TuijinJishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 2 (2023)

Time Series Analysis Based Air Pollution Level Prediction Using Long Short Term Memory Recurrent Neural Network with Attention Mechanism

K. Azhahudurai[1], Dr. V. Veeramanikandan[2]
[1]Research Scholar,Department of Computer and Information Science,
Annamalai University, Annamalai Nagar.
[2]Assistant Professor,Department of Computer Science,Thiru Kolanjiappar
Government Arts College, Vridhachalam.

Abstract

Air pollution has grown significantly as a threat to the environment during the past few decades. It has had a profound effect on the ecosystem, public health, and safety. For government agencies to design plans for pollution prevention, predicting the extent of air pollution becomes essential for maintaining the environment and environmental protection. One of the efficient approaches to solving the problem of predicting the level of air pollutants is time series analysis. The drawbacks of conventional machine learning algorithms in time series analysis are solved by employing deep learning algorithms in a variety of ways. Deep Learning models learn features and dynamics exclusively and directly from the data, in contrast to Machine Learning models, such as autoregressive models (AR) or exponential smoothing, where feature engineering is conducted manually and frequently certain parameters are adjusted while taking domain expertise into account. To forecast the level of pollution in the following hour using the current hour's weather and pollution levels, a Recurrent Neural Network (RNN) model based on Long Short Term Memory (LSTM) units has been trained, tested, and optimized. The predictive model has been developed with both uni-variate, which takes into account only one feature, and multi-variate, which takes into account many features. To improve the performance of the model on long input sequences attention mechanism has been implemented.

Keywords: Recurrent Neural Network, Long Short Term Memory (LSTM) Units, Deep Learning, Machine Learning Algorithms, Time Series Analysis, Attention Mechanism.

1. Introduction

One of the most significant public health issues is air pollution, particularly in metropolitan areas where most of industry and traffic flow. Both the environment and human health are impacted by it. Today, the environment of Earth is also in danger from air pollution [1]. The mortality rate from air pollution is a major public health issue, and by 2050, this number is predicted to have doubled [2]. The World Health Organisation (WHO) recently announced that over ninety percent of the global population are exposed to air that is hazardous and exceeds recommended limits, which is the primary cause of the high prevalence of early mortality and illness.Additionally, 4.6 million individuals pass away each year from conditions directly linked to environmental pollution [3]. Depending on the type of chemical present and the length of exposure, health issues worsen. In light of this, we may define a number of types of pollutants that fall into two categories: primary and secondary, depending on their sources. The first group of air pollutants includes those that are passed on directly from an atmospheric source. The pollutants that result from the chemical reactions of the primary pollutants in the atmosphere make up the secondary group [4]. By enabling citizens, families, businesses, and governmental bodies to prepare ahead, accurate air quality forecasts can have significant social

and economic benefits. For predicting air quality, a variety of analytical models are put forth, both conventional (based on statistical models) and non-conventional (using AI) methods.

The time-series approach is typically utilized in the field of air quality forecasting to comprehend association of cause-and-effect. In the traditional forecast approaches Auto-regressive and moving average algorithms were employed in either long term or short term prediction of air pollutants. Deep Learning has become an essential component of the present-day version of time series analysis models due to a growing amount of data and computer power in recent years, producing excellent results. In case of conventional machine learning approaches such as autoregressive models (AR) or exponential smoothing, the feature engineering is manually carried out and frequently some parameters are optimized based on the domain knowledge. The term ARIMA refers for Autoregressive (AR) and Moving Average (MA) techniques combined in the construction of a composite time series model. Despite its simplicity, this model might produce useful findings. In order to handle the embedded autocorrelation in the data, it provides parameters to take seasonality, long-term trend, autoregressive, and moving average factors into account. Similar to ARIMA models, forecasts using exponential smoothing are based on weighted averages; however, in this case, each observation is given a different diminishing weight, and as time goes on, the significance of each observation decreases. Conventional ML algorithms suffer from the fact that missing values can significantly impact how well the models perform; they are unable to identify intricate patterns in the data; and they often perform better for short-term forecasts rather than for long-term projections. The drawbacks of conventional machine learning are overcome by the use of deep learning for prediction of time series in a variety of ways.

