GRU-LSTM with Attention-based Forecasting for Enhanced Air Quality

Sudhir Kumar, Vaneet Kour, Praveen Kumar, Anurag Deshmane, Rajiv Misra, *Senior Member, IEEE,*

***Abstract*—Air pollution, which is responsible for many chronic diseases and early deaths, has recently caused human health concerns to spread worldwide. Poor air quality has detrimental consequences on human beings and crops and significant adverse political, sociological, and economic consequences. To offer relevant and useful solutions, attain acceptable air quality, and develop preventative strategies, and it is crucial to put more effort into precise forecasting of environmental air pollution. Our research proposes an adaptable and effective deep learning-driven model to predict ambient pollutant concentrations of PM2.5. This research introduces the use of an Attention mechanism coupled with the GRU, LSTM, and hybrid model of GRU+LSTM in the forecasting of air quality in order to build a predictive modeling technique based on deep learning architecture. The proposed forecasting model's accuracy is then assessed through experimental validation using a publicly available data set on air pollution. The proposed Attention_GRU+LSTM model performs better than existing Deep Learning Models, such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Attention_GRU, Attention_LSTM, and GRU+LSTM, and has successfully enhanced the air pollution forecasting ability, based on new findings.**

***Index Terms*—Deep learning, GRU, LSTM, PM2.5 concentration prediction.**

I. Introduction

REGARD for environmental preservation, well-being, and safety have been garnering a lot of attention globally as a result of the rising environmental risks to the world. In urban regions and industrialized nations, air pollution is becoming a serious issue and is one of the main causes of global warming. In developing nations, particularly in bigger urban centers with a high concentration of emission sources, such as vehicles and industrial activity, reducing air pollution is a critical concern. Numerous epidemiological studies demonstrated the impact of specific Chemical substances, including sulfur dioxide (SO2), nitrogen dioxide (NO2), ozone (O3), or dust particles, on the health of the general population, with sensitive individuals, such as asthmatics, young children, and the elderly, being particularly affected [15]. Today, air quality is a multidisciplinary issue that involves experts in epidemiology, transport modeling, pollution emissions and transformation, spatial systems, forecasts, local governments, and industrialists. In order to ensure appropriate air quality, monitoring the environment is crucial [4]. Much work has been done in recent years to improve air quality [13], [16]. For instance, practically all nations have constructed air quality networks (made up of several measuring stations) to track the concentrations of various air contaminants. In order to forecast the concentrations of carbon monoxide (CO), nitrogen monoxide (NO), nitrogen dioxide (NO2), sulphur dioxide (SO2), and ozone (O3), PM2.5 this study aims to develop an effective deep learning data-driven model. Reliable air pollution forecasting provides useful information that enables individuals to take the appropriate actions to avoid unfavorable outcomes. Additionally, it enables the implementation of more effective measures for safeguarding the public's health and preventing air pollution crises [5]. Over the past four decades, numerous model- and data-based techniques have been developed to enhance air pollution modeling and forecasting [6], [14], [15]. When predicting air pollution, traditional time-series models are frequently used in the literature [2], [8], [20]. These models include Holt-Winters models,seasonal-ARIMA, and autoregressive integrated moving averages(ARIMA) and their variants. These models include Holt-Winters models,seasonal-ARIMA, [31], [32] and autoregressive integrated moving averages(ARIMA) [9] and its variants. However, in cases where the concentration levels of pollutants exhibit erratic fluctuations, these approaches often result in notable forecasting errors. [36]. Additionally, the linear structure of statistical models, such as AR and ARIMA, prevents reliable forecasting of the nonstationary and nonlinear ambient air pollution [19]. The more adaptable machine learning models, such as neural network forecasting and support vector machines, have been frequently used to enhance air pollution forecasting in order to address the aforementioned limitation [7], [25], [28]. The complex interactions between process variables could be adequately modeled by machine learning models without the necessity for explicit model formulation to be given. Over the past two decades, numerous machine-learning techniques have been employed for the purpose of air pollution forecasting. A wavelet-based neural network approach, for instance, has been suggested in [25] for forecasting hourly, daily mean, and daily maximum concentrations of ozone (O3), sulphur dioxide (SO2), carbon monoxide (CO), nitrogen oxides (NO), nitrogen dioxide (NO2), and dust particles (PM2.5) one step in advance. In particular, this method divides each time series of each air pollutant into various time-scale components using the maximum overlap wavelet transform (MODWT). The Elman network is then used to analyze these time-scale elements. The wavelet network technique is utilized

Sudhir Kumar, Vaneet Kour, Praveen Kumar, Anurag Deshmane, Rajiv Misra are with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, India. (Email: sudhir_2221cs14@iitp.ac.in, vaneet_2221cs15@iitp.ac.in, praveen_2221cs11@iitp.ac.in, anurag_2001cs08@iitp.ac.in, and rajivm@iitp.ac.in)

Manuscript received April 19, 2021; revised August 16, 2021.

in this study for one-step-ahead forecasting. For improving multistep air pollutant concentration forecasting, [23] proposes a hybrid approach that combines the advantages of the multi-agent evolutionary genetic algorithm (MEGA), the empirical wavelet transform (EWT), and the nonlinear autoregressive network with exogenous inputs (NARX). The air pollutant series, in particular, is broken down using the EWT, and the MAEGA model then uses the optimised NARX neural network to anticipate air pollutant concentrations. Results indicated that when used to anticipate PM2.5, SO2, NO2, and CO, PM2.5 concentration levels in Beijing, China, our model outperformed more traditional methods. Nevertheless, real-time forecasting is not suitable for wavelet transform techniques due to their requirement for batch data. Researchers have merged an Artificial Neural Network (ANN) model with numerical models to improve the forecasting of daily air pollution levels (such as SO2, NO2, and PM10) based on meteorological parameters [16]0. Similar to this, Arhami et al. [7] created a hybrid forecasting model that combines Monte Carlo simulations (MCS) and artificial neural networks (ANN). Their model does uncertainty analysis in addition to hourly pollutant concentration predictions (particularly for NO2, NO, O3, CO, and PM10). To improve forecasting, this hybrid model makes use of a number of carefully chosen meteorological input factors. A promising technique for predicting air pollution has been demonstrated when ANN and MCS are combined. In both academia and industry, deep learning has lately become a promising study area for modeling and forecasting time series data [11], [12], [24], [29], [34]. The fundamental concept behind this forecasting technique involves the integration of three LSTM models, which collectively form a prediction model that incorporates data from adjacent industrial air quality monitoring stations and other pollution sources. This method of forecasting comprises several essential components, including the fusion of three LSTM models, the incorporation of information from nearby industrial air quality monitoring stations, and the assessment of external pollution sources. Furthermore, in the referenced study [30], the PM2.5 concentration levels were forecasted using the convolutional-based bidirectional gated recurrent unit (CBGRU) method.. This forecasting method outperforms SVR, gradient boosting regressor (GBR), LSTM, GRU, and bidirectional GRU in precisely forecasting PM2.5 concentration levels by combining three LSTM models, incorporating data from nearby industrial air quality monitoring stations, as well as external pollution sources. Reference [3] describes how to anticipate ambient air pollutant concentrations of PM2.5, PM10, NO2, CO, O3, and SO2 using LSTM optimized using a particle swarm optimization approach. In contrast, reference [8] offers a strategy for forecasting PM10 particles in various parts of the city of Skopje. In this method, RNN and LSTM models are combined to improve prediction accuracy. The results demonstrate that incorporating data from both air quality and weather significantly enhances the predictive accuracy of the combined LSTM, RNN, and LSTM models. Furthermore, the combination of RNN + LSTM consistently outperforms the ARIMA method. Utilizing RNN, LSTM, and GRU architectures, deep learning models are used to forecast air quality. Both meteorological time series and data on air quality are included in the AirNet dataset, which is used to test these models [10]. Through the analysis conducted in [10], it has been established that the GRU model exhibits superior performance in predicting PM10 concentrations compared to the RNN and LSTM architectures. The deep learning method suggested in references [33] and [21] for forecasting South Korean air pollution levels uses stacked autoencoder (SAE) architecture for data learning and training. therefore, in forecasting air pollution concentrations, reference [22] recommends an LSTM model-based method. The forecasting approach outlined in [21] is capable of automatically extracting crucial data, such as spatiotemporal correlations within air pollution concentrations. This characteristic enables the approach to recognize and make use of the pertinent relationships that are present in the data. Air quality spatiotemporal prediction has been carried out using STDL (spatiotemporal deep learning) architectures. To extract spatial-temporal relationships and anticipate air quality up to 48 hours in advance, reference [3] proposes a hybrid CNN and LSTM model. This method efficiently combines the benefits of CNN and LSTM to identify and make use of the spatiotemporal patterns in air pollution data. In their paper referenced as [33], Chiou-Jye and Ping-Huan offered a comparison of the CNN-LSTM model's performance with other, more traditional machine learning models in terms of their capacity to predict PM2.5. In their study, they presented an entirely novel deep learning model that combines a GRU+LSTM deep model with an attention mechanism. This novel method aims to improve the precision of PM2.5 concentration predictions in ambient air pollution. This paper makes three contributions.

