TRANSFORMING AIR QUALITY MONITORING WITH DEEP LEARNING: A PREDICTIVE APPROACH VIA IOT SENSORS

Akavaram Swapna[1],
Sri Priya Nagula Malyala[1],
Chigurlapalli Swathi[1]

TRANSFORMING AIR QUALITY MONITORING WITH DEEP LEARNING: A PREDICTIVE APPROACH VIA IOT SENSORS

Akavaram Swapna[1], Sri Priya Nagula Malyala[1],Chigurlapalli Swathi[1]
[1]Department of Computer Science Engineering,
[1]Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana

ABSTRACT

The ability to forecast and analyze air quality has significantly improved over time. We previously placed a great deal of reliance on conventional techniques like statistical models and simple equations. These methods, however, found it difficult to convey the dynamic and intricate character of air pollution. In an effort to enhance air quality forecasts, scientists and researchers have been using artificial intelligence (AI), machine learning, and big data analytics. Conversely, air pollution is a major worldwide problem that impacts not only our surroundings but also our health and welfare. Additionally, it is connected to cardiovascular and respiratory conditions, which raises the number of illnesses and fatalities. Precise forecasts of air quality enable public authorities, governments, and people to respond promptly in order to mitigate pollution, protect public health, and enhance urban development. We require precise study and forecast of air quality in order to address this urgent issue. The shortcomings of conventional approaches for predicting air quality are what drove us to create this AI model. As we've shown, these approaches frequently lack precision and have trouble taking into account the many variables driving air pollution. With its capacity to analyze enormous volumes of real-time data and spot intricate patterns, artificial intelligence (AI) presents a promising way to improve the precision and dependability of air quality forecasts. As a result, this work presents a cutting-edge Artificial Intelligence (AI) model that is intended to accurately and efficiently forecast and assess air quality. This model attempts to satisfy the increasing need for trustworthy real-time air quality information by combining state-of-the-art AI algorithms and data analytics methodologies.

Keywords: Artificial Intelligence, Air Quality, Machine Learning, Big Data Analytics, Respiratory, Cardiovascular

1. INTRODUCTION

Energy consumption and its consequences are inevitable in modern age human activities. The anthropogenic sources of air pollution include emissions from industrial plants; automobiles; planes; burning of straw, coal, and kerosene; aerosol cans, etc. Various dangerous pollutants like CO, CO_2, Particulate Matter (PM), NO_2, SO_2, O_3, NH_3, Pb, etc. are being released into our environment every day. Chemicals and particles constituting air pollution affect the health of humans, animals, and even plants. Air pollution can cause a multitude of serious diseases in humans, from bronchitis to heart disease, from pneumonia to lung cancer, etc. Poor air conditions lead to other contemporary environmental issues like global warming, acid rain, reduced visibility, smog, aerosol formation, climate change, and premature deaths. Scientists have realized that air pollution bears the potential to affect historical monuments adversely [1]. Vehicle emissions, atmospheric releases of power plants and factories, agriculture exhausts, etc. are responsible for increased greenhouse gases. The greenhouse gases adversely affect climate conditions and consequently, the growth of plants [2]. Emissions of inorganic carbons and greenhouse gases also affect plant-soil interactions [3]. Climatic fluctuations not only affect humans and

animals, but agricultural factors and productivity are also greatly influenced[4]. Economic losses are the allied consequences too.

The Air Quality Index (AQI), an assessment parameter is related to public health directly. higher level of AQI indicates more dangerous exposure for the human population. Therefore, the urge to predict the AQI in advance motivated the scientists to monitor and model air quality. Monitoring and predicting AQI, especially in urban areas has become a vital and challenging task with increasing motor and industrial developments. Mostly, the air quality-based studies and research works target the developing countries, although the concentration of the deadliest pollutant like PM2.5 is found to be in multiple folds in developing countries [5]. A few researchers endeavoured to undertake the study of air quality prediction for Indian cities. After going through the available literature, a strong need had been felt to fill this gap by attempting analysis and prediction of AQI for India.

