JADAVPUR UNIVERSITY

MASTER DEGREE PROJECT

Missing Value Imputation for the Particulate Matter Concentration Using Machine Learning

A project submitted in fulfilment of the
requirements for the degree of
Master of Computer Application

in

Computer Science and Engineering
Jadavpur University

By

Dipak Maity
University Roll No.: 001910503031
University Registration No.: 149891 of 2019-20
Exam Roll No.: MCA226030

Department of Computer Science and Engineering

Under the Guidance of

Dr. Sarbani Roy
Professor
Department of Computer Science and Engineering
Faculty of Engineering and Technology,
Jadavpur University, Kolkata
June 2022

Declaration of Authorship

I, Dipak Maity, declare that this project titled, “Missing Value Imputation for the Particulate Matter Concentration Using Machine Learning” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for research degree at this University.
- Where any part of this project has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely my own work.
- I have acknowledged all main sources of help.
- Where the project is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:__________________________

Date: ____________________________

To whom it may concern

This is to certify that the work on this thesis entitled "Missing Value Imputation for the Particulate Matter Concentration Using Machine Learning" has been satisfactorily completed by Dipak Maity, Roll No: 00910503031, University Registration No.: 149891 of 2019-20. It is a bona-fide piece of work carried out under my supervision at Jadavpur University, Kolkata-700032, for partial fulfillment of the requirements for the degree of Master of Computer Application from the Department of Computer Science and Engineering, Jadavpur University for the academic session 2019-2022.

Prof. Sarbani Roy
(Supervisor)
Department of Computer Science & Engineering
Jadavpur University

Prof. Anupam Sinha
Head of The Department
Department of Computer Science & Engineering
Jadavpur University

Prof. Chandan Majumder
Dean
Faculty of Engineering & Technology
Jadavpur University

Certificate of Approval

(Only in case the project is approved)

This is to certify that the thesis entitled "Missing Value Imputation for the Particulate Matter Concentration Using Machine Learning" is a bona-fide record of work carried out by Dipak Maity in fulfilment of the requirements for the award of the degree of Master of Computer Application from the Department of Computer Science and Engineering, Jadavpur University during the period of January 2022 to June, 2022. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Examiners:

____

(Signature of Examiner)

Date:

____

(Signature of Examiner)

Date:

Jadavpur University

Abstract

Faculty of Engineering and Technology, Jadavpur University
Department of Computer Science and Engineering
Master of Computer Application

by Dipak Maity

For the past few years, Delhi has been considered one of the most polluted cities in India. The people of here suffer from air pollution related diseases (Asthma, respiratory inflammation etc.) almost all the year round. The amount of air pollution is much higher for urbanization. Harmful pollutants like NO_2, SO_2, ground-level O_3, PM2.5, PM10 are abundant in the air here. Of which PM2.5 is one of the harmful pollutants. It is very important to determine the concentration of these elements in the air to reduce pollution. Although many monitoring stations have been set up for this purpose, they often fail to provide accurate information. In this work concentration of PM2.5 has been predicted with the help of some machine learning and deep learning models. Concentration of PM2.5 has been taken at one-hour intervals for a whole year in twenty-eight cities of Delhi. In order to get the best accuracy, the models have been hyperparameter-tuned. Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R2-Score metrices has been used to measure accuracy. Feature selection has also been done using different methodologies to see if better results are available. K-Nearest Neighbor (KNN) algorithm for regression has given us the best accuracy in this work.

Keywords: *Machine Learning, Regression, Hyperparameter tuning, Time series analysis.*

Acknowledgements

On the submission of “Missing Value Imputation for the Particulate Matter Concentration Using Machine Learning”, I wish to express gratitude to the Department of Computer Science and Engineering for sanctioning a thesis work under Jadavpur University under which this work has been completed. I would like to express my sincere gratitude to my respected guide Dr. Sarbani Roy, Professor, Department of Computer Science and Engineering, Jadavpur University for her unfailing guidance, prolific encouragement, constructive suggestions and continuous involvement during each and every phase of this research work. I feel deeply honoured that I got the opportunity to work under her guidance. I would like to express my heartfelt gratitude to Asif Iqbal Middya, Research Fellow, Jadavpur University, Kolkata, for his suggestions and unwavering support. I would also wish to thank Prof. Anupam Sinha, Head of Department of Computer Science and Engineering, Jadavpur University, Prof. Chandan Majumder, Dean of Faculty of Engineering and Technology, Jadavpur University for providing me all the facilities and for their support to the activities of this research. I would like to express my gratitude and indebtedness to my parents and all my family members for their unbreakable belief, constant encouragement, moral support and guidance. Last, but not the least, I would like to thank all my classmates of Master of Computer Application batch of 2019-2022, for their co-operation and support. Their wealth of experience has been a source of strength for me throughout the duration of my work.

Regards,
Dipak Maity
University Roll Number: 001910503031
University Registration No.: 149891 of 2019-20
Department of Computer Science and Engineering
Jadavpur University

Contents

List of Figures

List of Abbreviations

PM	**P**articulate **M**atter
CNN	**C**onvolutional Neural Network
ANN	**A**rtificial **N**eural **N**etwork
KNN	**K**- **N**earest **N**eighbor
RMSE	**R**oot **M**ean **S**quare **E**rror
MAE	**M**ean **A**bsolute **E**rror
MI	**M**utual **I**nformation

In Dedication to my family for supporting me all the way!

