

Time Series Approximating of Air Contaminants using Joint CNN-LSTM RNN Neural Architectures

Lalitha D

Assistant Professor, Department Computer Science, IT, AI
& ML, Srinivasan College of Arts & Science, Perambalur –
621212, Tamilnadu, India

Kalaivani S

Assistant Professor, Department Computer Science, IT, AI
& ML, Srinivasan College of Arts & Science, Perambalur –
621212, Tamilnadu, India,

ABSTRACT - One of the largest concerns facing the world today is air pollution, which can aggravate pre-existing medical illnesses, lead to mental health difficulties, and cause lung and respiratory ailments as well as cardiovascular problems. It can also lead to a decline in the health of the earth. Thus, it becomes imperative to lessen and raise public awareness of these issues brought on by air pollution. It is simpler to control and reduce the dangers associated with air pollution and to guarantee a safe level of pollutant concentration in the area when an accurate approach of air pollution forecasting is used. It also aids in evaluating the threats that low air quality standards pose to the ecology and climate. Planning daily activities, avoiding high alert zones, and putting into practice efficient pollution control measures can all be made easier with accurate forecasting. Data related to the weather or meteorology has time series properties. A convolutional neural network (CNN) and long short-term memory (LSTM) combination forecasting technique for PM 2.5 is presented. The LSTM model is based on machine learning and offers the advantage of analyzing relationships between time series data using its memory function. The CNN-LSTM can offer a dependable forecast with the maximum prediction accuracy. Additionally, this forecasting technique offers a fresh avenue for future research for other forecasting applications.

Keywords: pollution forecasting, convolutional neural network, long short-term memory, PM 2.5, time series data

1. INTRODUCTION

Air is a basic necessity for all life on Earth to survive and grow. It has an impact on health and shapes how the economy develops. Today's air quality is deteriorating with more severe air pollution because of the growth of industrialization, the rise in the number of private automobiles, and the combustion of fossil fuels. Numerous pollutants, including SO₂, NO₂, CO₂, NO, CO, NO_x, PM_{2.5}, and PM₁₀, are present in the atmosphere. Researchers have undertaken a substantial amount of relevant research in response to the growing severity of environmental pollution issues, and air pollution forecasting has played a critical role in those studies. Research shows that there are three main categories into which air pollution forecasting techniques can be roughly classified: statistical, artificial intelligence, and numerical forecasting techniques. Recently, a few hybrid models have been put forth that have the potential to increase forecast accuracy. Granular materials in the atmosphere, whether solid or liquid, make up atmospheric particulate matter. Particulate matter (PM) and thick haze are common in most places across the world [1]. Medical study indicates that PM damages human DNA, the immunological system, the central nervous system, the respiratory system, and the cardiovascular system to varying degrees.

The foundation for forecasting urban air pollution is the development of a realistic and accurate forecasting model. In the science of big data, forecasting is essential and can be used to predict an object's future evolution based on historical data. Therefore, "pollution forecasting" can be defined as the estimation of the concentration of pollutants at a future date. By giving an early warning of dangerous air pollution, air quality forecasting effectively protects public health. Meteorological components can forecast urban air pollution episodes to give an early warning. The average level of air quality in India during November, 2023 is presented in the Fig. 1 [2]. Majority of the metropolitan cities showed high level of pollution due to industrial and vehicle effluents.

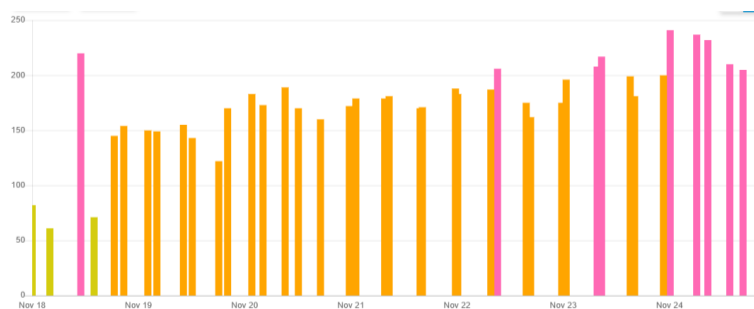


Fig.1. AQI level in India (third week of Nov, 2023)

Given the increasing frequency of urban air pollution incidents, air pollution forecasts must to incorporate emergency warnings as a crucial component of the whole emergency system, alongside risk avoidance management and emergency interventions. Based on the forecast of meteorological factors, the air pollution early warning system is activated before to the occurrence of substantial urban air pollution. As soon as practicable, corresponding emergency measures are put in place to lessen pollution output and lessen its effects. This paper examines the theory and use of such forecasting models to offer a clear viewpoint on air pollution forecasting. The benefits and drawbacks of various forecasting techniques are also discussed, based on a comparison of the approaches. The purpose of this study is to give researchers a convenient summary of air pollution forecasting techniques that they may use for future research.