1) The first contribution focuses on creating an attention deep learning architecture (Attention_GRU+LSTM) using the established GRU, LSTM, and attention mechanism.
2) The second contribution consists of three forecasting experiments (2 days, 5 days, and 10 days) that validate the suggested method.
3) The proposed forecasting model is compared to various other reliable recurrent neural network models that were employed in the investigation in the study's concluding section. This comparison aims to demonstrate and assess the effectiveness and predictability of each model in terms of ambient air pollution. The results indicate that the Attention_GRU+LSTM technique developed in this study outperformed existing deep learning models such as GRU, Attention_GRU, LSTM, Attention_LSTM, and GRU+LSTM. Furthermore, it provided satisfactory predictive performance for various types of air contaminants in the ambient environment.

The preparatory information required for this investigation is included in the following part, along with a brief overview of the Attention _GRU+LSTM forecasting approach. Using a publicly available dataset of air pollution data, Section 2 illustrates how well the strategies under consideration work. Describe METHODOLOGY and PERFORMANCE ANALYSIS in Sections 3 and 4 respectively. The setup and Results of

Section 5's use of different deep learning models are shown. Section 6 concludes this analysis and identifies prospective directions for future research.

II. Architecture of PM 2.5 Air quality Prediction

In this section, We briefly explain the PM2.5 prediction system using a deep learning model. According to the figure1, 4 components comprise the entire system. In the first component, we installed DHT11 sensors to measure temperature and humidity, CO2 sensors to measure atmospheric CO2, and other sensors to measure air quality. Component 2 in the system architecture presented in figure 1 uses edge devices like a Raspberry Pi and an Arduino Uno to gather data from Component 1. In order to collect precise environmental data, these edge devices are carefully positioned in prime areas. Transmission Control Protocol (TCP) and MQ Telemetry Transport (MQTT) are two of the protocols used to send the data that has been gathered to Component 3. The sharing of telemetry data in low-bandwidth environments is made possible by the lightweight open messaging protocol MQTT. For network clients with constrained resources, it is very helpful. This protocol makes it simple to transfer data over a network between application programs and computing equipment. A communication standard called Transmission Control Protocol (TCP) makes it possible for application programs and computing equipment to exchange messages with one another via a network. It offers data packet distribution that is dependable and well-organized. In conclusion, Component 2 makes use of edge devices to gather data from Component 1, and the data is then sent to Component 3 utilizing protocols like TCP and MQTT, where MQTT is specially made for effective communication in low-bandwidth conditions. It's made to ensure that data and messages are successfully sent through networks and to send packets across the internet. The 5G and 6G telecom service providers were described in Component 3 as connecting to the cloud with high bandwidth and low latency, which allowed us to increase the system's efficiency. The system's fourth and most crucial component comes last. It's responsible for training and testing the model with research data sets. Figure2 depicts the flow of the PM 2.5 prediction System using a deep learning model. In the first step, we collect the data from the sensors from the real environment using IoT Sensors; after that, we pre-process the data with some techniques like missing values, encoding categorical variables, and then normalizing the data using the max-min algorithm. The total dataset was divided into two parts: 20% for model testing and 80% for model training. This division makes sure that the model is trained on the majority of the data, which enables it to recognize patterns and make predictions. The model's effectiveness and capacity to generalize to new data during testing are next evaluated using the remaining fraction of the data. This method aids in evaluating the model's efficiency and projecting how well it will function with hypothetical data. This paper uses GRU, LSTM, GRU-LSTM, Attention _GRU, Attention _LSTM, and Attention _GRU+LSTM to predict the PM 2.5 with the available train data set and validate the out with the testing data set. In the final step, we compare all the models with respect to the performance metric (RMSE, R squared, MSE, and MAE).

III. METHODOLOGY

This section briefly explains the fundamental ideas behind Attention Mechanism, GRU, and LSTM. The proposed Attention_GRU+LSTM deep learning-driven technique is then shown in fig 3. Figure 2 presents the general schematic of the suggested forecasting approach.

A. Attention Mechanism

The basic goal of the attention mechanism is to mimic human behavior by focusing on a few key locations. For instance, when viewing an image, the brain concentrates more intently on a particular area of interest. The human visual system served as the model for the attention mechanism, which was adapted for neural machine translation [11] and image processing [34]. With a weighted sum approach based on a context vector, Concentrating on particular features are the main goal of the training phase. At each time step t , the context vector V is defined in the following manner:

$$\mathbf{V}_t = \Sigma \alpha_t . \mathbf{h}_t \tag{1}$$

A recurrent network is typically used, and The model that generates the hidden states and feeds them into the attention model is represented by h_t. The term α_t stands for the weights from the normalized attention model calculated as follows:

$$\alpha_t = softmax(e_t) \tag{2}$$

The attention model's weights are based on alignment scores, which are represented by the symbol e_t., are calculated using a feedforward neural network, and are dependent on previously hidden state h_{t-1}.

$$\mathbf{e}_t = \sigma(\mathbf{W}_a \mathbf{h}_{t-1} + \mathbf{b}_a) \tag{3}$$

where (W_a, b_a) is the attention model's weights matrix and bias vector, respectively, that were calculated during training. In fact, the pertinent segment of the time-series input at time t is dynamically represented by the attention model context vector V. It is computed as a normalized, probabilistic attention model weight that represents the importance of a specific data point in the sequence through a weighted sum. The attention mechanism enables the learning process to concentrate on particular portions within the data sequence by going back to its memory during the prediction phase. This makes it possible for the model to emphasize critical components within the Feature area, improving its capacity to recognize significant patterns and make precise predictions [24].In the neural network layer, only one layer is used for additive attention. A feed-forward network with a hyperbolic tangent (tanh) nonlinearity serving as the default activation function makes up this layer. The network's ability to concentrate on pertinent information and recognize significant patterns is improved by the additive attention mechanism, which enables the network to dynamically assign relevance or weights to various segments of the input sequence. Contrarily, matrix multiplications are