Time series forecasting has long been a very important topic of research in many domains as numerous domain data are collected and processed as time series. The following are the components of a time series data;

- Trend - The overall trajectory of the data, as determined by excluding any short-term impacts like seasonal changes or noise, is known as the long-term trend.
- Seasonality - It describes recurring, cyclical oscillations throughout the course of a time series.
- Stationarity - Time series have the crucial quality of stationarity. If a time series' mean, variance, and covariance don't change much over time, it is considered to be stationary. There are numerous transformations that can be used to extract stationary segments from a non-stationary process.
- Noise - Each time series includes noise, which is defined as erratic variations brought on by issues in the data collection or aggregation process.
- Autocorrelation - In order to detect seasonality and trend in time series data, autocorrelation, which is the correlation among the raw time series and a time delayed version of the series, is used.

With world’s economy and industrialization expanding quickly in recent years, environmental pollution is becoming a more severe issue [6]. In particular, air pollution is important. The American Heart Association came to the conclusion that breathing in air contaminated with particulate matter (PM) increases the risk of cardiovascular morbidity and death [7]. Elevated daily mortality is especially linked to particle mass components present in the aerodynamic diameter size range below 2.5 um [9], [10], and PM2.5 is one of the primary components of haze [8]. The management of the environment and human health depend greatly on monitoring and forecasting PM2.5 concentration. Due to several complicated aspects, such as nonlinear features in both space and time, which will have a significant influence on the forecast accuracy [12], the mechanics and procedure of PM2.5 production are quite complex [11]. Therefore, a thorough analysis is crucial. This information on air quality has a clear periodicity and is tightly tied to time, making it a time series data [13]. The timely nature of the data has made time series prediction a popular subject [14].

Because of the unknowable variations in air pollution level patterns and circumstances, anticipating time-related data is a challenging challenge. In this study, we looked for the solutions that would provide the greatest results in terms of less prediction mistakes. With this objective, the LSTM-RNN deep learning model has been employed to predict the level of air pollutant.

2. Literature Survey

Researches in the area of Image analysis, speech recognition, and text processing have all benefited from the widespread usage of deep learning, which has emerged as one of the most significant approaches in the area of machine learning. Convolutional neural networks (CNN) [26], recurrent neural networks (RNN), and self-encoders networks are some of the most widely used deep learning algorithms. CNN has the benefit of feature extraction, while RNN excels in mining time series data. Various established time series prediction techniques exist, such as ARMA [15], ARIMA [15], SARIMA [16], SVR [17], Backpropagation neural network [18], Probabilistic network [19], and others [27, 28]. These approaches, however, can no longer satisfy the real need owing to an excessive length of training time as the volume and complexity of the collected data rise. The time series model enables the prediction of PM2.5 with the advancement of deep learning.

To forecast PM2.5 contamination, the authors of [20] suggested using a long short-term memory-fully connected (LSTM-FC) neural network. The feasibility and applicability of the CNN-LSTM model to forecast PM2.5 concentration were confirmed by the work in [21, 25]. In [22], a CNN-LSTM hybrid model was created to forecast the 8-h average ozone level in Beijing City for the following day. However, the requirement for forecasting PM2.5 in everyday life couldn't be satisfied by existing approaches because to their low accuracy and lengthy predictability. Meanwhile, it is critical to create an improved model for predicting PM2.5 concentration due to the complexity of PM2.5 generation, the exceptional precision and efficiency need for prediction, and the challenge of deep learning network model in stability.

In order to anticipate the PM2.5 concentration for the following day (the next 24 hours), a RNN-LSTM model is developed. Attention mechanism is incorporated into the model in order to further improve prediction accuracy. Uni-variate LSTM model and the multivariate LSTM model are constructed with single and multiple attributes, and their performance were compared and examined to see which model is the performing better for PM2.5 prediction. Finally, mean absolute error (MAE) and root mean square error (RMSE), two performance indicators, are used to assess the models.

3. Overview of Lstm Rnn Architecture

Deep Learning algorithms learn features and patterns exclusively from the temporal sequence data and they are capable of learning more complex patterns. Each input in a gated recurrent unit, long short-term memory, or recurrent neural network correlates to an output generated at the same time step. Although there isn't always a link between each input and each output, in many real-world situations it is requiredto output a sequence given an input series of varying length. The sequence-to-sequence mapping models were addressed by the Encoder-Decoder model for Recurrent Neural Networks. An encoder-decoder receives a series of data as input and outputs the most likely following sequence. Two sub-models make up the model; the decoder, which is in charge of moving through the output time steps while reading from the context vector, and the encoder, which is responsible for encoding the entire sequence into a fixed length vector called a context vector.