2. LITERATURE SURVEY

In [6],Gopalakrishnan (2021) combined Google's Street view data and ML to predict air quality at different places in Oakland city, California. He targeted the places where the data were unavailable. The author developed a web application to predict air quality for any location in the city neighborhood. Sanjeev [7] studied a dataset that included the concentration of pollutants and meteorological factors. The author analyzed and predicted the air quality and claimed that the Random Forest (RF) classifier performed the best as it is less prone to over-fitting. Castelli et al. [8]endeavoured to forecast air quality in California in terms of pollutants and particulate levels through the Support Vector Regression (SVR) ML algorithm. The authors claimed to develop a novel method to model hourly atmospheric pollution. Doreswamy et al. [9] investigated ML predictive models for forecasting PM concentration in the air. The authors studied six years of air quality monitoring data in Taiwan and applied existing models. They claimed that predicted values and actual values were very close to each other. In [10], Liang et al. studied the performances of six ML classifiers to predict the AQI of Taiwan based on 11 years of data. The authors reported that Adaptive Boosting (AdaBoost) and Stacking Ensemble are most suitable for air quality prediction, but the forecasting performance varies over different geographical regions.

3. PROPOSED METHODOLOGY

3.1 Overview

Air quality prediction using IoT sensor data is a critical application that leverages technology to monitor, assess, and forecast air quality conditions in various environments. This process involves collecting real-time data from a network of IoT sensors deployed in different locations, analyzing this data, and using it to make predictions about air quality. In the process of collecting and managing data from IoT sensors, the information gathered is carefully stored within a centralized database or cloud-based platform. This data is marked with timestamps, providing details about the location of the sensors, the specific type of sensors used, and the actual measurements recorded. This meticulous record-keeping ensures that we have a comprehensive dataset to work with.Prior to delving into data analysis, there is a crucial step known as data preprocessing. During this phase, the data undergoes a series of operations aimed at refining it for further analysis. These operations include addressing missing data points, handling outliers, and, if necessary, converting data into standardized formats. This step ensures that the data is in its best possible condition for accurate analysis.Once the data is pre-processed, the next step involves feature engineering. Here, we extract and create relevant features from the raw sensor data.
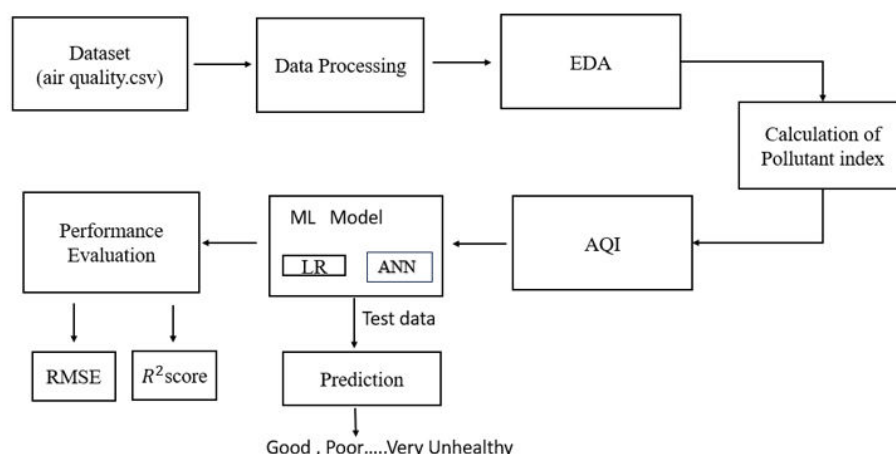

Figure 1: Overall design of proposed air quality prediction.

3.2ANNClassifier

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images. Interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt's model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.

$$y = \begin{cases} 1, \text{if } \overbrace{\sum_i w_i x_i}^{\text{weighted sum}} - T > 0 \\ 0, \text{otherwise} \end{cases}$$
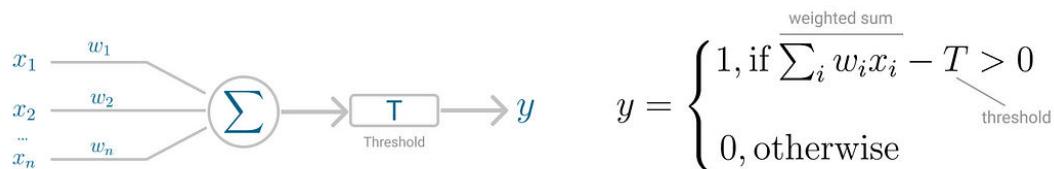
Fig. 2: Perceptron neuron model (left) and threshold logic (right).

Threshold T represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

With this discrete output, controlled by the activation function, the perceptron can be used as a binary classification model, defining a linear decision boundary.

