy. Relational autoencoder for feature extraction. In *2017 International joint conference on neural networks (IJCNN)*, pages 364–371. IEEE, 2017.

[54] Yesi Novaria Kunang, Siti Nurmaini, Deris Stiawan, Ahmad Zarkasi, et al. Automatic features extraction using autoencoder in intrusion detection system. In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 219–224. IEEE, 2018.

[55] Zonghai Zhu, Zhe Wang, Dongdong Li, and Wenli Du. Nearcount: Selecting critical instances based on the cited counts of nearest neighbors. *Knowledge-Based Systems*, 190:105196, 2020.

[56] Fabrizio Angiulli. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd international conference on Machine learning*, pages 25–32, 2005.

[57] Lei Bao, Cao Juan, Jintao Li, and Yongdong Zhang. Boosted near-miss undersampling on svm ensembles for concept detection in large-scale imbalanced datasets. *Neurocomputing*, 172:198–206, 2016.

[58] Stefanos Ougiaroglou and Georgios Evangelidis. Efficient dataset size reduction by finding homogeneous clusters. In *Proceedings of the Fifth Balkan Conference in Informatics*, pages 168–173, 2012.

[59] Jingnian Chen, Caiming Zhang, Xiaoping Xue, and Cheng-Lin Liu. Fast instance selection for speeding up support vector machines. *Knowledge-Based Systems*, 45:1–7, 2013.

[60] Pablo Hernandez-Leal, J Ariel Carrasco-Ochoa, J Fco Martínez-Trinidad, and J Arturo Olvera-Lopez. Instancerank based on borders for instance selection. *Pattern Recognition*, 46(1):365–375, 2013.

[61] Derek A Pisner and David M Schnyer. Support vector machine. In *Machine learning*, pages 101–121. Elsevier, 2020.

[62] Peter Englert. Locally weighted learning. In *Seminar Class on Autonomous Learning Systems*. Citeseer, 2012.

[63] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[64] Joel Luis Carbonera and Mara Abel. A density-based approach for instance selection. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 768–774. IEEE, 2015.

[65] Duksan Ryu, Jong-In Jang, and Jongmoon Baik. A hybrid instance selection using nearest-neighbor for cross-project defect prediction. *Journal of Computer Science and Technology*, 30(5):969–980, 2015.

[66] Joel Luís Carbonera. An efficient approach for instance selection. In *International conference on big data analytics and knowledge discovery*, pages 228–243. Springer, 2017.

[67] Yunsheng Song, Jiye Liang, Jing Lu, and Xingwang Zhao. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251:26–34, 2017.

[68] Haizhou Du, Shengjie zhao, and Daqiang zhang. Robust local outlier detection. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 116–123, 2015.

[69] Hoang Vu Nguyen, Hock Hee Ang, and Vivekanand Gopalkrishnan. Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *International Conference on Database Systems for Advanced Applications*, pages 368–383. Springer, 2010.

[70] Arthur Zimek, Matthew Gaudet, Ricardo JGB Campello, and Jörg Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 428–436, 2013.

[71] Priya Roy, Chandreyee Chowdhury, Dip Ghosh, and Sanghamitra Bandyopadhyay. Juindoorloc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity. *Wireless Personal Communications*, 106(2):739–762, 2019.

[72] Joaquín Torres-Sospedra, Raúl Montoliu, Adolfo Martínez-Usó, Joan P Avariento, Tomás J Arnau, Mauri Benedito-Bordonau, and Joaquín Huerta. Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems. In *2014 international conference on indoor positioning and indoor navigation (IPIN)*, pages 261–270. IEEE, 2014.

[73] Fengxi Song, Zhongwei Guo, and Dayong Mei. Feature selection using principal component analysis. In *2010 international conference on system science, engineering design and manufacturing informatization*, volume 1, pages 27–30. IEEE, 2010.

[74] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

[75] Abdulsalam Alsmady and Fahed Awad. Optimal wi-fi access point placement for rssi-based indoor localization using genetic algorithm. In *2017 8th international conference on information and communication systems (ICICS)*, pages 287–291. IEEE, 2017.

[76] Baoding Zhou, Wei Tu, Ke Mai, Weixing Xue, Wei Ma, and Qingquan Li. A novel access point placement method for wifi fingerprinting considering existing aps. *IEEE Wireless Communications Letters*, 9(11):1799–1802, 2020.