The encoder is made up of a stack composed of multiple recurrent units, which can be GRU cells, LSTM cells, or basic RNNs. Each unit receives a single input sequence element, gathers data from that element, and propagates it to subsequent elements. The function of the selected recurrent unit is used to calculate the hidden state vector $h(t)$. The input vector $x(t)$and the prior hidden state $h(t-1)$ are both subjected to the function with the proper weights:

$$h(t) = f(Wh(t-1) + Ux(t)) \quad \text{Eq. 1}$$

$h(t)$ encompasses all essential information encoded from the previous inputs and hidden representation. The context vector, which serves as the starting hidden state for the decoder, is the resultant hidden state obtained from the encoder of the model. In order to aid the decoder in producing precise forecasts, it incorporates the information for all input segments.The decoder is made up of an array made up of recurrent units. A hidden state $s(t-1)$ generated by the preceding unit is accepted by each recurrent unit, which then generates an output $y(t)$

and a hidden state s(t) of its own.The function of the selected recurrent unit is used to calculate the hidden state s(t):

$$s(t) = f(Wh(t-1)) \quad \text{Eq. 2}$$

A probability vector is produced by computing the output y(t) of the softmax function utilising the hidden state at the present time step s(t) and the appropriate weight:

$$\hat{y}(t) = \text{softmax}(Vs(t)) \quad \text{Eq. 3}$$

Since the inputs and outputs have no association and their lengths might vary, this model's strength resides in its ability to map sequences of various lengths to one another. As a result, an entirely new set of issues can now be resolved utilizing such architecture which is shown in Fig. 1.When the sequence length increases, it becomes very difficult to condense a long sequence into one vector, and the model frequently forgets the early parts of the input sequence when analyzing the latter parts. This strategy works well for short sequences. This explains why numerous investigations demonstrate that as sequence size increases, this model performs worse.

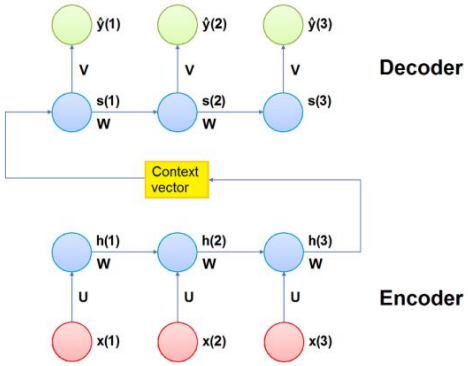

Figure 1 Schematic view of Encoder-Decoder Architecture [5]

3.1 Attention Mechanism

The Encoder-Decoder Model was developed to perform better on extended input sequences, and the Attention Mechanism is one of the fundamental frontiers in Deep Learning. The fundamental goal is to enable the decoder to access encoder data selectively when decoding. This is accomplished by creating a unique context vector for each time step of the decoder, computing it in function of the most recent hidden state and of all the hidden states of the encoder, and assigning trainable weights to each of them (shown in Fig. 2).

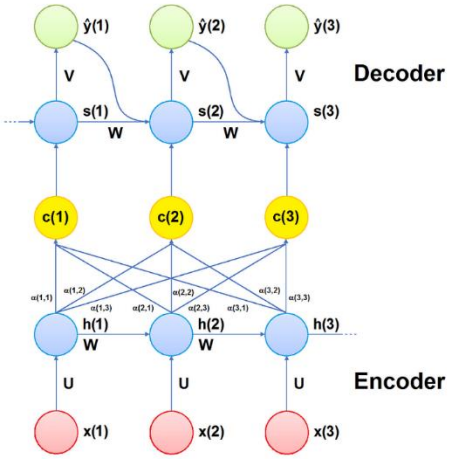

Figure 2 Schematic view of Encoder-Decoder Architecture with Attention [5]

In this way, the attention mechanism provides the various input sequence components varying degrees of priority while paying closer attention to the inputs that are more pertinent.The fundamental distinction between the Attention mechanism and the Encoder-Decoder model is the computation of a unique context vector $c(t)$ for each decoder time step t.