Chapter 1

Introduction

1.1 Overview

At present the rate of increase in the amount of toxic gas (O_3, NO, NO_2, etc.) and Particular Matters (PM) in the air become a major problem. The main reason for this is urbanization. As a result, many kinds of diseases appear in the human body. Monitors have been installed in various parts of the city to measure the concentration of these pollutants in the air. Since these are the electronics devices, they may not always work properly. As a result, accurate information is sometimes not available. Many have worked before to solve this problem, predicting the concentration of pollutants using various Machine Learning models. But they have used much less model and put more emphasis on deep learning models. In this work, we have tested on for machine learning models (K-nearest neighbors, Random Forest, Gradient Boosting, Extreme Gradient Boosting) and two deep learning model (Artificial neural network, Convolutional Neural network). We have worked on concentration of PM2.5. We have also focused more on appropriate feature selection. We have done this with the help of five methods (Radius-Based, Triangulation, Clustering, Mutual Information, Kl-Divergence).

1.2 Motivation

As the amount of toxic gas in the air increases, the air quality decreases. PM2.5 is one of the most harmful pollutants. Atmospheric particulate matter with a diameter less than 2.5 are considered as PM2.5 pollutant. Asthma, impair of lung function, various cardiovascular diseases when its amount in the air increases. Our responsibility as human beings to make the people of society aware of this. Many areas have been monitored for awareness. But sometimes, monitoring stations are not able to give the correct value. If a monitoring station goes bad, then the correct idea about the air in that area is not available. Different models have been used to know the air quality of those stations. But they also had many limitations. In our work we have adopted some new methods, such as proper feature selection for a particular station.

1.3 Contribution

We have basically worked with PM2.5 pollutant. Our main tasks were to find the best set of parameters, find the best model, find the best approach for feature selection. We worked on huge datasets, with 9504 datapoints for each station. First, we tried to find the best set of parameters for the models. To get better results, we have used five popular methodologies for feature selection. We have also tested our dataset on many models. Our main contribution in this work are

- A number of machine learning and deep learning models have been used for monitoring the missing value of PM2.5 concentration.
- The study has been performed with real world PM2.5 dataset containing more than 250000 datapoints.
- In addition, a number of methodologies has been adopted for feature selection (I.e., site/station selection).

1.4 Organization

The rest of the project is organized as follows. Chapter 2 discusses the work that has been done before related to our project and a detailed discussion of the models. Problem statement is given first in Chapter 3, Data collection and data preparation are discussed. A workflow has also been given with brief discussion. Then we write about proposed approaches. Final outputs are analyzed in Chapter 4. This section discusses what performance metrics are used and accuracy of the models. Conclusion has been given in Chapter 5.

Chapter 2

Preliminaries

In this chapter, we have presented the existing work related to missing value imputation in air pollution time series. Moreover, the models used for missing value imputation are discussed.

2.1 Related Work

Many studies have been done before to predict PM2.5 in air. The ensemble-kNN technique was used by M. Yang and et al. [10] to anticipate the monthly medium- to a long-term runoff on Danjiangkou Reservoir in China. They demonstrated that their model is effective and dependable. Guan & Sinnott [11] conducted a study on air pollution prediction using machine learning approaches. They used LSTM (Long Short-Term Memory) Networks to analyze data on air pollution in Melbourne, Australia. The LSTM network has been found to be fairly capable of detecting PM2.5 concentrations in the air. Joharestani et al. [12] present a few machine learning-based methods for PM2.5 prediction. On multisource remote sensing data, they used XGBoost, Random Forest, and deep learning to estimate PM2.5 polluting particles in Tehran's metropolitan region, Iran. In terms of R2-Score, MAE, and RMSE values, XGBoost is the highest performing model in contrast to the other two [11]. Five air pollution stations in Tehran were utilized by Shamsoddini et al. [13]. Using an artificial neural network and a random forest, researchers used and meteorological data to forecast PM2.5. They employed a built-in Random Forest (RF) function to attain a maximum value of R2 = 0.49.an estimate of the importance of a characteristic

Nabavi et al. [14] attempted to determine the geographical distribution of PM2.5 using (Aerosol Optical Depth) AOD10 and 1 km MAIAC data, over Tehran. R2 = 0.68 was the highest value they could get. They said that the Dark target algorithms based on the brightness of scenes as an indicator of aerosel. A hybrid CNN-LSTM model was developed by Taoying et al. [15] for predicting the next 24h PM2.5 concentration in Beijing, They have used the last 7days data. Their model gives better result for small training time. Jiang et al. [16] develop a deep temporal convolutional neural network (DeepTCN) to forecast PM2.5 concentration by modeling the data patterns of historical pollutant concentrations, meteorological data, and discrete time variables' data. Their model has enhanced the capacity to model PM-related factor data patterns and may be utilized as a potential tool for PM concentration predicting. Singh et al. [17] estimate the concentration of PM2.5 in Delhi's atmosphere. They had considered atmospheric and surface factors like wind speed, temperature, pressure, etc. They have proposed the extreme gradient boosting model as the best model for predicting concentration of PM2.5.