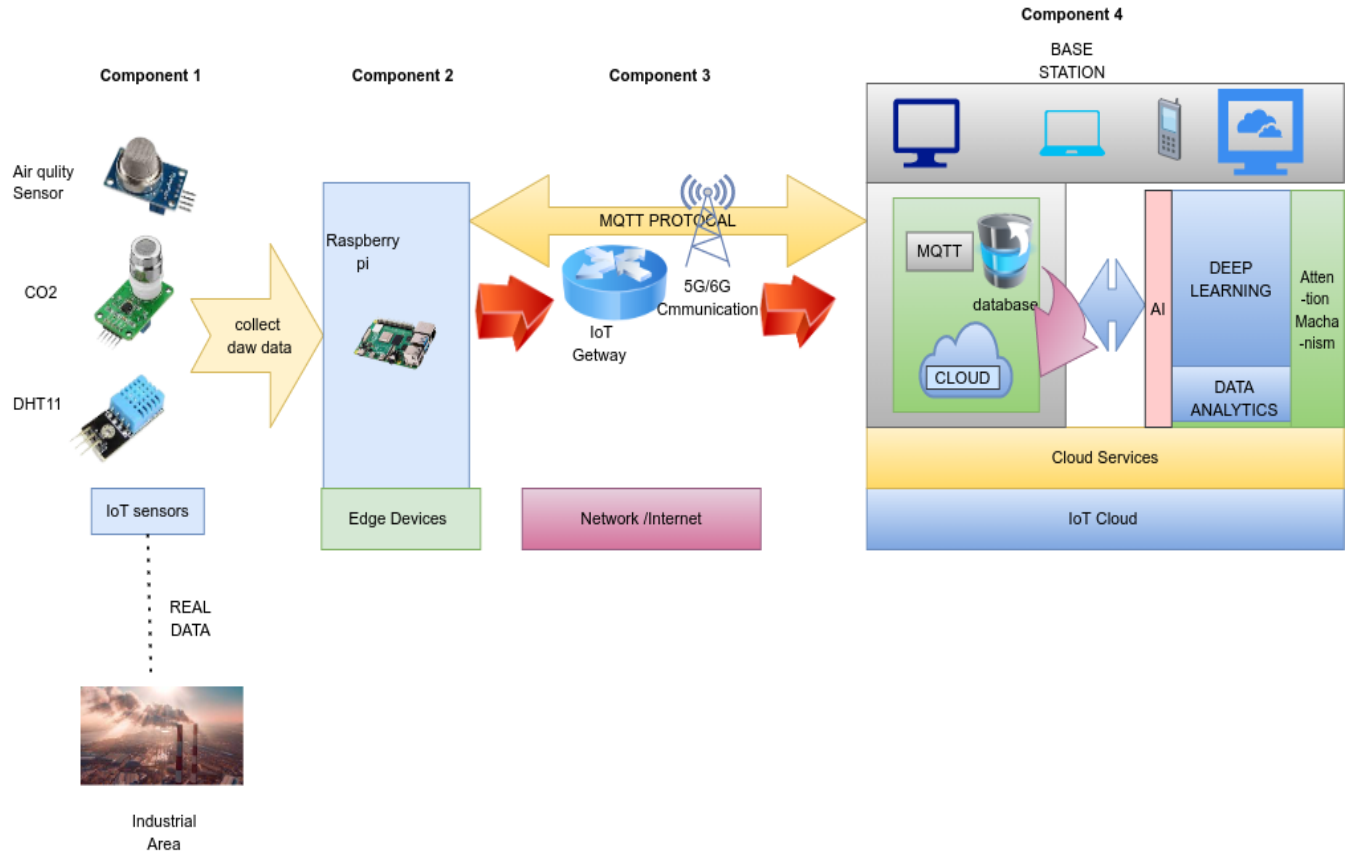


Fig. 1: ARCHITECTURE OF PM 2.5 AIR QUALITY PREDICTION

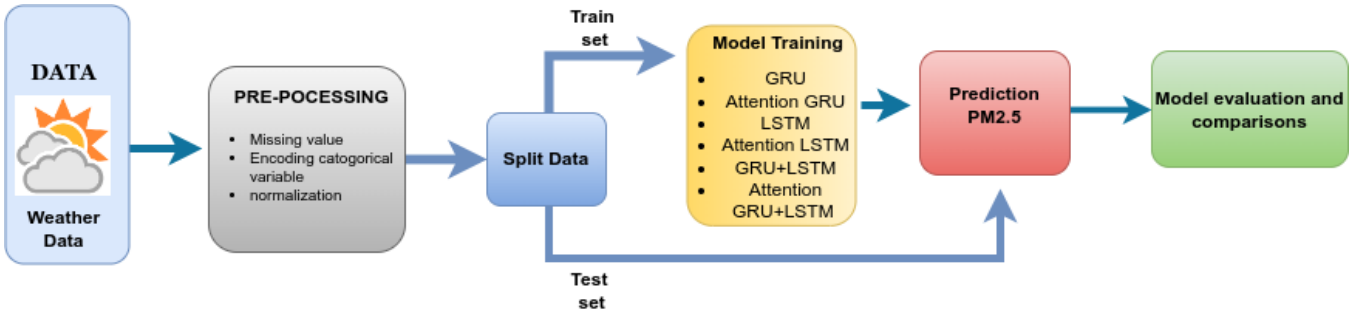


Fig. 2: Workflow for predicting PM2.5 concentrations

used in multiplicative attention to reduce the hidden states and compute the attention scores; the dot product can then be activated using various activation functions with tunable parameters.

B. Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) was used in this study to make predictions. The RNN (Recurrent Neural Network) is a particular kind of neural network created especially for handling sequential input. For jobs involving sequence prediction, it is renowned for its effectiveness. The ability of RNNs to preserve hidden states or memory that can recognize temporal connections within the input allows them to excel at modeling and comprehending sequential patterns [27], [35]. The vanishing gradient problem and the inflating gradient problem are the two main problems that can affect simple RNNs. It is difficult for RNNs to successfully learn long-term dependencies in sequences as a result of these issues.

When the gradients generated during the backpropagation step decrease exponentially as they spread over time, it is known as the ”vanishing gradient problem.” This problem makes it difficult for the RNN to gather and spread information across lengthy sequences, which makes it challenging to understand long-term dependencies. The exploding gradient issue, on the other hand, appears when the gradients grow exponentially during backpropagation. As a result, the network’s weights may train in an unstable manner, making it challenging to successfully optimize the model. Hence GRU provides solutions to the above challenges. In GRU, the

Update gate and reset gate are the two gate units that make up each GRU memory cell; equations (4) – (7) define the operators to store the data for the various states. There are multiple variables at play in equations 4 through 7. In these equations, the input is denoted by x_t and the hidden state is denoted by h_t. In addition, the variables r_t and z_t stand for the reset and update gates, respectively, and h_t is used to store the updated value of the hidden state. The proposed architecture of

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \textbf{ [Reset gate]} \quad (4)$$

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \textbf{ [Update gate]} \quad (5)$$

$$h_t{}^{\sim} = \tanh(W_h.[r_t.h_{t-1}, x_t] + b_h) \textbf{ [Updating value]} \quad (6)$$

$$h_t = (1 - z_t).h_{t-1} + z_t.h_t{}^{\sim} \textbf{[Hidden state update]} \quad (7)$$

the GRU model is shown in figure 5. It has three GRU layers, each followed by a dropout layer, then two dense layers, and finally, an output layer. The description of each layer is given below :

GRU Layer: The update gate, reset gate, and current memory gate are the three gates that make up the GRU (Gated Recurrent Unit) layer. To process sequential data, each gate is essential. The following is a list of each of their roles:

1) **Update gate:** A key factor in determining how much future information will rely on historical data is the update gate. It controls the information flow from the earlier concealed state to the present hidden state, much to the output gate in an LSTM. The GRU may update its hidden state selectively, conserving pertinent information while discarding irrelevant or out-of-date information, thanks to the flexibility to modify the update gate.
2) **Reset gate:** It is used to establish the degree to which the old data should be forgotten. It determines how much the prior concealed state should be factored into the present hidden state. The GRU is able to adapt to shifting patterns in the data by choosing which elements of the prior hidden state to ignore by modifying the reset gate.
3) **Current memory gate:** The reset gate's current memory gate, though frequently disregarded, is a crucial part of the system. The mean of the input is set to zero and non-linearity is introduced. It aids the GRU in focusing on the most pertinent and recent information by minimizing the impact of earlier information on the present information. Information is communicated to the GRU's future states via the memory gate now in use.

In this paper, the first layer of GRU consists of 64 neurons with a return sequence set to true, depicting whether to return the last output in the output sequence or the full sequence. The second and third GRU layer consists of 32 and 16 neurons, respectively, with the second layer being set to true to return the output sequence.

Dropout : To address the issue of overfitting in the system, a dropout layer is added after each GRU layer. This dropout layer helps in reducing the likelihood of overfitting by randomly deactivating a portion of the GRU layer's units during training.

Dense Layer : As depicted in Figure 5, the fully-connected layer is employed to establish connections between the 16 neurons of the third GRU network layer and all 64 neurons of the subsequent layer. In order to combine and turn the retrieved characteristics into meaningful predictions or outputs in the following fusion layer, the fully-connected layer is essential.

Output Layer : This layer will be responsible for determining the output corresponding to the tested input.

C. Long Short-Term Memory (LSTM)

When it comes to capturing long-range dependencies in sequential data, LSTMs address the shortcomings of conventional RNNs. They solve the vanishing gradient issue and enable the network to store and transmit data over a wider time range. The memory cell is the most important part of an LSTM since it has long-term memory. Each memory cell of LSTM has 3 gate components: The forget gate, the input gate, and the output gate.

The input gate regulates the entry of fresh information into the memory cell, while the forget gate decides which information to discard. The output gate governs the flow of information from the cell. Memory cells play a vital role in enabling LSTM to store information for extended periods and to obtain sequential patterns in time-series data more effectively. All parameters used in LSTM are defined using the following equations.

$$i_t = \sigma(W_i.h_{t-1} + x_t + b_i)\textbf{(Input gate)} \quad (8)$$

$$f_t = \sigma(W_f.h_{t-1} + x_t + b_f)\textbf{(Forget gate)} \quad (9)$$

$$o_t = \sigma(W_o.h_{t-1} + x_t + b_o)\textbf{(Output gate)} \quad (10)$$

$$c_t{}^{\sim} = \tanh(W_c.h_{t-1}, x_t] + b_c))\textbf{(Memory cell candidate)} \quad (11)$$

$$c_t = f_t.c_{t-1} + i_t.c_t{}^{\sim}\textbf{(Memory cell)} \quad (12)$$

$$h_t = o_t.\tanh(c_t)\textbf{(shadow state)} \quad (13)$$

$$y_t = h_t\textbf{(cell output)} \quad (14)$$

The input gate is i_t, the output gate is o_t, and the forget gate is denoted by f_t. The time step is denoted by the index t. As a result, the input and hidden states at this time are x_t and h_t, respectively. Cell status at the moment is represented by c_t, and its updated value is represented by c_tCAP. The cell state's updated value is regulated by the hyperbolic tangent function, while the sigmoid function governs the activation of the forget gate, input gate, and output gate. The weight matrices assigned to the forget gate, input gate, output gate, and cell state are denoted as W_f, W_i, W_o, and W_c, respectively, while the corresponding bias matrices are represented by b_f, b_i, b_o, and b_c".

The LSTM model is shown in the above figure 6 using all the equations from (8-14). It has three LSTM layers, each followed by a dropout layer, then two dense layers, and finally, an output layer. The model is trained, and weights are adjusted after back-propagation. The description of each layer model is given below :

Input Layer: The input is of dimensions (35036, 4, 1) depicting a total of 35040 = 4*24*356 rows, where each row corresponds to every hour in a day for a year. Every row has information from the previous three hours of data since it is a windowed data set. LSTM Layer: Our data set has 8 features associated with PM2.5 concentration. The windowed data with the information from the previous 3 hours will enter the hidden layer of LSTM. Our first LSTM layer consists of 64 neurons with a return sequence set to true, depicting whether to return the last output in the output sequence or the full sequence. The second and third LSTM layer consists of 32 and 16 neurons, respectively, with the second layer being set to true to return sequence. Dropout: To prevent over-fitting in our LSTM model, we have incorporated dropout layers after each LSTM layer. This is accomplished by randomly deactivating 50% of the cells, which lessens the neural network's propensity to learn complex patterns throughout each learning session that are unique to the training data but inapplicable to fresh data. The dropout layer assists in making the neurons less trustworthy and prevents over-fitting, while the weights of the active neurons are then used for the total neural network, including the update weights. In the network, the dropped-out neurons are represented by solid lines.