It finds the separating hyperplane that minimizes the distance between misclassified points and the decision boundary. The perceptron loss function is defined as below:

$$\underbrace{D(w, c)}_{\text{distance}} = - \sum_{i \in \text{M}} \overset{\text{output}}{y_i} (x_i w_i + c)$$

misclassified observations

To minimize this distance, perceptron uses stochastic gradient descent (SGD) as the optimization function. If the data is linearly separable, it is guaranteed that SGD will converge in a finite number of steps. The last piece that Perceptron needs is the activation function, the function that determines if the neuron will fire or not. Initial Perceptron models used sigmoid function, and just by looking at its shape, it makes a lot of sense! The sigmoid function maps any real input to a value that is either 0 or 1 and encodes a non-linear function. The neuron can receive negative numbers as input, and it will still be able to produce an output that is either 0 or 1.

But, if you look at Deep Learning papers and algorithms from the last decade, you'll see the most of them use the Rectified Linear Unit (ReLU) as the neuron's activation function. The reason why ReLU became more adopted is that it allows better optimization using SGD, more efficient computation and is scale-invariant, meaning, its characteristics are not affected by the scale of the input. The neuron receives inputs and picks an initial set of weights random. These are combined in weighted sum and then ReLU, the activation function, determines the value of the output.
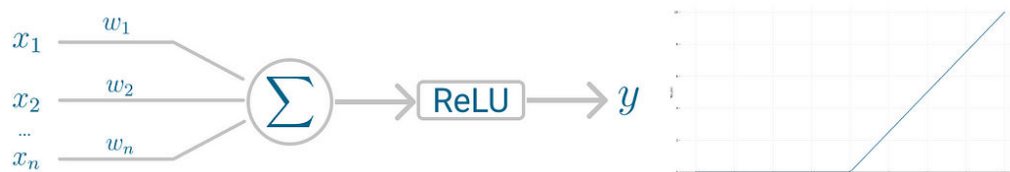

Fig. 3: Perceptron neuron model (left) and activation function (right).

Perceptron uses SGD to find, or you might say learn, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

3.2.2 ANN

The ANN was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A ANN has input and output layers, and one or more hidden layers with many neurons

stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a ANN can use any arbitrary activation function. ANN falls under the category of feedforward algorithms because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer. If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.
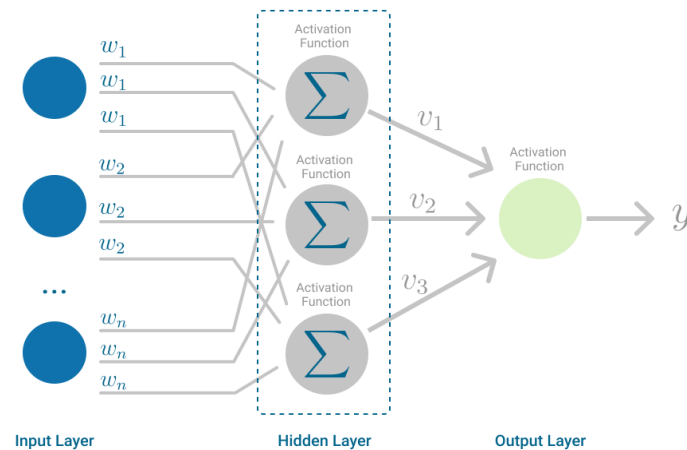

Fig. 4: Architecture of ANN.

Backpropagation: Backpropagation is the learning mechanism that allows the ANN to iteratively adjust the weights in the network, with the goal of minimizing the cost function. There is one hard requirement for backpropagation to work properly. The function that combines inputs and weights in a neuron, for instance the weighted sum, and the threshold function, for instance ReLU, must be differentiable. These functions must have a bounded derivative because Gradient Descent is typically the optimization function used in ANN. In each iteration, after the weighted sums are forwarded through all layers, the gradient of the Mean Squared Error is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. That's how the weights are propagated back to the starting point of the neural network. One iteration of Gradient Descent is defined as follows:

$$\Delta_w(t) = -\varepsilon \frac{dE}{dw_{(t)}} + \alpha \Delta_{w(t-1)}$$

This process keeps going until gradient for each input-output pair has converged, meaning the newly computed gradient hasn't changed more than a specified convergence threshold, compared to the previous iteration.

4. RESULTS AND DISCUSSION

Figure 5 shows the representation of a dataset containing air quality measurements, where ach row corresponds to a specific measurement taken at a particular point in time and location. The columns likely include various attributes discussed in dataset description.