2.2 Popular Models for prediction

A machine learning model is a piece of code. That are used to identify specific patterns. You train a model on a set of data and give it an algorithm to use to find the nature of the dataset. It is used to get some idea about data it hasn't seen before and make predictions about it once you've trained it. For our project, we use KNN, Random Forest, Boosting, and Neural network Algorithm to predict data.

2.2.1. k-Nearest Neighbor: The k-NN is a machine learning algorithm that can perform both regression and classification issues[1].KNN regression is called non-parametric method. In case regression predict the test data by calculating an average of the numerical value of k nearest neighbors. Analyst set the size of the neighborhood. To find the k nearest neighbors it uses some distance method (Euclidean, Manhattan, Minkowski). In this project, we predict missing value with the help of kNN regression.

2.2.2. Random Forest: Using an ensemble learning approach and a large number of decision trees Random Forest Models are constructed. It also can be used for both regression and classification. It was proposed by Ho [2]. In the case of time series data, we have to transform it into a supervised learning problem. To conduct a better Random Forest model, we need to tune the parameters. For Random Forest regression, we used RandomForestRegressor() model. It gives an average of the predictions made by the trees in the forest. Random forest algorithm is slower in computation. 'n_estimator','max_feature' hyperparameters are used to increase the predictive power.

2.2.3. Gradient Boosting: Gradient boost is a machine learning algorithm that uses the 'Boosting' ensemble approach. Gradient boost, like other boosting models, combines numerous weak learners sequentially to generate a strong learner. Gradient boost often uses decision trees as weak learners. It achieves low bias and low variance in this way. All of the trees are connected in a succession, with each tree attempting to reduce the mistake of the one before it. Because of this sequential link, boosting algorithms are often slow to train (which the developer may modify using the learning rate option), but they are also quite accurate.

2.2.4. Extreme Gradient Boosting: Xtreme Gradient Boosting (XGBoost) is a popular machine learning package based on Tianqi Chen's gradient boosting technique [3]. In comparison to previous methods, it has greater control over overfitting by adopting a more regularised model formalization. It has a high success rate. especially in Kaggle contests for structured features [3]. XGBoost contains a lot of hyperparameters, but we're only going to use four of them here. Choosing the optimum values for these hyperparameters might result in a large boost in one's ultimate score, thus selecting the ideal values is crucial. To accomplish so, we'll use the scikit-learn GridSearchCV method to execute a cross-validated grid search. It's worth noting that this can take a long time because it tests every possible hyperparameter combination, meticulously looking for the optimal outcome.

2.2.5 Artificial neural network: Artificial neural networks (ANN) are a type of machine learning approach in which a machine learns patterns of varied complicated situations (responses) for future predictions using a collection of linked units (neurons) [4]. ANN follows the working process of neurons in human brain. Neural networks are constructed with the help of a set of neurons that are placed in layers. Those layers are called the input layer, hidden layer, and output layer [5]. Although there is no set of rules for how many layers there will be [5]. Each layer has an activation function attached to each of the neurons. It is responsible for producing non-linearity in the relationship. ANN works in two phases, forward propagation and backward propagation. The process of adding weights after multiplying them by each feature is known as forward propagation. The outcome also includes the bias. The process of updating the model's weights is known as backward propagation. An optimization function and a loss function are needed for backward propagation. It has been found that ANN models are powerful models for complex problems and prediction. We used only one hidden layer, since it was giving same the result for more than one hidden layer. We have compiled it with ‘rmsprop’ optimizer.

2.2.6. Convolutional Neural Network: Convolutional Neural Network (CNN) is also another powerful deep learning method. A CNN is made up of an input layer, an output layer, and numerous hidden layers in between, similar to other neural networks. These layers do operations on the data in order to learn characteristics unique to the data. It has three types of network structures 1D CNN, 2D CNN, and 3D CNN [6]. 1D CNN is used for sequence data processing [8], 2D CNN is often used for image and text recognition [7], and 3D CNN is used for medical image and video data recognition [9]. So we are using 1D CNN. Keras provides the conv1D class. We added flat and dense layers and compiled it with optimizers.

Chapter 3

Proposed Methodology

In this chapter we have discussed the methodology to address the problem of missing value imputation in particulate matter time series data. First a problem statement is provided. After that the details of workflow, dataset and proposed approach is illustrated.

3.1 Problem Statement

The main goal of this study is to find the best machine learning & deep learning models for predicting the concentration of PM2.5 in the air with the help of multi-time series data. We have been given the concentration of PM2.5 by monitoring twenty-eight stations in Delhi. Information has been given for a whole year with an interval of one hour. Now suppose a station is unable to give the right information about the pollutant PM2.5. To solve this problem, we have to work in two ways. The first one is to find which stations can be taken as a neighbor of that station so that by monitoring these stations we can predict the missing value of PM2.5 for that station. For this, we have to implement different methodologies like Triangulation, Clustering, Kl-Divergence, Mutual Information, etc. The second one is to find the best models for prediction, which can be done by comparing the predicted values with actual values.