Dense Layer: A fully linked layer is incorporated into the neural network design in figure 6. The 16 neurons from the third LSTM network layer are linked with all 16 neurons in the fully-connected layer in this particular instance by the fully-connected layer.

Output Layer: As shown in figure 6, the completely integrated layer will then be linked to the fusion layer's prediction feature data. The anticipated PM2.5 value for the upcoming 48 hours will be obtained by back-propagating the hidden layer's weights. The final estimated PM2.5 value will be derived by combining the resulting data with the neural network's output. The projected characteristics will be given various weights inside the final layer in order to emphasize the association between each feature and the local station data.

D. Proposed model (Attention_GRU+LSTM)

The proposed architecture of the Attention_GRU-LSTM model is shown in the figure 3 It have two GRU layers, each followed by a dropout layer, then two LSTM layers followed by dropout layers, then three dense layers, and finally, an output layer. After the last phase, we use an attention mechanism to concentrate on risk information for predicting air pollution. Relu served as the system's activation function. The description of each the layer is given below :

GRU Layer: Our first and second GRU layer consists of 64 neurons and 32 neurons respectively with the return sequence set to true, depicting whether to return the last output in the output sequence or the full sequence.

Dropout: To prevent over-fitting, we inserted dropout layers following each GRU and LSTM layer.

Dense Layer:The 16 neurons in the second LSTM network layer are all connected to the 16 neurons in the fully-connected layer in Figure 3. With the use of this link, the fully-connected layer's weights and activation function may be created. These functions are then used in the following fusion layer.

Output Layer: This layer will be responsible for determining the output corresponding to the test input.

IV. Performance Analysis

Common statistical methods like the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R-Squared) are frequently used to assess the accuracy of forecasts. These techniques help to measure the discrepancy between actual and expected values. In the current study, we carried out a comparative analysis of three models—GRU, LSTM, and GRU-LSTM—across three error measures, including RMSE, MAE, and MAPE [12].

1) Root Mean Square Error (RMSE): RMSE measures the difference between the predicted and actual values and is typically used to assess forecasts over different time intervals. A larger RMSE value indicates a greater degree of misalignment between the predicted and actual values

$$\mathrm{RMSE}(y, \hat{\mathbf{y}}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}} \tag{4}$$

2) Mean Absolute Error (MAE): MAE measures the average absolute difference between two values without considering whether the differences are positive or negative. It is an absolute measure of dispersion and does not involve any phase cancellation. Therefore, the average error rate is the same as the mean absolute error.

$$\mathrm{MAE}(y, \hat{\mathbf{y}}) = \frac{\sum_{i=0}^{N-1}|y_i - \hat{y}_i|}{N} \tag{5}$$

3) R-Squared : The coefficient of determination (R^2) is a metric that evaluates the effectiveness of a statistical model in predicting an outcome based on its dependent variable. R^2 ranges from 0 (worst) to 1 (best), indicating the model's accuracy level in making predictions. In other words, the higher the R^2 value, the better the model predicts the outcome.

$$\mathrm{R2}(y, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - y)^2} \tag{6}$$

4) MSE : One of the most used evaluation metrics is a mean squared error (MSE). The following calculation demonstrates that the residual sum of squares and MSE are strongly related. The distinction is that your focus now is on the average error rather than the total error.

$$\mathbf{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{7}$$

V. EXPERIMENTS AND RESULTS

In-depth information regarding the experimental conditions and execution process is provided in this section, which also discusses the data utilized in this paper. Additionally, we offer a discussion and an analysis of the outcomes.

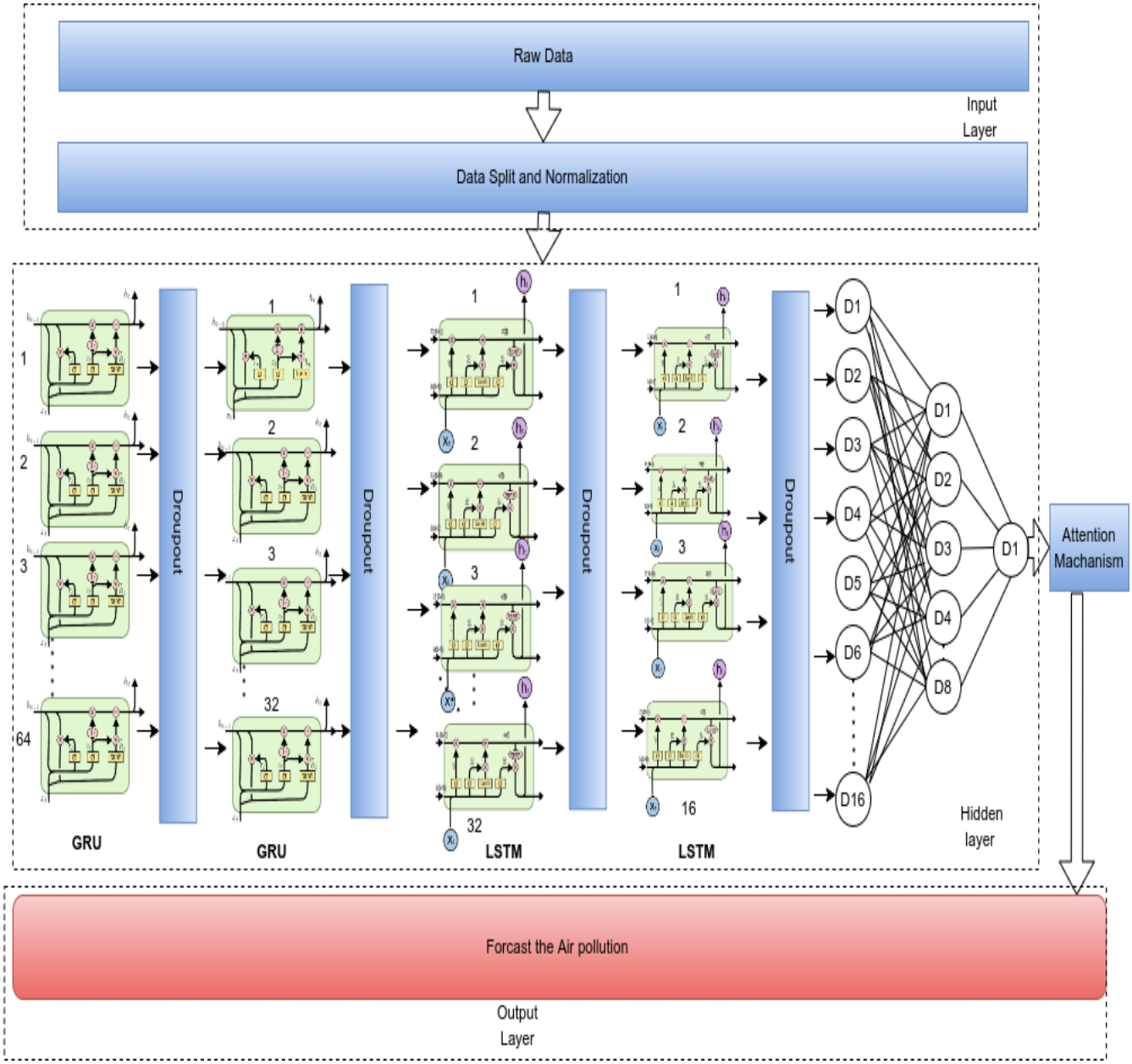


Fig. 3: Proposed Attention_GRU-LSTM Model

1) Data Collection: We used research data set available on the website https://www.epa.gov/outdoor-air-qualitydata/download-daily-data [1].

The pollutants were measured over a specific five-year timeframe. The dataset comprises data on weather conditions and pollution levels at the US embassy in Beijing, China, collected hourly for a duration of five years. It is a time series dataset that includes information on PM2.5 concentration levels, as well as weather factors such as dew point, temperature, pressure, wind direction, wind speed, and cumulative hours of snow and rain. A detailed breakdown of the dataset columns can be found in Figure 4. This figure presents descriptive statistics for the ambient air pollution datasets in Beijing. A perfectly symmetric distribution, including the normal distribution, exhibits a skewness of zero and a kurtosis of 3. However, Figure 4 indicates that the time-series dataset for ambient air pollution in Beijing does not follow a Gaussian distribution, displaying positive support and varying intervals of variability.