Air Quality Prediction

SOI: [   ] NOI: [   ] RPI: [   ] SPMI: [   ] [Predict]

Figure 5: Sample HTML webpage using Flask server

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	location_monitoring_station	pm2_5	date
0	150	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN		NaN	NaN	1990-02-01
1	151	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN		NaN	NaN	1990-02-01
2	152	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN		NaN	NaN	1990-02-01
3	150	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN		NaN	NaN	1990-03-01
4	151	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN		NaN	NaN	1990-03-01

Figure 6: Sample dataset of air quality measurements taken at a particular location and time.

Figure 6 displays a list of features that are considered important for training ML model to classify air quality. The features are variables or attributes that will be used by the ML model to make predictions or classification. These features are selected based on their potential impact on air quality and their relevance to the prediction task.

	state	location	type	so2	no2	rspm	spm	pm2_5
0	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	4.8	17.4	0.0	0.0	0.0
1	Andhra Pradesh	Hyderabad	Industrial Area	3.1	7.0	0.0	0.0	0.0
2	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.2	28.5	0.0	0.0	0.0
3	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.3	14.7	0.0	0.0	0.0
4	Andhra Pradesh	Hyderabad	Industrial Area	4.7	7.5	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
435737	West Bengal	ULUBERIA	RIRUO	22.0	50.0	143.0	0.0	0.0
435738	West Bengal	ULUBERIA	RIRUO	20.0	46.0	171.0	0.0	0.0
435739	andaman-and-nicobar-islands	Guwahati	Residential, Rural and other Areas	0.0	0.0	0.0	0.0	0.0
435740	Lakshadweep	Guwahati	Residential, Rural and other Areas	0.0	0.0	0.0	0.0	0.0
435741	Tripura	Guwahati	Residential, Rural and other Areas	0.0	0.0	0.0	0.0	0.0

435742 rows × 8 columns

Figure 7: Important features for the proposed ML model.

Figure 7 showing the header information for an index that quantifies the pollution levels of individual pollutants, specifically sulphur dioxide (SO2) and nitrogen dioxide (NO2). Figure 4 displays the header information for an overall Air Quality Index (AQI) calculated using the data values from the Figure 3. AQI is a composite index that provides a simplified way to understand air quality by condensing multiple pollutants into a single value. The header includes details about the AQI scale, for different air quality levels, and the categories used to classify air quality (e.g., good, moderate, unhealthy, etc.).

	so2	SOi			no2	Noi
0	4.8	6.000		0	17.4	21.750
1	3.1	3.875		1	7.0	8.750
2	6.2	7.750		2	28.5	35.625
3	6.3	7.875		3	14.7	18.375
4	4.7	5.875		4	7.5	9.375

Figure 8: Header of individual pollutant index for SO2 and NO2.

Figure 8 is a visualization for the classification of air quality based on the calculated AQI values. The classification involves different categories such as "good," "moderate," "poor," "unhealthy," "very unhealthy," and "hazardous." These categories indicate the level of pollution and associated health risks.

	state	SOi	Noi	Rpi	SPMi	AQI
0	Andhra Pradesh	6.000	21.750	0.0	0.0	21.750
1	Andhra Pradesh	3.875	8.750	0.0	0.0	8.750
2	Andhra Pradesh	7.750	35.625	0.0	0.0	35.625
3	Andhra Pradesh	7.875	18.375	0.0	0.0	18.375
4	Andhra Pradesh	5.875	9.375	0.0	0.0	9.375

Figure 9: Header of Air Quality Index calculated from every data value.

	state	location	type	so2	no2	rspm	spm	pm2_5	SOi	Noi	Rpi	SPMi	AQI	AQI_Range
0	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	4.8	17.4	0.0	0.0	0.0	6.000	21.750	0.0	0.0	21.750	Good
1	Andhra Pradesh	Hyderabad	Industrial Area	3.1	7.0	0.0	0.0	0.0	3.875	8.750	0.0	0.0	8.750	Good
2	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.2	28.5	0.0	0.0	0.0	7.750	35.625	0.0	0.0	35.625	Good
3	Andhra Pradesh	Hyderabad	Residential, Rural and other Areas	6.3	14.7	0.0	0.0	0.0	7.875	18.375	0.0	0.0	18.375	Good
4	Andhra Pradesh	Hyderabad	Industrial Area	4.7	7.5	0.0	0.0	0.0	5.875	9.375	0.0	0.0	9.375	Good

```
Good            219643
Poor             93272
Moderate         56571
Unhealthy        31733
Hazardous        18700
Very unhealthy   15823
```

Figure 10 shows that Obtainedclassification ofair quality as good, moderate, poor, unhealthy, very unhealthy, and Hazardous.