3.2 Workflow

In fig 3.1 we have given a workflow of our whole work. The main goal of our work was to find the best model, find the best set of parameters of the models and determine the right features. we did it in two steps. First, we tried to find the best set of parameters for the models. We have imputed the missing value of PM2.5 with them. Then we did feature selection with the help of some popular methods to get good results. Then again, we predicted the missing value with those new datasets. For our work, we have taken the concentration of PM2.5 in twenty-eight cities of Delhi over one year, with an interval of one hour. Our data set had no null value. The details are discussed in section 4.3. We have taken a station named Shadipur as a target station. Because according to the map of all the cities, that we have collected data, Shadipur is located at their center. Worked with more ensemble learning based models. At first, we worked with the whole dataset, that is, we did not omit any feature. We have split the dataset into Train Set and Test Set. We vary the test set containing 10% to 40% data with an interval of 5%. Then for our work, we have used some popular machine learning and deep learning models, which are discussed in section 2.2. The best set of parameters has been found before making any prediction. To find the set of best parameters for each models, we have used the GridSearchCv function which belongs to Scikit-learn's(or SK-learn) model selection package. By which we can get a better result from the results we get using default parameters. Some models have done hyperparameter tuning manually, as it took a long time.

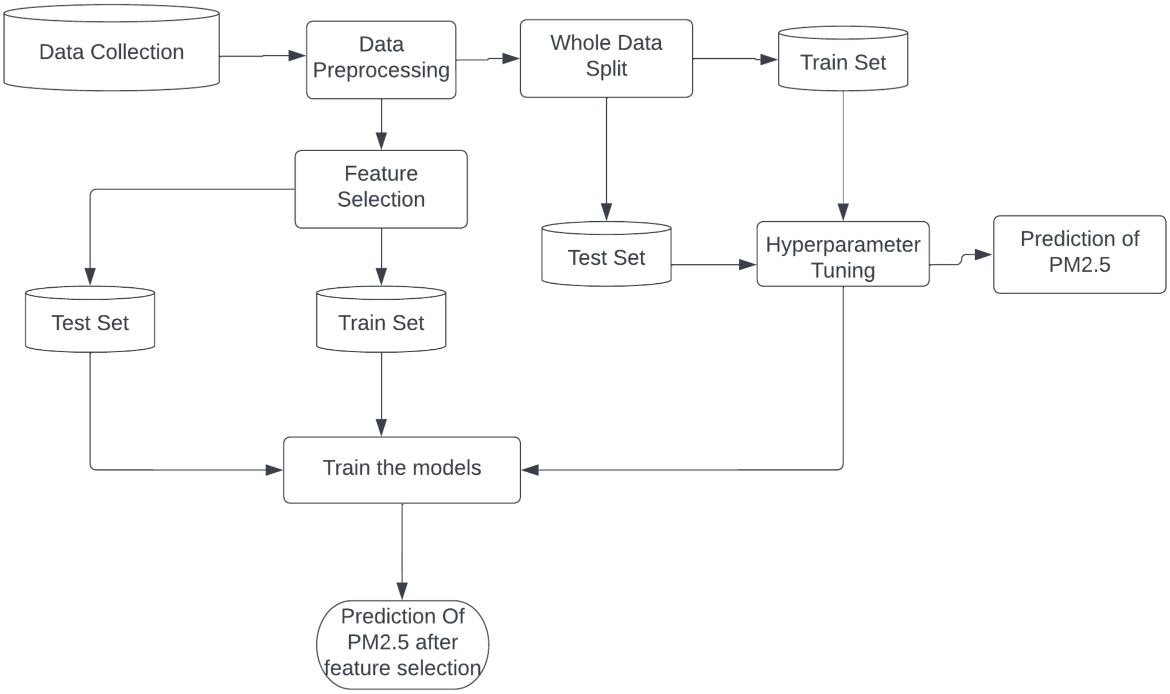

Fig 3.1: Workflow for predicting missing atmospheric PM2.5 from multi time series data

We have created new datasets from the existing dataset, by selecting a few specific stations, for a specific target station. We have done this with the help of some methods like Radius-Based, clustering, KL-divergence, Triangulation, Mutual-Information. which are discussed in detail in section 3.4. Again, we split the new datasets into train-test sets. In this case, we have taken 20% data into the test set. Then all the models are trained for each dataset with the best parameters, which we got by hyper tuning. Finally, we have calculated the accuracy for each model with predicted values and actual values. To find the accuracy we have used different kinds of performance matrices like MAE, RMSE, R2 score, and correlation.

3.3 Data Collection

For our work we have considered PM2.5 pollutant. We have taken data from twenty-eight stations situated in the congested area of India, Delhi. The data collected from the website of Central Pollution Control Board (https://www.cpcb.nic.in). We took data of a year for everyday in one-hour intervals. Our data had no null/garbage/missing values. The total no of data points in our dataset is 9504. Also, we have collected latitude and longitude of each station.