These steps were followed to determine the context vector $c(t)$ for time step t. First, the so-called alignment scores $e(j,t)$ are calculated with the following weighted sum for each combination of time steps j of the encoder and t of the decoder:

$$e(j,t) = V_a \tanh\left(U_a s(t-1) + W_a h(j)\right) \quad \text{Eq. 4}$$

W_a, U_a, and V_a are trainable weights in this equation that are referred to as attention weights. The hidden states of an encoder are related with the weights W_a, those of a decoder are associated with the weights U_a, and the function used for calculating the alignment score is defined by the weights V_a.The scores $e(j,t)$ are normalised using the softmax function over the encoder time steps j to provide the attention weights (j,t) for each time step t:

$$\alpha(j,t) = \frac{\exp(e(j,t))}{\sum_{j=1}^{M} \exp(e(j,t))} \quad \text{Eq. 5}$$

The relevance of the input from time step j for decoding the output from time step t is captured by the attention weight (j,t). The weighted total of all the encoder's hidden values derived using the attention weights is the context vector, or $c(t)$:

$$c(t) = \sum_{j=1}^{T} \alpha(j,t) h(j) \quad \text{Eq. 6}$$

By using this context vector, it is possible to focus more on the inputs in the input sequence that are more pertinent.

3.1.1 Decoder

The decoder is now given the context vector c(t), and it determines the probability distribution of the following output. Each time step in the input is subject to this decoding operation. Then, using the context vector $c(t)$, the hidden state $s(t-1)$ and output $y(t-1)$ from the previous time step as inputs, the recurrent unit function is used to compute the current hidden state $s(t)$:

$$s(t) = f\left(s(t-1), \hat{y}(t-1), c(t)\right) \quad \text{Eq. 7}$$

As a result, the model may determine correlations between various input sequence components and associated output sequence components utilizing this approach.The output of the decoder is determined for each time step using the weighted hidden state and the softmax function:

$$\hat{y}(t) = \text{softmax}(V_s(t)) \quad \text{Eq. 8}$$

Even in the midst of lengthy input sequences, the attention mechanism produces positive consequences. The advantage of the attention mechanism is that it is more interpretablethan other Deep Learning models. Additionally, the attention mechanism will produce excellent results in time series analysis since it enables recall of every element in the input sequence and the ability to identify the elements that are most important when constructing a response.

4. Methodology

The overall process of constructing a predictive model for time series data is presented in Fig. 4. Before building a model the data samples are preprocessed to handle missing values and label values of combine wind direction attribute. The missing values are imputed using a linear interpolation technique. By joining dots along a straight line in ascending order, linear interpolation refers to guessing a missing value. In the same ascending sequence as the preceding values, it guesses the unknown value. The most common method used in numerous researches is interpolation when imputing missing values with the average. When working with time-series data, interpolation is frequently employed as missing values are to be filled with preceding one or two values. The z-score method is used to standardize the data. It uses z-score normalization and min-max normalization, the two most used scale normalizing techniques. The most common and accurate way to represent the original raw data is by Z-score normalization when the data fits into a normal distribution. Each data point is converted into a positive or negative value that indicates how many standard deviations away from the mean it is as part of the Z-score normalization procedure.