A. Data pre-processing and normalization

At first, all categorical values are converted into numeric values. Label Encoding is a technique used to convert categorical columns into numerical columns so that they can be fitted by machine learning models that only take numerical data. It is an important pre-processing step in a machine-learning project. Label encoding converts categorical data into numerical data by assigning a unique number(starting from 0) to each class of data. Then, feature scaling is performed on the dataset by using Min-Max Scaling. Min-Max Scaling is a data scaling technique that sets the minimum value of

	pollution	dew	temp	pressure	w_speed	snow	rain
count	43800.000000	43800.000000	43800.000000	43800.000000	43800.000000	43800.000000	43800.000000
mean	94.013516	1.828516	12.459041	1016.447306	23.894307	0.052763	0.195023
std	92.252276	14.429326	12.193384	10.271411	50.022729	0.760582	1.416247
min	0.000000	-40.000000	-19.000000	991.000000	0.450000	0.000000	0.000000
25%	24.000000	-10.000000	2.000000	1008.000000	1.790000	0.000000	0.000000
50%	68.000000	2.000000	14.000000	1016.000000	5.370000	0.000000	0.000000
75%	132.250000	15.000000	23.000000	1025.000000	21.910000	0.000000	0.000000
max	994.000000	28.000000	42.000000	1046.000000	585.600000	27.000000	36.000000

Fig. 4: AMBIENT AIR POLLUTION DATASETS: STATISTICS SUMMARY

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 4, 64)	14208
dropout (Dropout)	(None, 4, 64)	0
gru_1 (GRU)	(None, 4, 32)	9408
dropout_1 (Dropout)	(None, 4, 32)	0
gru_2 (GRU)	(None, 16)	2400
dropout_2 (Dropout)	(None, 16)	0
dense (Dense)	(None, 64)	1088
dense_1 (Dense)	(None, 16)	1040
dense_2 (Dense)	(None, 1)	17

Fig. 5: Architecture of GRU

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 4, 64)	18688
dropout_3 (Dropout)	(None, 4, 64)	0
lstm_4 (LSTM)	(None, 4, 32)	12416
dropout_4 (Dropout)	(None, 4, 32)	0
lstm_5 (LSTM)	(None, 16)	3136
dropout_5 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 16)	272
dense_4 (Dense)	(None, 1)	17

Fig. 6: Architecture of LSTM

a feature to zero and the maximum value to one, as shown in Equation 1. This method compresses the data to a specific range, typically between 0 and 1, without altering the shape of the original distribution. Scaling the features to a predefined range transforms the data, making it easier to compare and analyze.

Layer (type)	Output Shape	Param #
gru_3 (GRU)	(None, 4, 64)	14208
dropout_6 (Dropout)	(None, 4, 64)	0
gru_4 (GRU)	(None, 4, 32)	9408
dropout_7 (Dropout)	(None, 4, 32)	0
lstm_3 (LSTM)	(None, 4, 32)	8320
dropout_8 (Dropout)	(None, 4, 32)	0
lstm_4 (LSTM)	(None, 16)	3136
dropout_9 (Dropout)	(None, 16)	0
dense_5 (Dense)	(None, 16)	272
dense_6 (Dense)	(None, 8)	136
dense_7 (Dense)	(None, 1)	9

Fig. 7: Architecture of GRU+LSTM

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{8}$$

B. Co-Relation

Air quality is influenced by weather parameters such as atmospheric pressure, temperature, and relative humidity. For instance, high wind speed will lower PM2.5 concentrations, high humidity will typically worsen air pollution, and high air pressure will eventually improve air quality. Therefore, the major challenges of forecasting air quality heavily depend on meteorological factors. Figure 14 shows all the co-relation of the different air quality parameters.

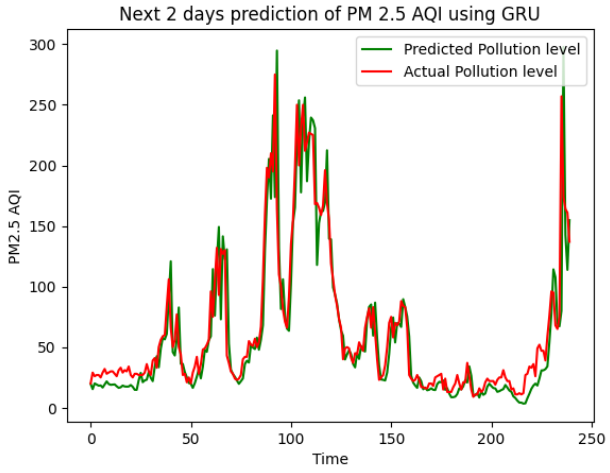
Next 2 days prediction				
Model	**RMSE**	**R squared**	**MSE**	**MAE**
GRU	12.568367	0.472556	157.963852	8.961138
Attention GRU	14.106950	0.335515	199.006027	11.002945
LSTM	12.657800	0.465023	160.219894	8.304810
Attention LSTM	12.657800	0.465023	160.219894	8.304810
GRU+LSTM	12.657800	0.465023	160.219894	8.304810
Attention GRU+LSTM	**10.735380**	**0.615183**	**115.248390**	**6.401103**
Next 5 days prediction				
GRU	27.115187	0.841551	735.233337	17.752413
Attention GRU	28.591871	0.823823	817.495056	18.446920
LSTM	26.031569	0.853962	677.642578	16.476482
Attention LSTM	27.122917	0.841460	735.652588	17.353659
GRU+LSTM	30.624588	0.797882	937.865356	17.158680
Attention GRU+LSTM	**27.649176**	**0.835249**	**764.476929**	**16.498899**
Next 10 days prediction				
GRU	25.467136	0.808499	648.575012	14.936353
Attention GRU	25.865757	0.802457	669.037354	14.686987
LSTM	24.359396	0.824796	593.380188	14.482867
Attention LSTM	24.450315	0.823486	597.817932	14.199401
GRU+LSTM	27.669481	0.773946	765.600159	14.351439
Attention GRU+LSTM	**24.778614**	**0.818714**	**613.979675**	**13.351772**

TABLE I: PERFORMANCE COMPARISON OF THE PROPOSED APPROACH(Attention GRU+LSTM)

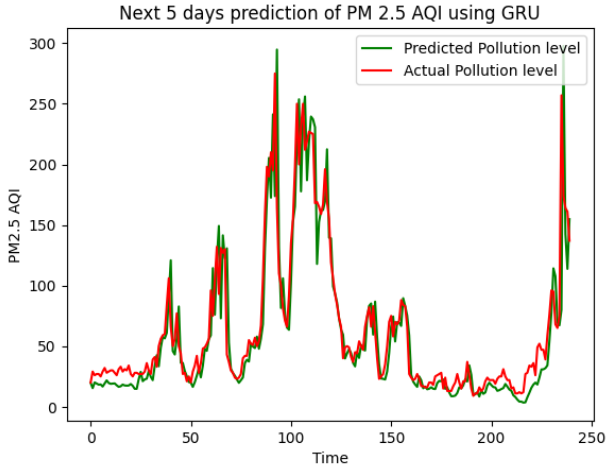
C. Experiment Setup and Results

The first experiment series assesses how well the examined models predict PM 2.5 on different days. To achieve this, each model in this uni-variate forecasting is supervised and trained to understand all of the temporal associations present in the time-series data measurement of each pollutant. An Intel i7 CPU with a quick algorithm-based approach and 16GB of RAM was used to implement this research. Under Ubuntu 18 LTS, we utilized Python 2.7.0 with Tensorflow/Keras: 2.12.0, pandas: 1.5.3, numpy: 1.22.4, sklearn: 1.2.2, and plotly: 5.13.1. The collection of hyper-parameters used in these experiments using the training data-sets is constant across all models, Specifically, the batch size is 32, and there are 100 epochs. The proposed model has two GRU layers(64,32), followed by two dropouts, each with 64 and 32 kernel units respectively shown in figure 3. The proposed model consists of two 64 by 32 kernel GRU layers and two 32 by 16 kernel LSTM layers with dropout to prevent over-fitting, respectively. The typical back-propagation approach has been utilized to fine-tune model parameters [17], [18], [26]. The collected testing data validation metrics for the years 2010 to 2014 are listed in Table I.

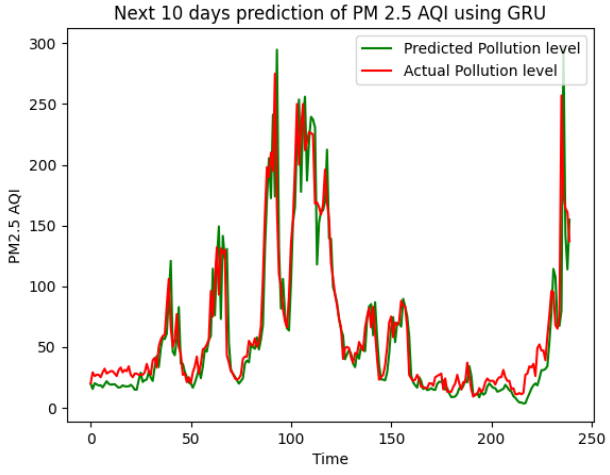
The Attention _GRU+LSTM method has been found to outperform other solutions for this particular univariate time-series forecasting problem after a variety of validation criteria, including RMSE, R-squared, MSE, and MAE, was evaluated. Based on the evaluation criteria, this method is the best option because it exhibits great efficiency and acceptable accuracy. On forecasting all examined pollutants observed from 2010 to 2014, the suggested Attention_GRU+LSTM strategy outperforms the recurrent models (i.e., GRU, LSTM, GRU+LSTM, Attention_GRU, and Attention _LSTM). The proposed Attention_GRU+LSTM strategy, as achieved, has the best R2 score and the lowest forecasting errors (RMSE, MAE, MAE, and MSE). It might be credited to its ability to identify pertinent characteristics utilizing the attention technique while modeling time dependencies. It's important to note that the performance of the attention mechanisms—Attention _GRU and Attention _LSTM—is better than that of the LSTM and GRU-represented unidirectional models. For the next two days' forecasting, the proposed method (Attention GRU+LSTM) had the lowest RMSE (=10.735380), MEA (=6.401103), MSE (=115.248390), and R Square (=0.615183). Additionally, it is visible for 5 and 10 days in Table I. The values of RMSE, R squared, MSE, and MAE for five days are 27.649176, 0.835249, 764.476929, and 16.498899, respectively, as given in Table I. Figures 5 depict the forecast using the model GRU for days 2, 5, and 10 accordingly. Use the same model and apply the Attention Mechanism to discover the findings in Figures 9 to achieve better results than the GRU model. On-time series data, LSTM models deliver the best results. Since we were working with the same dataset, we applied it and discovered the performance matrices in Table I and Figures 10, which illustrate the results of our forecasts for 2 days, 5 days, and 10 days, respectively. We used the attention mechanism on the same model to improve the results shown in Figures