3.4 Proposed Approach

For our work, we have chosen two ways. The first one is finding the best model and the second one is feature selection. For each model, we have found the best set of parameters with the help of GridSearchCv. Also, for some models we have checked by manually since the GridSearchCv function took a lot of time. In that case first we run models with their default set of parameters, then changed the value of one parameter by keeping the rest fixed e.g., for Random Forest model we have varied the 'n_estimator' parameter from 5 to 1000. For feature selection we have taken five different methos.

i) **Radius Based:** In this method we taken Shadipur as our target station. In this approach, to select closest station of Shadipur we have taken a circle of certain radius. We have changed the value of radius from 5 units to 25 units in an interval of 5 units. So, there are fewer stations within the radius of 5 units and more stations within the radius of 25 units.

ii) **Triangulation:** To properly describe the border of features, the Feature Point Triangulation (FPT) approach is used in connection with spatial subdivision utilising a quadtree data structure. In this approach, the points in the quadtree data structure that are closest to the centre of the quadrants are found and utilised to build four triangles per quadrant. quadrants (polygons), which are then utilised to make triangular lists. In this work we have considered six triangles.

iii) **Clustering:** In this approach we have divided our dataset into different groups in basis of different cluster. Basically, here we have constructed a dataset by grouping all those stations who are in the same cluster of Shadipur. For clustering we have determined the latitude and longitude of each city Here cluster of each station are defined by the k-means clustering Algorithm. The value of 'k' centroid was defined from2 to 20 in an interval of 2.

iv) **Mutual Information**: To selecting appropriate training data we have taken Mutual Information (MI) approach. MI is model neutral. MI measures the entropy drops under the condition of target value. MI (feature; target) = Entropy(feature)-Entropy (feature\target). MI score will fall in the range from 0 to infinity. Higher value indicates the closer connection between this features and target. In this work, we get 0.53 as best MI score.

v) **Kullback-Leibler Divergence:** The Kullback-Leibler divergence (hereafter written as KL divergence) is a measure of how a probability distribution differs from another probability distribution. Classically, in Bayesian theory, there is some true distribution P(X); we'd like to estimate with an approximate distribution Q(X). In this context, the KL divergence measures the distance from the approximate distribution Q to the true distribution P. KL-divergence is not symmetric. Actually, it measures the amount of information loss, when q(x) makes a approximation of p(x). Less value indicates, it is a better approximation. The KL-divergence from Q to P on some space X can be written as

$$D_{KL}(P|| \mathrm{Q}) = \sum_{x \in X} P(x) \ln (\frac{P(x)}{Q(x)})$$

Chapter 4

Result and Analysis

In this section first we noted down the libraries, tools, that we have used in this work. Then a detailed analysis of the results is given.

4.1 Experimental Setup

We have implemented our all models and methods using python programming language. We have used the Scikit-learn (https://scikit-learn.org) library to implement machine learning models. Also, we have used matplotlib library (https://matplotlib.org) for plotting graphs and charts, pandas (https://pandas.pydata.org) library for data manipulation that is for import and analyze dataset, numpy (https://numpy.org) library to handle arrays. To calculate training and testing time we have used time library. We have used Microsoft office excel to store the datapoints and results.

4.2 Results & Discussion

In the study, for comparison of models, methodologies we have used mean of absolute error (MAE), root mean of squared error(RMSE),R2 Score and correlation between them. MAE is calculated as the average of absolute error of all sample of test set. Root mean squared error (RMSE) is the square root of the mean of the square of all of the error.

$$\text{MAE} = (1/n) * \sum|P_i - O_i|$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(P_i - O_i)^2}$$

Where, Σ is a fancy symbol that means "sum", P_i is the predicted value for the i^{th} observation in the dataset O_i is the observed value for the i^{th} observation in the dataset, n is the sample size. R-Squared (R2) is a metric for assessing the performance of regression machine learning models. Unlike other metrics such as MAE or RMSE it is not a measure of how accurate the predictions are, but instead a measure of fit. The R2 score gives an indication as to how much of the variation is explained by the independent variables in the model. The R2 score ranges from 1, a perfect score, to negative values for under-performing models. An important reminder when looking at the R2 scores from different models is that the variance found in a dataset is not comparable across datasets, meaning that R2 scores cannot be used to directly compare model performance.

$$R^2=1-(RSS)/(TSS),$$

R^2 is coefficient of determination, RSS is sum of squares of residuals, TSS is total sum of squares. All the models used in this work have been compared with the value of MAE, RMSE, R2 Score.

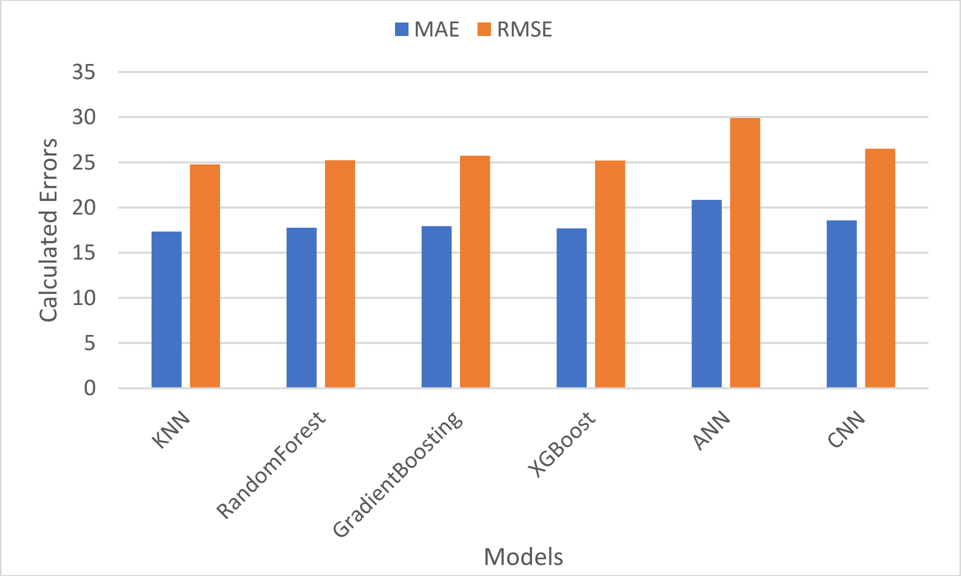

Fig 4.1: Calculated error against different models.