[77] David E Goldberg. *Genetic algorithms*. pearson education India, 2013.

[78] Christopher R Houck, Jeff Joines, and Michael G Kay. A genetic algorithm for function optimization: a matlab implementation. *Ncsu-ie tr*, 95(09):1–10, 1995.

[79] Ali Karci. Novelty in the generation of initial population for genetic algorithms. In Mircea Gh. Negoita, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 268–275, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[80] Adam Lipowski and Dorota Lipowska. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196, 2012.

[81] Pietro S Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

[82] Waad Bouaguel. A new approach for wrapper feature selection using genetic algorithm for big data. In *Intelligent and Evolutionary Systems*, pages 75–83. Springer, 2016.

[83] Ang Li, Jingqi Fu, Huaming Shen, and Sizhou Sun. A cluster-principal-component-analysis-based indoor positioning algorithm. *IEEE Internet of Things Journal*, 8(1):187–196, 2020.

[84] Jiang Xiao, Kaishun Wu, Youwen Yi, and Lionel M Ni. Fifs: Fine-grained indoor fingerprinting system. In *2012 21st international conference on computer communications and networks (ICCCN)*, pages 1–7. IEEE, 2012.

[85] Mansour Sheikhan, Mahdi Bejani, and Davood Gharavian. Modular neural-svm scheme for speech emotion recognition using anova feature selection method. *Neural Computing and Applications*, 23(1):215–227, 2013.

[86] Han Zou, Yiwen Luo, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. A mutual information based online access point selection strategy for wifi indoor localization. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 180–185. IEEE, 2015.

[87] Xiao Chen and Shengnan Zou. Improved wi-fi indoor positioning based on particle swarm optimization. *IEEE Sensors Journal*, 17(21):7143–7148, 2017.

[88] Xuan Du and Kun Yang. A map-assisted wifi ap placement algorithm enabling mobile device's indoor positioning. *IEEE Systems Journal*, 11(3):1467–1475, 2016.

[89] Farrukh Shahzad, A Rauf Baig, Sohail Masood, Muhammad Kamran, and Nawazish Naveed. Opposition-based particle swarm optimization with velocity clamping (ovcpso). In *Advances in Computational Intelligence*, pages 339–348. Springer, 2009.

[90] Na Dong, Xing Fang, and Ai-guo Wu. A novel chaotic particle swarm optimization algorithm for parking space guidance. *Mathematical Problems in Engineering*, 2016, 2016.

[91] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[92] Hongtao Ye, Wenguang Luo, and Zhenqiang Li. Convergence analysis of particle swarm optimizer and its improved algorithm based on velocity differential evolution. *Computational intelligence and neuroscience*, 2013, 2013.

[93] Mojtaba Ahmadieh Khanesar, Mohammad Teshnehlab, and Mahdi Aliyari Shoorehdeli. A novel binary particle swarm optimization. In *2007 Mediterranean conference on control & automation*, pages 1–6. IEEE, 2007.

[94] Ayan Kumar Panja, Chandreyee Chowdhury, Priya Roy, Sakil Mallick, Sukanto Mondal, Soumik Paul, and Sarmistha Neogy. Designing a framework for real-time wifi-based indoor positioning. In *Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020*, pages 71–82. Springer, 2021.

[95] Georgios Douzas and Fernando Bacao. Geometric smote a geometrically enhanced drop-in replacement for smote. *Information Sciences*, 501:118–135, 2019.

[96] Lara Lusa and Rok Blagus. Evaluation of smote for high-dimensional class-imbalanced microarray data. In *2012 11th International Conference on Machine Learning and Applications*, volume 2, pages 89–94. IEEE, 2012.

[97] AlphaJi, EvanSong1220 BeaN, Qiang Xu inversion, Kumius, and Yuanchao Shu. Indoor location and navigation, 2021.

[98] Zhinong Jiang, Yuehua Lai, Jinjie Zhang, Haipeng Zhao, and Zhiwei Mao. Multi-factor operating condition recognition using 1d convolutional long short-term network. *Sensors*, 19(24):5488, 2019.

[99] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.