(a)

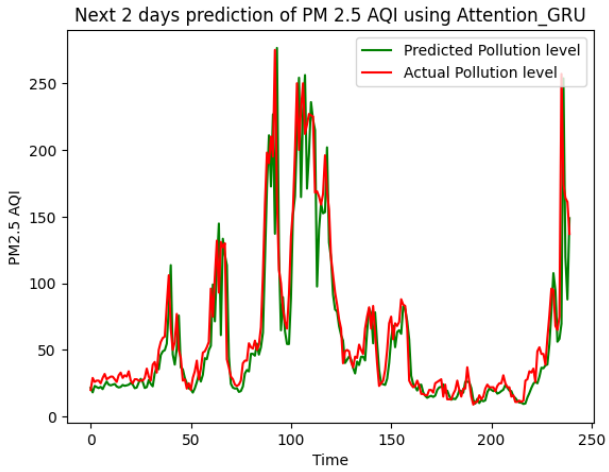


(b)

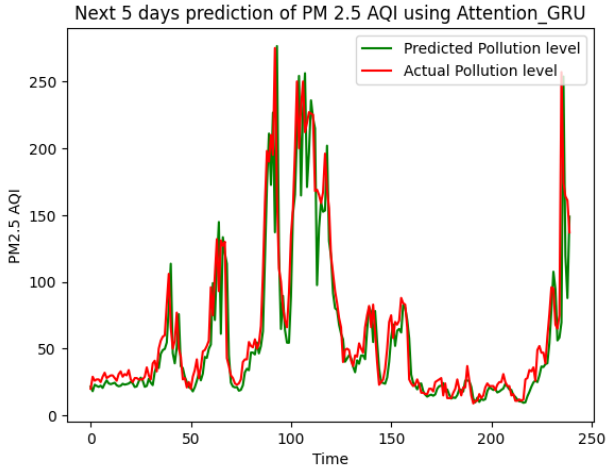


(c)

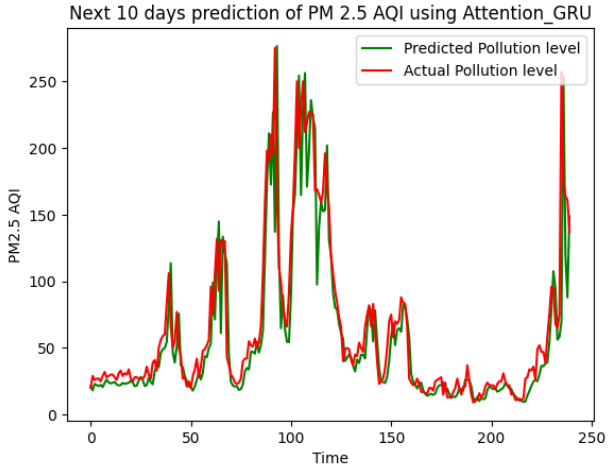
Fig. 8: (a) Using GRU, the next 2 days' PM 2.5 AQI forecasted.
(b) Using GRU, the next 5 days' PM 2.5 AQI forecasted.
(c) Using GRU, the next 10 days' PM 2.5 AQI forecasted.



(a)



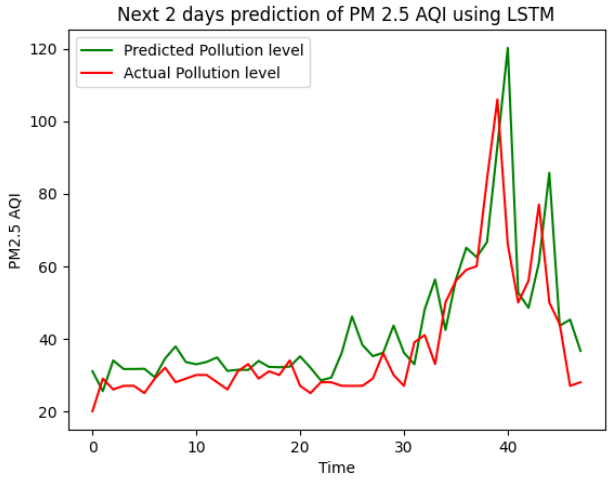
(b)



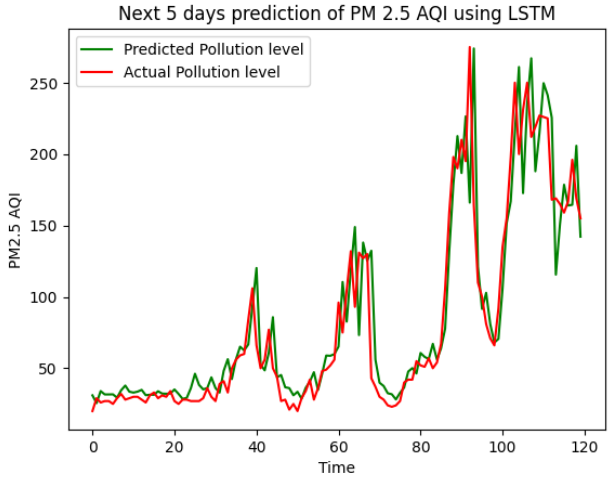
(c)

Fig. 9: (a)Using Attention _GRU, the next 2 days' PM 2.5 AQI forecasted. (b) Using Attention _GRU, the next 5 days' PM 2.5 AQI forecasted.(c) Using Attention _GRU, the next 10 days' PM 2.5 AQI forecasted.

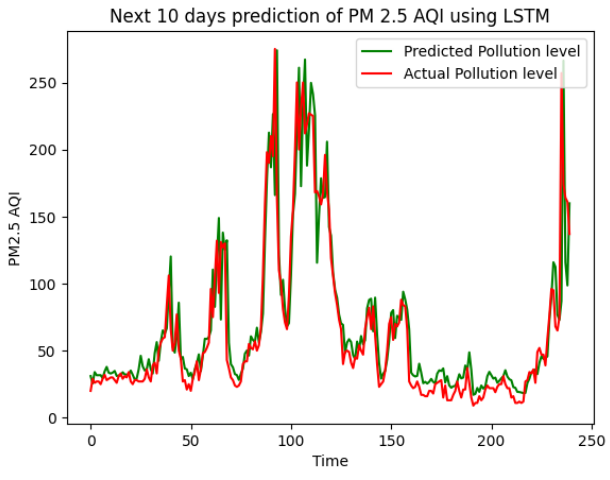
11 as a result.



(a)

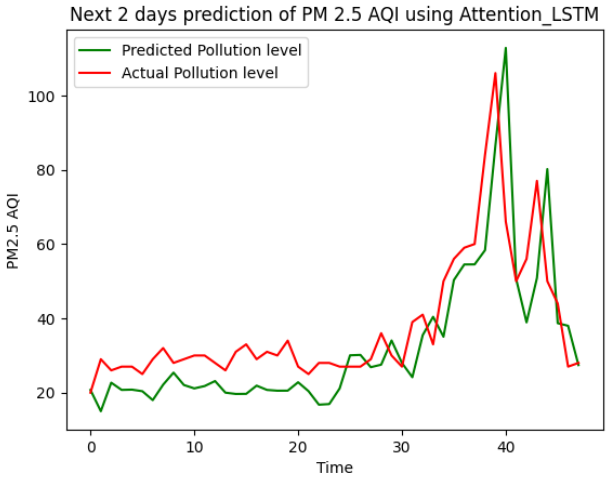


(b)

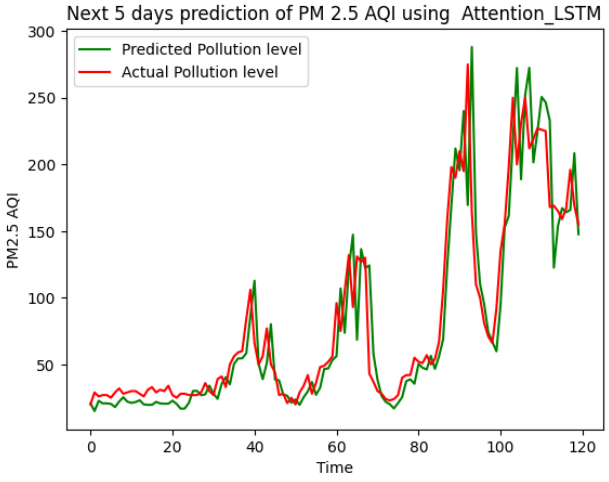


(c)