From fig-4.1, we have represented the performance of all models with their best set of parameters. It turns out that the KNN model has given the best result with MAE 17.33 and RMSE 24.79. All off these performances are calculated after finding the best set of parameters. We have done this before feature selection, that is, predicting over the whole dataset. For this kind of dataset deep learning models are failed to give the best result. In this work we have taken two deep learning models, Artificial neural network (ANN) and Convolutional neural network(CNN). In this case, the size of the test set was 20% as of the dataset. Then we also look at how much time a model is taking for predicting, for different sizes of test set.

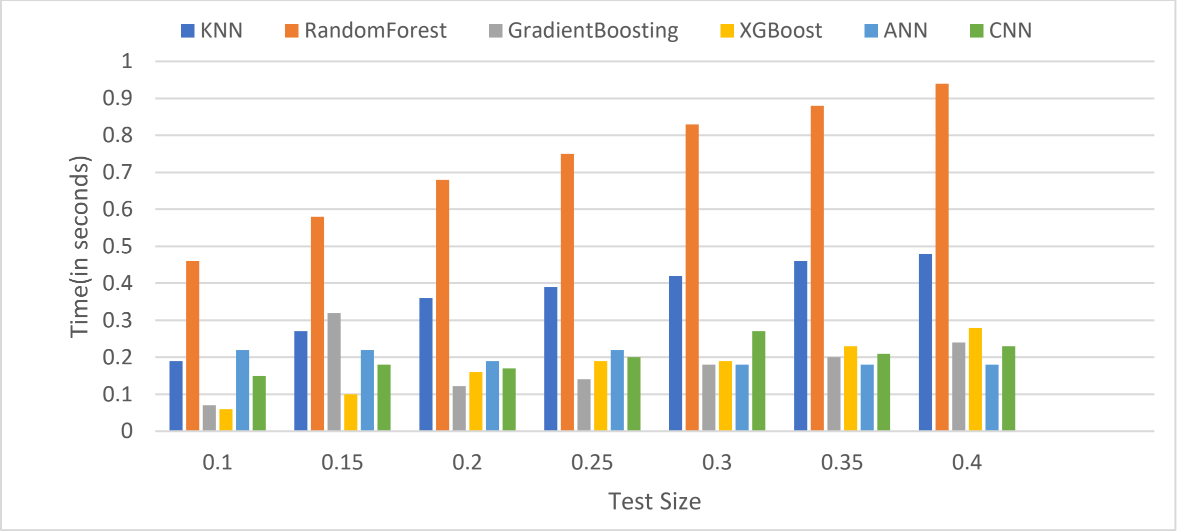

Fig 4.2: Time taken by different model for various test set size.

Fig 4.2 shows that models are taking different time for different test sizes. For example, XGBoost took the least time for 0.1 test size, ANN took the least time for 0.4 test size. That is, they did not follow any specific order. However, for all types of test size, the Random Forest Model has taken a long time. Since its best set of parameters has 1000 'n_estimators'. How long the deep learning models take depends on their layer, batch size, epochs. We also saw that the same model was taking different time for the same test size. We analyze the radius-based method with the help of graphs in fig 4.3. We have plotted the MAE value of all the models on the basis of radius-based approach. We can see that performance of the models has improved as the radius has increased. However, looking at the graph, it seems that the performance of the model remains the same after a certain radius. That means, they converge in a certain value. So, it can be said that this approach will be helpful if working with any large datasets.