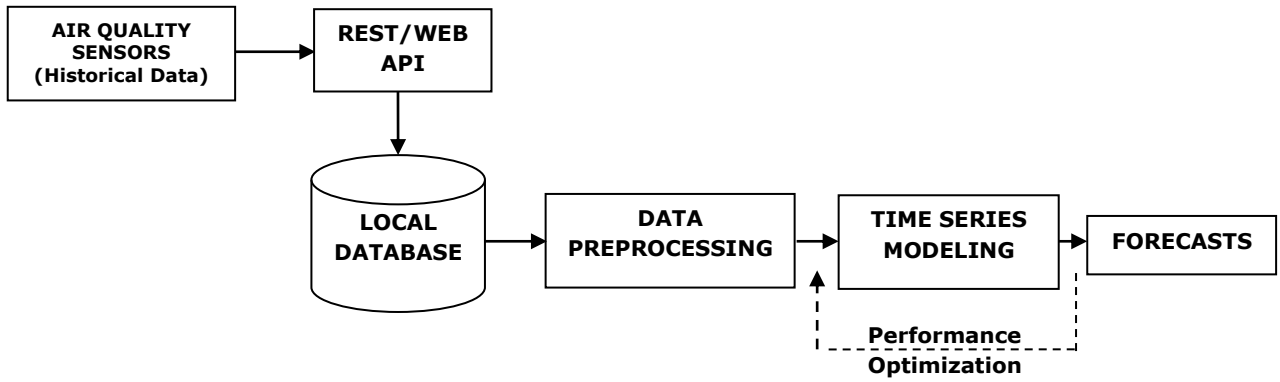

Figure 4 Block Diagram of time series analysis process

Algorithm: Model Training and Testing LSTM

Input: Data (PM2.5), Features (Univariate/ Multivariate)

Output: Evaluation metrics of the predicted data

Data split : 75% training + validation and 25% test data

#Normalization

1: object1 ← Z-scoreScaler()

2: object2 ← Z-scoreScaler()

3: Data2 ← object1.fittransform(Data)

4: Features2 ← object2.fit transform(Features)

#Train and Validation

5: count ← length(Data2) ∗ 0.75

6: X ← Data2(0 : count)

7: V ← X ∗ 0.20

8: X1 ← X − V

9: Z ← Features2(0 : count)

10: v ← Z ∗ 0.20

11: Z1 ← Z − v

#Data for Evaluation

12: x ← Data(count :)

13: z ← Features(count :)

#Model_Hyperparameters

14: epochs ← [100,200,300, 400, 500]

15: activation ← [tanh, relu]

#Training LSTM

16: for each i in activation do

17: model ← Sequential()

18: model.add(LSTM(64), activation = i)

19: model.add(Dense(1))

20: model.compile(loss = mae, optimizer = adam)

21: for each j in epochs do

22: model.fit(Z1, X1, validation = (v, V), epochs = j)

23: forecast ← model.predict(z)

24: forecast ← object1.inverse transform(forecast)

25: return rmse, mae, mape, rrse

26: end for

27: end for

To check for underfitting and overfitting, the LSTM RNN with and without attention was trained independently for various numbers of epochs. The 'tanh' and 'relu' activation function for LSTM was used to test for the nonlinearity of 'relu' and to enhance the model. It caused the construction of two loops, adding up to 10 different LSTM models. The 'adam' optimizer was used to train the model. In place of the conventional stochastic gradient descent method, Adam is an optimization technique that may be used to iteratively update network weights depending on training data. Adam additionally uses the average of the second moments of the gradients, as opposed to adjusting the parameter learning rates based on the average first moment as in RMSProp.

Table. 1 Model Layer and Hyperparameter Details

Hyperparameter	Value
No. of LSTM layers	02
No of LSTM Units	32 and 64
Size of Fully connected layer	32
No of Epochs	100, 200, 300,400, and 500
Activation Function	Tanh and ReLu
Optimizer	ADAM

Learning Rate	0.001
Dropout rate	0.15
Momentum	0.9
Batch size	64

This research utilizes a model with two LSTM layer each consisting of 32 and 64 LSTM units respectively followed by a fully connected layer for prediction. Model prediction performance was evaluated using the RMSE, MAPE, and MAE scores. The mean of absolute error (MAE) is known as MAE. The fact that MAE estimates mistakes on a comparable scale suggests that it considers both large and small errors equally. To properly investigate the forecasts, it is not enough. MAPE Mean Absolute Percentage Error, a type of normalized absolute error, enables comparison of mistakes across data sets with different sizes. MAPE processes and averages each piece of data, which helps it identify more errors and outliers. Additionally, since the estimation of the actual data will be less than the anticipated data, it is beneficial to penalize negative mistakes. However, because the errors are squared, the RMSE Root Mean Square Error is useful for punishing greater mistakes because larger values are often given more weight.