[100] Philip Tchatchoua, Guillaume Graton, Mustapha Ouladsine, Julien Muller, Abraham Traoré, and Michel Juge. 1d resnet for fault detection and classification on sensor data in semiconductor manufacturing. In *2022 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–6. IEEE, 2022.

[101] Isha Garg, Priyadarshini Panda, and Kaushik Roy. A low effort approach to structured cnn design using pca. *IEEE Access*, 8:1347–1360, 2019.

[102] H Cho and SM Yoon. Applying singular value decomposition on accelerometer data for 1d convolutional neural network based fall detection. *Electronics Letters*, 55(6):320–322, 2019.

[103] Álvar Arnaiz-González, Marcin Blachnik, Mirosław Kordos, and César García-Osorio. Fusion of instance selection methods in regression tasks. *Information Fusion*, 30:69–79, 2016.

[104] Li Ai-jun and Zhang Peng. Research on unbalanced data processing algorithm base tomeklinks-smote. In *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*, pages 13–17, 2020.

[105] Show-Jane Yen and Yue-Shi Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*, pages 731–740. Springer, 2006.

[106] Debashree Devi, Biswajit Purkayastha, et al. Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance. *Pattern Recognition Letters*, 93:3–12, 2017.

[107] J Arturo Olvera-López, J Ariel Carrasco-Ochoa, and J Martínez-Trinidad. A new fast prototype selection method based on clustering. *Pattern Analysis and Applications*, 13(2):131–141, 2010.

[108] Xin-She Yang. Metaheuristic optimization: algorithm analysis and open problems. In *International Symposium on Experimental Algorithms*, pages 21–32. Springer, 2011.

[109] Russell A Brown. Building a balanced kd tree in o (kn log n) time. *arXiv preprint arXiv:1410.5420*, 2014.

[110] Pedro Yuri Arbs Paiva, Camila Castro Moreno, Kate Smith-Miles, Maria Gabriela Valeriano, and Ana Carolina Lorena. Relating instance hardness to classification performance in a dataset: a visual approach. *Machine Learning*, pages 1–39, 2022.

[111] Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. An instance level analysis of data complexity. *Machine learning*, 95(2):225–256, 2014.

[112] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[113] David Sánchez-Rodríguez, Pablo Hernández-Morera, José Ma Quinteiro, and Itziar Alonso-González. A low complexity system based on multiple weighted decision trees for indoor localization. *Sensors*, 15(6):14809–14829, 2015.

[114] Feng Yu, Ming Hua Jiang, Jing Liang, Xiao Qin, Ming Hu, Tao Peng, and Xin Rong Hu. An indoor localization of wifi based on support vector machines. In *Advanced Materials Research*, volume 926, pages 2438–2441. Trans Tech Publ, 2014.

[115] Lu Xuanmin, Qiu Yang, Yuan Wenle, and Yang Fan. An improved dynamic prediction fingerprint localization algorithm based on knn. In *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 289–292. IEEE, 2016.

[116] Uzair Ahmad, Andrey Gavrilov, Uzma Nasir, Mahrin Iqbal, Seong Jin Cho, and Sungyoung Lee. In-building localization using neural networks. In *2006 IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6. IEEE, 2006.

[117] Zeping Yang and Daqi Gao. An active under-sampling approach for imbalanced data classification. In *2012 Fifth International Symposium on Computational Intelligence and Design*, volume 2, pages 270–273. IEEE, 2012.

[118] Jun-Hyung Kim and Youngbae Hwang. Gan-based synthetic data augmentation for infrared small target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.

[119] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 289–293. IEEE, 2018.

[120] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

[121] Wei Zhang, Kan Liu, Weidong Zhang, Youmei Zhang, and Jason Gu. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing*, 194:279–287, 2016.

[122] Mai Ibrahim, Marwan Torki, and Mustafa ElNainay. Cnn based indoor localization using rss time-series. In *2018 IEEE symposium on computers and communications (ISCC)*, pages 01044–01049. IEEE, 2018.

[123] Yerbolat Khassanov, Mukhamet Nurpeiissov, Azamat Sarkytbayev, Askat Kuzdeuov, and Huseyin Atakan Varol. Finer-level sequential wifi-based indoor local-

ization. In *2021 IEEE/SICE International Symposium on System Integration (SII)*, pages 163–169. IEEE, 2021.