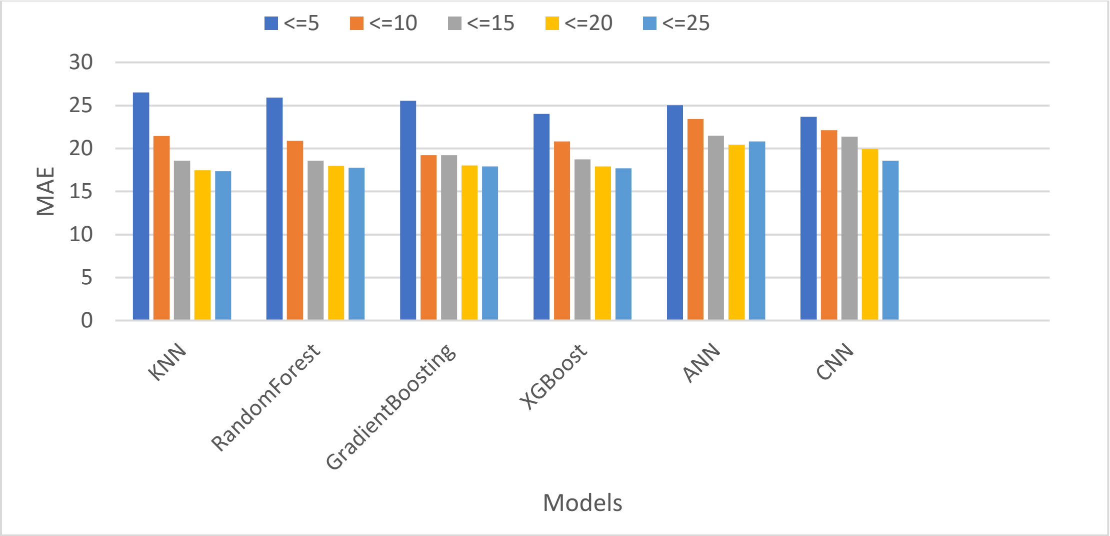

Fig 4.3 Predicting MAE for all models for different radius

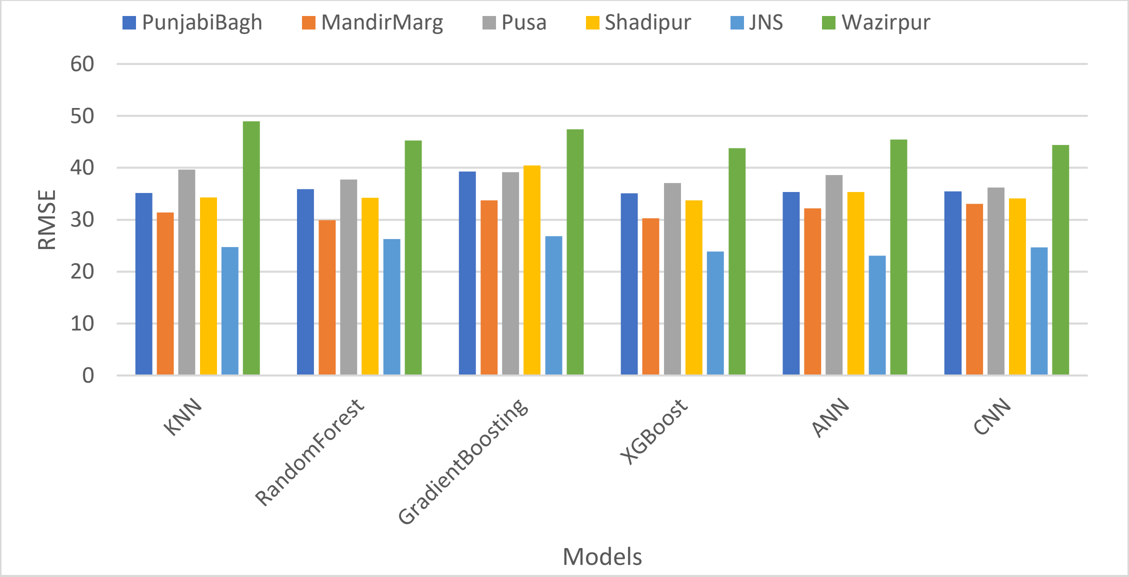

Fig 4.4 Predicting RMSE for all models for different Triangulation

In fig 4.4 we have analyzed the triangulation methodology. We have applied the triangulation method on six cities. Unlike other methods, the datasets that have been created in this process all have the same number of stations. Here for each model compared the RMSE value. The graph shows that JNS triangulation is giving best result with value of MAE = 11.46 and RMSE = 24.75 (considering KNN model). That is, the triangulation method selected the appropriate feature for JNS triangle. However, for other methods we have selected the city Shadipur as the target station. From this we can say that this method is very useful for some specific station.