Dataset Description

Beijing Multi-Site AirQuality, obtained from the UCI Machine Learning Repository, was the dataset utilized in this study [23]. It includes hourly data on air pollution collected from air-quality monitoring locations (stations) between January 1st, 2010, and December 31st, 2014. There are 7 climatic and meteorological features in it. Particulate matter (PM2.5) pollution is the key predictor variable. As an example the trend and seasonality in the time series data is shown in Fig. 3 as series plot. The different important attributes present in the dataset are listed below;

- Dew Point in in °C Td
- Temperature in °C
- Pressure in kPa
- Combined wind direction – categorical value
- Cumulated wind speed in km/hr
- Cumulated hours of snow in hrs
- Cumulated hours of rain in hrs

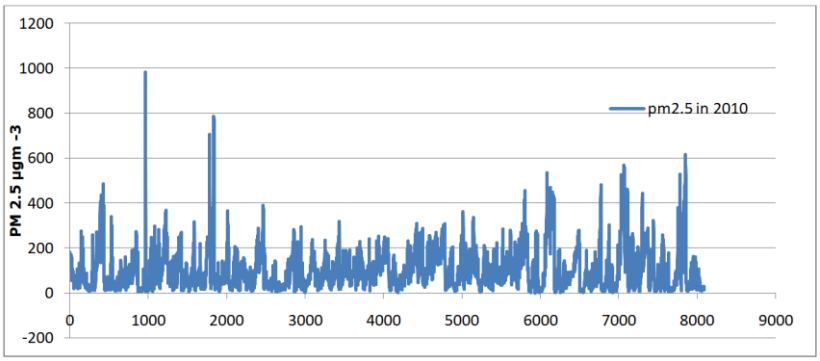

Figure 3 Seasonality and Trend in PM 2.5 data 2010

5. Experiments and Results

In this experiment, LSTM model was assessed under different scenario with and without attention mechanism; training with single attribute and multiple attribute to evaluate the performance of the models for the time series prediction task. Dew, temperature, atmospheric pressure, wind direction, wind speed, snow, and rainfall are the

different attributes used in the prediction of air pollution (PM 2.5) level. The Fig. 5 presents the correlation of different attributes used for building the prediction model. The plot shows that there is a higher level of correlation between the PM 2.5 level and dew point. The other attributes has lower correlation with the level of air pollutant. Hence during the model training in univariate mode, dew point is used and in multi variate model all the attributes are used.

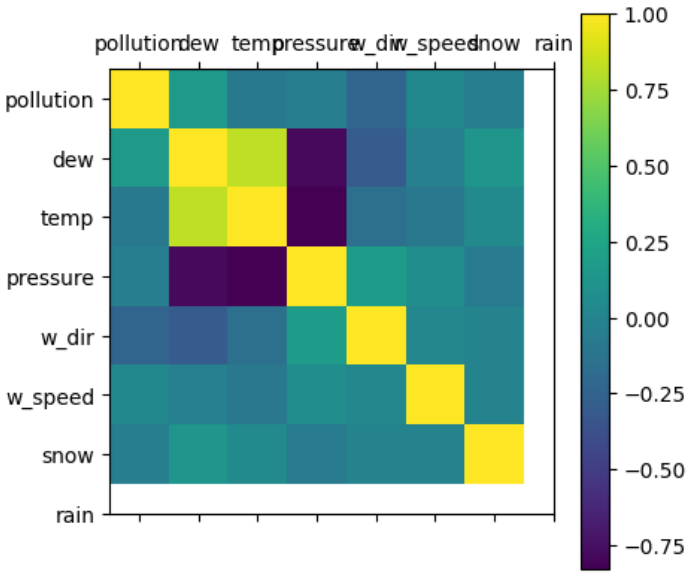

Figure 5 Correlation plot of attributes used in prediction

The deep learning based model when trained with attention mechanism outperformed the other models in terms of mean absolute percentage error. At the same time, LSTM model trained in univariate mode has the highest error in all assessment metrics compared with other models. The MAPE value is highest for LSTM univariate model, which is 34.98. Though there is a lower correlation levels between the PM 2.5 and other attributes, the model showed better performance when trained with multiple attributes. It shows that the attention mechanism can, in certain cases, increase the prediction accuracy by adaptively choosing the most pertinent input features. The model performance under different scenario is presented in Table 2.

Table. 2 Performance metrics values of model under different scenario

Model	RMSE	MAE	MAPE
LSTM Univariate	21.02	14.56	34.98
LSTM Multivariate	19.16	12.53	32.56
LSTM Univariate + Attention	19.85	13.12	33.08
LSTM Multivariate + Attention	17.65	10.64	30.45

Even though it has been demonstrated that LSTM with attention produces the best outcomes, it is crucial to understand which activation function was more effective at predicting. Table 3 lists the average outcomes for the 'tanh' and'relu' activation functions that the LSTM employed to anticipate PM2.5 values across all scenarios. The findings unambiguously show that LSTM performance increased with relu's nonlinearity [24].