Table 1 provides a comparison of two different machine learning models used for air quality prediction based on two evaluation metrics: Root Mean Squared Error (RMSE) and R-squared (R^2) score.

RMSE (Root Mean Squared Error): The RMSE is a metric used to measure the average magnitude of the errors between predicted values and actual (observed) values. It quantifies how well the predictions align with the actual data. A lower RMSE value indicates better predictive performance, as it means the model's predictions are closer to the actual values. From Table 1:

— For the "LR" model, the RMSE is 13.67.
— For the "ANN Model" model, the RMSE is 0.

A lower RMSE for the ANN Model suggests that it has smaller prediction errors compared to the LR model.

R^2-score (Coefficient of Determination): The R^2 score is a statistical measure that represents the proportion of the variance in the dependent variable that's explained by the independent variables in a regression model. It ranges from 0 to 1, where higher values indicate that the model's predictions closely match the actual data. An R^2 score of 1 indicates a perfect fit. From Table 1:

— For the "LR" model, the R^2 score is 0.9847.
— For the "ANN" model, the R^2 score is 0.999999.

The R^2 scores for both models are quite high, indicating that they both provide excellent fits to the data. However, the ANN Model score of 0.999 suggests an almost perfect fit, meaning that it captures the variability in the data extremely well. Finally, the ANN Model outperforms the LR model in terms of both RMSE and R^2 score, indicating its superior predictive capability and ability to explain the variance in air quality data.

Table 1: Comparison ofmodels.

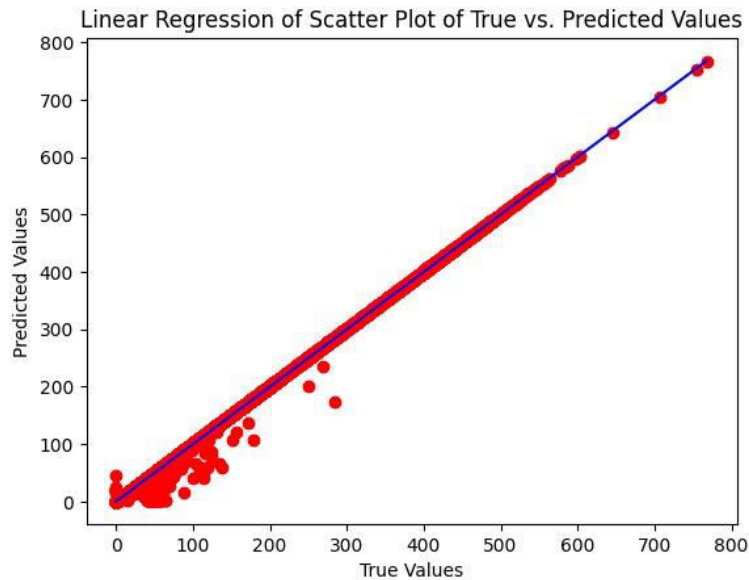| Model name	RMSE	R^2-score
LR	13.67	0.9847
ANN	1.16	0.999

Figure 11: Scatter plot of true and predicted values obtained using LR model.

Figure 11 is a scatter plot visualizes the performance of a LR model. In this plot, each point represents a data instance. The x-axis represents the true values (actual observations) of the target variable, while the y-axis represents the predicted values of the target variable made by the LR model. Each point on the plot corresponds to a data instance, where its position relative to the diagonal line (which represents a perfect prediction) indicates how well the model's predictions align with the actual data. If the points are close to the diagonal line, it suggests that the model's predictions are accurate. In Figure 12, the scatter plot illustrates the performance of aANN model. Each point on the plot represents a data instance, where the x-axis shows the true values of the target variable, and the y-axis shows the predicted values made by the ANN Model. The positioning of points relative to the diagonal line helps assess the accuracy of the model's predictions. From Figure 7, it indicates that the points cluster around the diagonal line are closely matches the actual values.
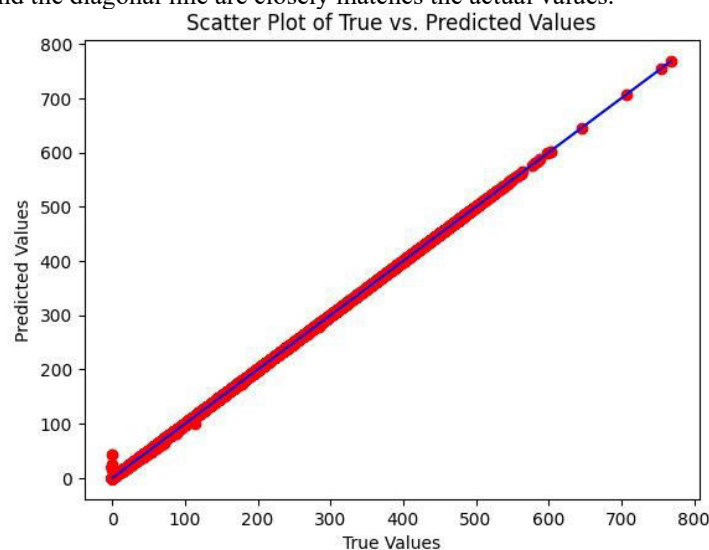
Figure 12: Scatter plot of true and predicted values obtained using ANNmodel.