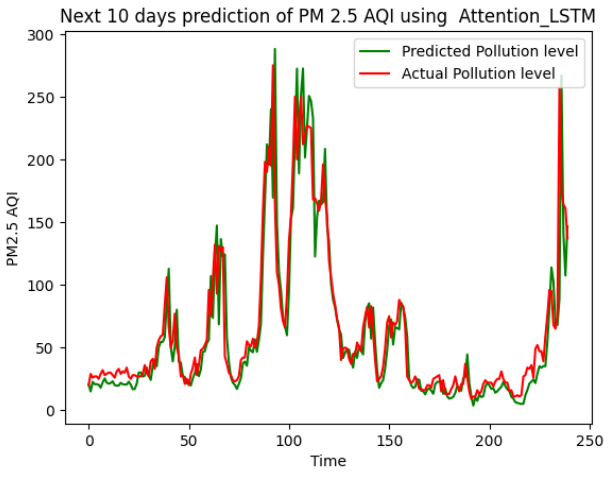
Fig. 10: (a)Using LSTM, the next 2 days' PM 2.5 AQI forecasted
(b) Using LSTM, the next 5 days' PM 2.5 AQI forecasted
(c)Using LSTM, the next 10 days' PM 2.5 AQI forecasted



(a)

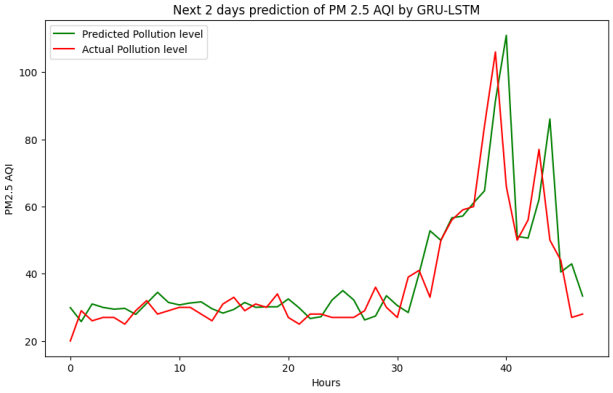


(b)

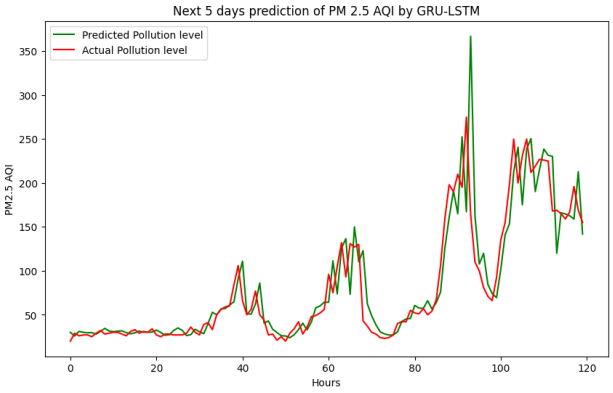


(c)

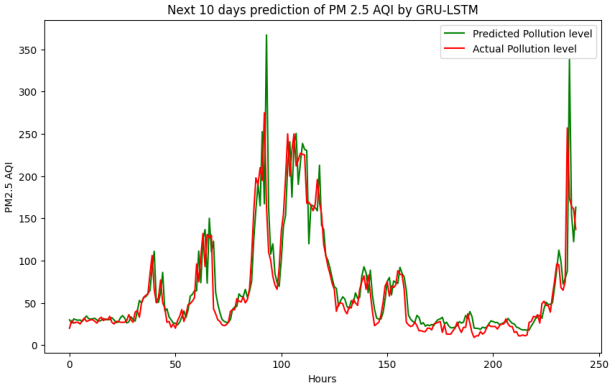
Fig. 11: (a) Using Attention _LSTM, the next 2 days' PM 2.5 AQI forecasted. (b) Using Attention _LSTM, the next 5 days' PM 2.5 AQI forecasted (c) Using Attention_LSTM, the next 10 days' PM 2.5 AQI forecasted



(a)

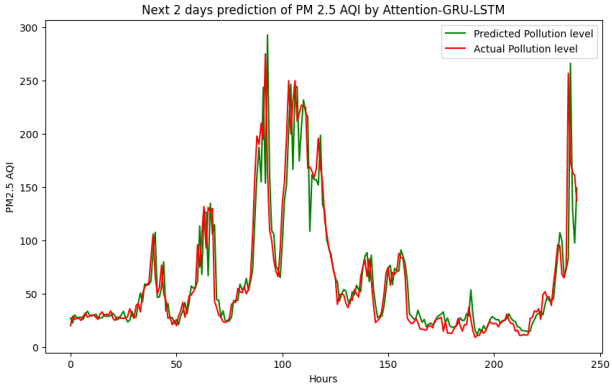


(b)

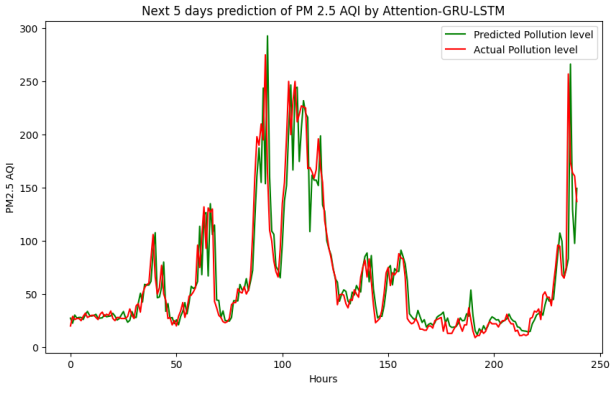


(c)

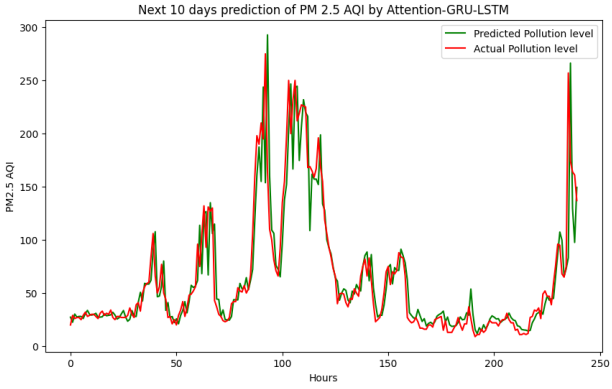
Fig. 12: (a)Using GRU+LSTM, the next 2 days' PM 2.5 AQI forecasted.
(b) Using GRU+LSTM, the next 5 days' PM 2.5 AQI forecasted.
(c) Using GRU+LSTM, the next 10 days' PM 2.5 AQI forecasted.



(a)



(b)



(c)

Fig. 13: (a) Using Attention_GRU+LSTM, the next 2 days' PM 2.5 AQI forecasted.
(b) Using Attention_GRU+LSTM, the next 5 days' PM 2.5 AQI forecasted
(c)Using Attention_GRU+LSTM, the next 10 days' PM 2.5 AQI forecasted

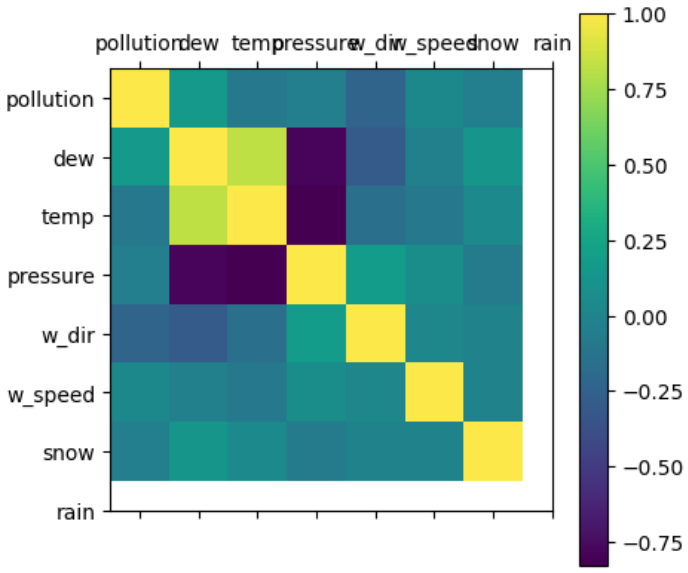


Fig. 14: The correlation matrix of the Beijing, China air quality features

Table I shows that the outcomes have fallen short of expectations, necessitating using a hybrid model to increase the system's precision. Therefore, we combine the GRU and LSTM models and forecast the outcomes depicted in Figures 12. We used the Attention _GRU+LSTM model since we needed a new proposed model to increase the accuracy of the entire system. The performance reported in Table 1 and the predicted outcomes for the next two, five, and ten days are in Figures 13.

VI. CONCLUSION

Most parts of the world are afflicted by air pollution, which is known to have harmful effects on health. The rapid advancement of industrial technology has had several adverse impacts on the environment. To ensure adequate air quality, it is crucial to monitor the surrounding environment. In this paper, developed a unique model to improve air pollution forecasting by adding an attention mechanism to the GRU+LSTM (Attention_GRU+LSTM). We carried out validation in this study to evaluate the performance using a performance metric approach. The outcomes depict, the Attention _GRU+LSTM model performed remarkably well in predicting PM 2.5 concentrations for forecasting periods of 2, 5, and 10. The model outperformed other approaches in terms of accuracy, demonstrating its effectiveness in predicting air pollution levels over long time horizons. RMSE, R squared, MSE, and MAE is four statistical metrics used to evaluate forecasting accuracy. Metrics showed that deep hybrid models, which focus on modeling temporal dependencies in unsupervised learning without the need for advanced, recurrent networks with gating and memory mechanisms, are highly effective. In time-dependent modeling, the use of variational inference approximation has yielded encouraging results. The research reported in this paper serves as a foundation for further research even though the proposed Attention _GRU+LSTM model has shown satisfactory predicting results for ambient air pollution. It provides opportunities for future investigation and development of forecasting methods, with the possibility to use variational inference and other cutting-edge techniques to improve the precision and capabilities of time-dependent modeling for air pollution prediction. The suggested Attention _GRU+LSTM based forecasting model will be enhanced by the development of a multiscale Attention _GRU+LSTM model that combines Attention _GRU+LSTM methods with wavelet-based multiresolution representation. Measurements of pollution may include noisy components that vary over time and contribute to various frequency components. Incorporating explanatory factors, such as meteorological information, when building deep learning models is another area for future advancement. It will also be intriguing to develop a system for early abnormal pollution identification in order to promote reactive control and enable avoiding exposure to abnormal pollution with high concentrations.