Table. 3 Performance metrics values of model under different activation function

Model	RMSE	MAE	MAPE
Tanh	22.02	15.16	28.54
ReLu	20.18	12.65	24.25

The graph presented in the Fig. 6 shows the correlation between the actual and the predicted air pollution levels using a LSTM model with attention mechanism. The model is capable of capturing the trend and seasonality pattern in the time series data with predicted values slightly deviating from the actual values of the air pollutant.

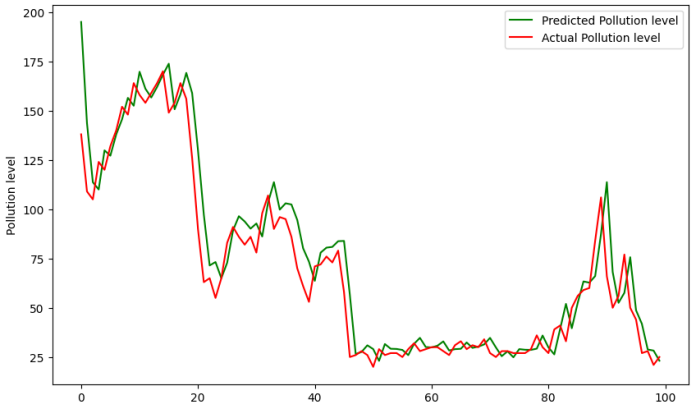

Figure 6 Result of prediction using LSTM with Attention Mechanism

6. Conclusion

The most widely used Deep Learning method for time series forecasting is recurrent neural networks since it enables accurate predictions on time series in a variety of scenarios. When used on lengthy sequences, RNNs have the vanishing gradient issue, which is their biggest drawback. The vanishing gradient issue with RNNs was addressed in this work using LSTM, which uses gates to control the information flow across the sequence chain. Using a separate context vector for each time step, the Attention mechanism is an extension of the Encoder-Decoder model that was used to address the Encoder-Decoder model's performance degradation in the presence of extended sequences. The features that are useful for predicting are selected using the attention weights on rows. It captures temporal information since the context vector is now a weighted total of the row vectors that hold the data over several time steps. This makes it easier to choose the relevant features from among the multivariate time series data to predict the respective target time series feature, PM 2.5 level. The techniques used produced positive outcomes. It does, however, limit our analysis to the model's sufficiency, which may be further enhanced by taking into account an ensemble of other forecast models. By taking feature engineering and better hyperparameter optimization into consideration, which will be a part of the future study, the obtained prediction results can be improved.

References

1. Kahraman, C. (2009) Risk Analysis and Crisis Response. Stochastic Environmental Research and Risk Assessment, 23, 413-414.
2. Hamanaka, R.B. and Mutlu, G.M. (2018) Particulate Matter Air Pollution: Effects on the Cardiovascular System. Frontiers in endocrinology, 9, Article 680.
3. World Health Organization (2021) Ambient (Outdoor) Air Pollution.https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health
4. Lyon, F. (1994) IARC Monographs on the Evaluation of Carcinogenic Risks to Humans. Some Industrial Chemicals, 60, 389-433.