Figure 13 shows the results of a sample prediction made by a trained ANN model on new or unseen test data. This sample prediction outcome helps to evaluate how well the model performs on unseen data and whether its predictions are accurate. By comparing the true and predicted values, the proposed system assessed whether the model generalizes well to new data points.

Prediction Result

Predicted Value: 12.199300000000001

Air Quality: Good

Go back to home

Figure 13: Sample prediction outcome with new test data.

5. CONCLUSION

In the realm of air quality prediction, the use of Artificial Neural Networks (ANN) has been pivotal in providing valuable insights and forecasts. ANN models demonstrate significant advantages over traditional Linear Regression (LR) models, particularly in handling complex relationships and mitigating overfitting. While LR models are straightforward and interpretable, they often struggle to capture the nuances of complex, non-linear interactions within air quality data. On the other hand, ANN models, especially when leveraging deep learning techniques, exhibit superior performance by automatically learning intricate relationships and interactions between various air quality parameters.The flexibility of ANN models allows them to adapt to diverse data patterns, making them well-suited for capturing the complex dynamics inherent in real-world air quality scenarios. Additionally, ANN models, particularly deep neural networks, can effectively handle large datasets with high dimensionality, which is often the case in air quality prediction tasks. Moreover, the ability of ANN models to generalize well to new data and their robustness against outliers further contribute to their superior performance.While ANN models have proven to be a formidable choice for air quality prediction, the field of air quality forecasting continues to evolve. Further exploration of feature engineering techniques, including the creation of novel features and the incorporation of additional environmental and meteorological data, can enhance the performance of ANN models. Additionally, the development of hybrid models that combine the strengths of ANN with other machine learning techniques, such as Random Forests or Gradient Boosting Machines, can lead to even more accurate and robust predictions.

REFERENCES

[1] Rogers CD (2019) Pollution's impact on historical monuments pollution's impact on historical monuments. SCIENCING.

[2] Fahad S, Sönmez O, Saud S, Wang D, Wu C, Adnan M, Turan, V (2021a) Plant growth regulators for climate-smart agriculture (1st ed.). CRC Press.

[3] Fahad, S, Sönmez O, Saud S, Wang D, Wu C, Adnan M, Turan V (2021b) Sustainable soil and land management and climate change (1st ed.). CRC Press.

[4] Sönmez O, Saud S, Wang D, Wu C, Adnan M, Turan V (2021) Climate change and plants: biodiversity, growth and interactions (S. Fahad, Ed.) (1st ed.). CRC Press.

[5] Rybarczyk Y, Zalakeviciute R (2021) Assessing the COVID-19 impact on air quality: a machine learning approach. Geophys Res Lett

[6] Gopalakrishnan V (2021) Hyperlocal air quality prediction using machine learning. Towards data science. https://towardsdatascie nce.com/hyperlocal-air-quality-prediction-using-machine-learn ing-ed3a661b9a71.

[7] Sanjeev D (2021) Implementation of machine learning algorithms for analysis and prediction of air quality. Int. J. Eng. Res. Technol. 10(3):533–538.

[8] Castelli M, Clemente FM, Popovič A, Silva S, Vanneschi L (2020) A machine learning approach to predict air quality in California. Complexity 2020(8049504):1–23. https://doi.org/10.1155/2020/8049504.

[9] Doreswamy HKS, Yogesh KM, Gad I (2020) Forecasting Air pollution particulate matter (PM2.5) using machine learning regression models. Procedia Comput Sci 171:2057–2066.

[10] Liang Y, Maimury Y, Chen AH, Josue RCJ (2020) Machine learning-based prediction of air quality. Appl Sci 10(9151):1–17. https:// doi.org/10.3390/app10249151