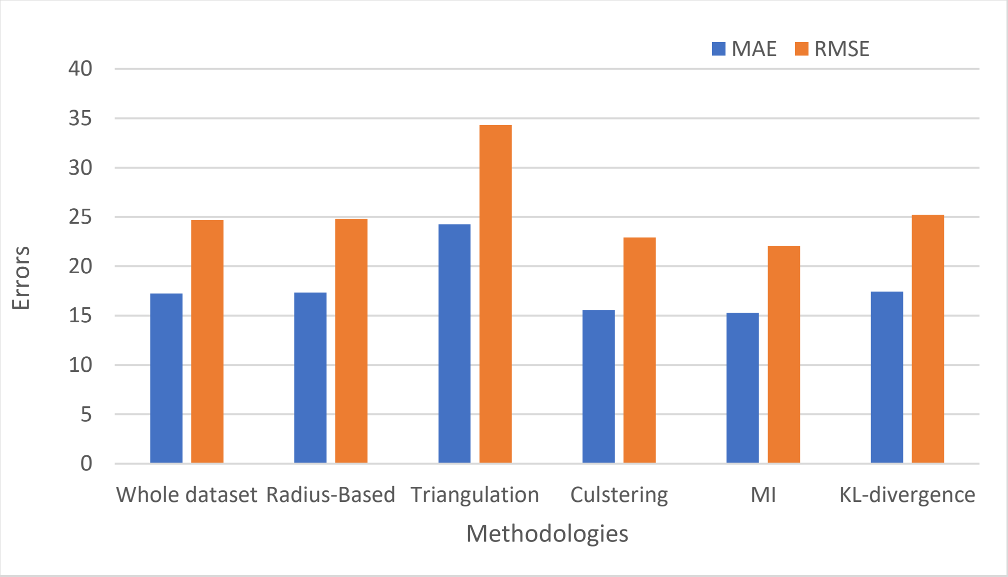

Fig 4.5: MAE and RMSE comparison for all methodologies of KNN model

In fig 4.5 indicates the comparison of MAE and RMSE of the proposed approaches for KNN model. From fig 4.5 can observe that, the difference in the value of accuracy before and after the feature selection is negligible. But we have shown the advantages of radius-based and triangulation methods in Fig-4.3 and Fig-4.4. From Fig-4.5 we can see that the clustering-based method has performed relatively well. It gave good results when we broke our dataset into two clusters. Since our target station is Shadipur, a dataset has been created with the stations in the cluster of Shadipur city. We have implemented our models with this dataset. By splitting the dataset in two, the number of stations on each dataset has halved. As a result, it took less time to execute and at the same time gave good results. While the MI and KL-divergence accuracy are not much of a change, all of these methods will work when time is taken as an indicator of good results. So, we can tell from this work that the con-model is one of the leading models for missing value prediction for PM2.5 concentration. And the methods that have been used in this work are really useful.

Chapter 5

Conclusion and Future work

This chapter gives a brief overview of the whole work. It also talks about the future application of this work and how it can be improved.

5.1 Conclusion

Machine models are widely used for concentration prediction of PM2.5. In this work we have proposed an ensemble learning based model K-nearest neighbor (KNN) regression. We have compared this model with Random Forest, Gradient Boosting, Extreme Gradient Boosting, and two deep learning model (Artificial Neural Network, Convolutional Neural Network). MAE, RMSE, R2-Score are used to compare them. Deep learning models are also giving good result, but they have failed for our dataset. We have worked with the best set of parameters for each model. While doing this work we have seen that it is very important to choose the right methodology for feature selection.

5.2 Future Work

This model can be used to predict the concentration of other particle in the air. In future study other machine learning models can be explore with this dataset. The methodology we have used for feature selection in this work can be used to work with a large dataset. Besides, the work of predicting concentration of PM2.5 is really helpful for human health.

References

1. C. Lytridis, A. Lekova, C. Bazinas, M. Manios, and V.G. Kaburlasos, “WINkNN: Windowed Intervals’ Number kNN Classifier for Efficient Time-Series Applications,” Mathematics, vol. 8, no. 3, 2020.

2. Ho, T.K. Random decision forests. In Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Montreal, QC, Canada, 14–15 August 1995; pp. 278–282.

3. Chen, T.; Guestrin, C. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD ’16, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794

4. Hamed Karimian1,2, Qi Li 2, Chunlin Wu2, Yanlin Qi2, Yuqin Mo2, Gong Chen2,4, Xianfeng Zhang2, Sonali Sachdeva3. Evaluation of Different Machine Learning Approaches to Predicting PM2.5 Mass Concentrations, Volume 19, Issue 6, June 2019

5. J.B. Ordieresa, *, E.P. Vergaraa , R.S. Capuzb , R.E. Salazarc, Neural network prediction model for fine particulate matter (PM2.5) on the USeMexico border in El Paso (Texas) and Ciudad Jua´rez (Chihuahua), Received 5 June 2003; received in revised form 4 October 2003; accepted 8 March 2004.

6. J. Zhao, X. Mao and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks", *Biomed. Signal Process. Control*, vol. 47, pp. 312-323, 2019.

7. O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj and D. J. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks", *J. Sound Vib.*, vol. 388, pp. 154-170, Feb. 2017.

8. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

9. H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures dataset characteristics and transfer learning", *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285-1298, May 2016.

10. A. Sagheer and M. Kotb, “Time series forecasting of petroleum production using deep LSTM recurrent networks,”, Neurocomputing, vol. 323, pp. 203-213, 2019.

11. R.O. Sinnott, Z. Guan, Prediction of air pollution through machine learning approaches on the cloud, in: 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT), IEEE, 2018, pp. 51–60.

12. M. Zamani Joharestani, C. Cao, X. Ni, B. Bashir, S. Talebiesfandarani, PM2. 5 prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data, Atmosphere 10 (7) (2019) 373.

13. Shamsoddini, A.; Aboodi, M.R.; Karami, J. Tehran air pollutants prediction based on Random Forest feature selection method. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. ISPRS Arch. 2017, 42, 483–488. [CrossRef]

14. Nabavi, S.O.; Haimberger, L.; Abbasi, E. Assessing PM2.5 concentrations in Tehran, Iran, from space using MAIAC, deep blue, and dark target AOD and machine learning algorithms. Atmos. Pollut. Res. 2019, 10, 889–903. [CrossRef]

15. TAOYING LI , MIAO HUA, AND XU WU, A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5), February 12, 2020.

16. Fuxin Jiangab, Chengyuan Zhangc, ShaolongSund, JingyunSune, Forecasting hourly PM2.5 based on deep temporal convolutional neural network and decomposition method, Volume 113, Part B, December 2021, 107988

17. Saurabh Kumar, Shweta Mishra, Sunil Kumar Singh , A machine learning-based model to estimate PM2.5 concentration levels in Delhi's atmosphere, 24 November 2020.