5. https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc.
6. J. H. Seinfeld, ‘‘Air pollution: A half century of progress,’’ AIChE J., vol. 50, pp. 1096–1108, Jun. 2004.
7. R. D. Brook, S. Rajagopalan, C. A. Pope, J. R. Brook, A. Bhatnagar, A. V. Diez-Roux, F. Holguin, Y. Hong, R. V. Luepker, M. A. Mittleman, A. Peters, D. Siscovick, S. C. Smith, L. Whitsel, and J. D. Kaufman, ‘‘Particulate matter air pollution and cardiovascular disease: An update to the scientific statement from the american heart association,’’ Circulation, vol. 121, no. 21, pp. 2331–2378, Jun. 2010.
8. R.-J. Huang et al., ‘‘High secondary aerosol contribution to particulate pollution during haze events in China,’’ Nature, vol. 514, no. 7521, pp. 218–222, Oct. 2014.
9. J. Schwartz, D. W. Dockery, and L. M. Neas, ‘‘Is daily mortality associated specifically with fine particles?’’ J. Air Waste Manage. Assoc., vol. 46, no. 10, pp. 927–939, Oct. 1996.
10. J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer, ‘‘The contribution of outdoor air pollution sources to premature mortality on a global scale,’’ Nature, vol. 525, no. 7569, pp. 367–371, Sep. 2015.
11. J. L. Wang, ‘‘Chemical composition and quantitative relationship between meteorological condition and fine particles in Beijing,’’ J. Environ. Sci., vol. 16, no. 5, pp. 860–864, 2004.
12. C. K. Chan and X. Yao, ‘‘Air pollution in mega cities in China,’’ Atmos. Environ., vol. 42, no. 1, pp. 1–42, Jan. 2008.
13. N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, ‘‘The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,’’ Proc. Roy. Soc. London A, Math., Phys. Eng. Sci., vol. 454, no. 1971, pp. 903–995, Mar. 1998.
14. Garima and S. Rani, ‘‘Review on time series databases and recent research trends in time series mining,’’ in Proc. 5th Int. Conf. Confluence Next Gener. Inf. Technol. Summit (Confluence), Sep. 2014, pp. 109–115.
15. G. E. Box and G. M. Jenkins, ‘‘Time series analysis forecasting and control,’’ J. Time, vol. 31, no. 4, pp. 238–242, 1976.
16. B. M. Williams, P. K. Durvasula, and D. E. Brown, ‘‘Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models,’’ Transp. Res. Rec., vol. 1644, no. 1, pp. 132–141, Jan. 1998.
17. A. J. Smola and B. Schölkopf, ‘‘A tutorial on support vector regression,’’ Statist. Comput., vol. 14, no. 3, pp. 199–222, Aug. 2004.
18. C. Haipeng, S. Ping, and H. Shengguo, ‘‘Study of aircraft hard landing diagnosis based on nerual network,’’ Comput. Meas. Control, vol. 16, no. 7, pp. 906–908, 2008.
19. S. Sun, C. Zhang, and G. Yu, ‘‘A Bayesian network approach to traffic flow forecasting,’’ IEEE Trans. Intell. Transp. Syst., vol. 7, no. 1, pp. 124–132, Mar. 2006.
20. J. Zhao, F. Deng, Y. Cai, and J. Chen, ‘‘Long short-term memory–fully connected (LSTM-FC) neural network for PM2.5 concentration prediction,’’ Chemosphere, vol. 220, pp. 486–492, Apr. 2019.
21. C.-J. Huang and P.-H. Kuo, ‘‘A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities,’’ Sensors, vol. 18, no. 7, p. 2220, Jul. 2018.
22. U. Pak, C. Kim, U. Ryu, K. Sok, and S. Pak, ‘‘A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction,’’ Air Qual., Atmos. Health, vol. 11, no. 8, pp. 883–895, Oct. 2018.
23. Zhang, Shuyi, et al. ”Cautionary tales on air-quality improvement in Beijing.” Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473.2205 (2017): 20170457.
24. Sachin S. Talathi and Aniket Vartak. Improving performance of recurrent neural network with relu nonlinearity, 2015; arXiv:1511.03771.
25. V. Veeramanikandan and M. Jeyakarthic, "A Futuristic Framework for Financial Credit Score Prediction System using PSO based Feature Selection with Random Tree Data Classification Model," 2019

International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 826-831.

26. Veeramanikandan, Varadharajan & Jeyakarthic, M.. (2020). Parameter-Tuned Deep Learning Model for Credit Risk Assessment and Scoring Applications. Recent Advances in Computer Science and Communications. 13.
27. Veeramanikandan, V., and M. Jeyakarthic. "An ensemble model of outlier detection with random tree data classification for financial credit scoring prediction system." International Journal of Recent Technology and Engineering (IJRTE) 8.3 (2019): 2277-3878.
28. Veeramanikandan, V., and M. Jeyakarthic. "Forecasting of Commodity Future Index using a Hybrid Regression Model based on Support Vector Machine and Grey Wolf Optimization Algorithm." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 10.10 (2019): 2278-3075.