REFERENCES

[1] https://www.epa.gov/outdoor-air-qualitydata/download-daily-data.
[2] MSK Abhilash, Amrita Thakur, Deepa Gupta, and B Sreevidya. Time series analysis of air pollution in bengaluru using arima model. In *Ambient Communications and Computer Systems: RACCCS 2017*, pages 413–426. Springer, 2018.
[3] Samaher Al-Janabi, Mustafa Mohammad, and Ali Al-Sultan. A new method for prediction of air pollution based on intelligent computation. *Soft Computing*, 24(1):661–680, 2020.
[4] Sharafat Ali, Tyrel Glass, Baden Parr, Johan Potgieter, and Fakhrul Alam. Low cost sensor with iot lorawan connectivity and machine learning-based calibration for air pollution monitoring. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2020.
[5] Davor Antanasijević, Viktor Pocajt, Aleksandra Perić-Grujić, and Mirjana Ristić. Multiple-input–multiple-output general regression neural networks model for the simultaneous estimation of traffic-related air pollutant emissions. *Atmospheric Pollution Research*, 9(2):388–397, 2018.
[6] Davor Antanasijević, Viktor Pocajt, Aleksandra Perić-Grujić, and Mirjana Ristić. Multiple-input–multiple-output general regression neural networks model for the simultaneous estimation of traffic-related air pollutant emissions. *Atmospheric Pollution Research*, 9(2):388–397, 2018.
[7] Mohammad Arhami, Nima Kamali, and Mohammad Mahdi Rajabi. Predicting hourly air pollutant levels using artificial neural networks coupled with uncertainty analysis by monte carlo simulations. *Environmental Science and Pollution Research*, 20:4777–4789, 2013.
[8] Mirche Arsov, Eftim Zdravevski, Petre Lameski, Roberto Corizzo, Nikola Koteli, Kosta Mitreski, and Vladimir Trajkovik. Short-term air pollution forecasting based on environmental factors and deep learning models. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 15–22. IEEE, 2020.
[9] Mirche Arsov, Eftim Zdravevski, Petre Lameski, Roberto Corizzo, Nikola Koteli, Kosta Mitreski, and Vladimir Trajkovik. Short-term air pollution forecasting based on environmental factors and deep learning models. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 15–22. IEEE, 2020.
[10] V Athira, P Geetha, Rab Vinayakumar, and KP Soman. Deepairnet: Applying recurrent networks for air quality prediction. *Procedia computer science*, 132:1394–1403, 2018.
[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
[12] Abdelkader Dairi, Fouzi Harrou, Sofiane Khadraoui, and Ying Sun. Integrated multiple directed attention-based deep learning for improved air pollution forecasting. *IEEE Transactions on Instrumentation and Measurement*, 70:1–15, 2021.
[13] Ke Gu, Zhifang Xia, and Junfei Qiao. Stacked selective ensemble for pm 2.5 forecast. *IEEE Transactions on Instrumentation and Measurement*, 69(3):660–671, 2019.

[14] Fouzi Harrou, Lionel Fillatre, Michel Bobbia, and Igor Nikiforov. Statistical detection of abnormal ozone measurements based on constrained generalized likelihood ratio test. In *52nd IEEE Conference on Decision and Control*, pages 4997–5002. IEEE, 2013.

[15] Fouzi Harrou, Farid Kadri, Sofiane Khadraoui, and Ying Sun. Ozone measurements monitoring using data-based approach. *Process Safety and Environmental Protection*, 100:220–231, 2016.

[16] Wilmar Hernandez, Alfredo Mendez, Rasa Zalakeviciute, and Angela Maria Diaz-Marquez. Analysis of the information obtained from pm 2.5 concentration measurements in an urban park. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6296–6311, 2020.

[17] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. *Neural Networks: Tricks of the Trade: Second Edition*, pages 599–619, 2012.

[18] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[19] Ping Jiang, Chen Li, Ranran Li, and Hufang Yang. An innovative hybrid air pollution early-warning system based on pollutants forecasting and extenics evaluation. *Knowledge-Based Systems*, 164:174–192, 2019.

[20] Ujjwal Kumar and VK Jain. Arima forecasting of ambient air pollutants (o 3, no, no 2 and co). *Stochastic Environmental Research and Risk Assessment*, 24:751–760, 2010.

[21] Xiang Li, Ling Peng, Yuan Hu, Jing Shao, and Tianhe Chi. Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research*, 23:22408–22417, 2016.

[22] Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, and Yuan Hu. Chengzeng you, and tianhe chi. long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental pollution*, 231:997–1004, 2017.

[23] Hui Liu, Haiping Wu, Xinwei Lv, Zhiren Ren, Min Liu, Yanfei Li, and Huipeng Shi. An intelligent hybrid model for air pollutant concentrations forecasting: Case of beijing in china. *Sustainable Cities and Society*, 47:101471, 2019.

[24] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[25] Amit Prakash, Ujjwal Kumar, Krishan Kumar, and VK Jain. A wavelet-based neural network model to predict ambient air pollutants' concentration. *Environmental Modeling & Assessment*, 16:503–517, 2011.

[26] Ruhi Sarikaya, Geoffrey E Hinton, and Anoop Deoras. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784, 2014.

[27] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[28] Krzysztof Siwek and Stanisław Osowski. Data mining methods for prediction of air pollution. *International Journal of Applied Mathematics and Computer Science*, 26(2):467–478, 2016.

[29] Qing Tao, Fang Liu, Yong Li, and Denis Sidorov. Air pollution forecasting using a deep learning model based on 1d convnets and bidirectional gru. *IEEE access*, 7:76690–76698, 2019.

[30] Qing Tao, Fang Liu, Yong Li, and Denis Sidorov. Air pollution forecasting using a deep learning model based on 1d convnets and bidirectional gru. *IEEE access*, 7:76690–76698, 2019.

[31] Luciana Maria Baptista Ventura, Fellipe de Oliveira Pinto, Laiza Molezon Soares, Aderval S Luna, and Adriana Gioda. Forecast of daily pm 2.5 concentrations applying artificial neural networks and holt–winters models. *Air Quality, Atmosphere & Health*, 12:317–325, 2019.

[32] Lifeng Wu, Xiaohui Gao, Yanli Xiao, Sifeng Liu, and Yingjie Yang. Using grey holt–winters model to predict the air quality index for cities in china. *Natural Hazards*, 88:1003–1012, 2017.

[33] Thanongsak Xayasouk and Hwamin Lee. Air pollution prediction system using deep learning. *WIT Trans. Ecol. Environ*, 230:71–79, 2018.

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[35] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arxiv 2014. *arXiv preprint arXiv:1409.2329*, 2014.

[36] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

Sudhir Kumar received the B. Tech degree from the Integral University Lucknow, India, in 2015 and the M. Tech degree from the National Institute of Technology, Hamirpur, India, in 2019. He is currently pursuing the Ph.D. degree from the Indian Institute of Technology, Patna, India. His current research interests include Unmanned Aerial Vehicles, Deep Learning, the Internet of Things (IoT), 5G, and Beyond Networks.

Vaneet Kour received the B. Tech degree from the Beant College of Engineering and Technology, Gurdaspur, India, in 2016 and the M. Tech degree from the Panjab University, Chandigarh, India, in 2020. She is currently pursuing the Ph.D. degree from the Indian Institute of Technology, Patna, India. Her current research interests include Unmanned Aerial Vehicles, Deep Learning, the Internet of Things (IoT), Object Detection.

Praveen Kumar received the B. E degree from the ITM Group of Institution, Gwalior M. P, India, in 2015 and the M. Tech degree from the National Institute of Technology, Jamshedpur, India, in 2019. He is currently pursuing the Ph.D. degree from the Indian Institute of Technology, Patna, India. His current research interests include Unmanned Aerial Vehicles, Deep Learning, Reinforcement Learning, the Internet of Things (IoT), 5G, and Beyond Networks.

Anurag Deshmaner He is currently pursuing B.Tech in CSE at Indian Institute of Technology Patna. His current research interests include Unmanned Aerial Vehicles, Machine Learning, Reinforcement Learning, the Internet of Things (IoT).

Rajiv Misra is currently working as a professor in the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Patna, India. He received his M.Tech. Degree from IIT Bombay and Ph.D. from IIT Kharagpur. His research interests include Distributed Systems, Cloud Computing, Big Data Computing, Consensus in Blockchain, Cloud IoT Edge Computing, Reinforcement Learning, Unmanned Aerial Vehicles, 5G, and Beyond-5G (B5G) Networks.