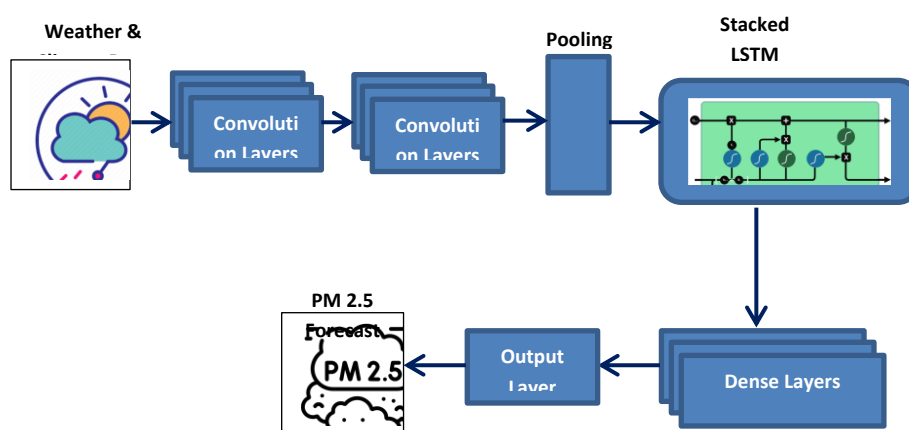


Fig. 2 Schematic view of CNN-LSTM based forecasting

Fig. 2 illustrates a hybrid CNN-LSTM network that is suggested for time series analysis. By integrating a CNN with a vanilla LSTM, it is possible to automatically create feature representations that are reliable and accurate. This is achieved by maintaining the stochastic and deterministic trends that are effectively encoded in sequence data and eliminating superfluous and irrelevant variables.

This study offers a CNN-LSTM based approach [3, 14] for forecasting AQI level of the next week in order to exploit the time series characteristics of meteorological data, deeply mine the data features, and increase the forecasting accuracy. This approach can significantly increase forecasting accuracy by combining the benefits of long short-term memory (LSTM) [4, 15], which can automatically identify the optimal mode appropriate for pertinent data, and convolutional neural networks (CNN) [5, 16], which can extract useful features from the data. Additionally, LSTM can identify the interdependence of data in time series data.

2. LITERATURE SURVEY

Currently, air quality is forecasted using two primary methods: the classical analysis approach and the machine learning method. The weather and climate data is a noisy, nonparametric dynamic system. The analysis of complicated, high-dimensional, and noisy

time series data is not appropriate for the conventional econometric techniques or equations with parameters. Mohamed Azharudheen, Kumudham, and Kalaivani [1] introduce a scalable deep learning–optimized data security architecture designed to support high availability in big data environments. Their work addresses the challenges of integrating advanced security mechanisms into large-scale data pipelines without degrading system performance. In their study on privacy-preserving big data applications, Mohamed Azharudheen and Vijayalakshmi [2] explore improved methods for securing large-scale analytical workflows. They emphasize that traditional privacy mechanisms struggle to operate efficiently under high-volume, high-velocity data conditions. Extending their research on privacy-focused data protection, Mohamed Azharudheen and Vijayalakshmi [3] develop a novel mechanism aimed at maximizing data availability while preserving privacy in cloud ecosystems. The study highlights weaknesses in classical encryption models, especially regarding computational overhead and key management vulnerabilities. Their proposed approach incorporates dynamic shuffling, key-splitting, and lightweight security layers to minimize reconstruction risks and strengthen confidentiality. This work contributes to ongoing efforts to design efficient, privacy-first architectures suitable for demanding cloud and IoT settings.

It is challenging to enhance the accuracy of the conventional time series analysis method. Time series data exhibit random walk properties [6]. Some researchers use time series linear forecasting models, such as vector autoregression (VAR) [7], Bayesian vector autoregression (BVAR) model [8], autoregressive integrated moving average mode (ARIMA) [9], and generalised autoregressive conditional heteroskedasticity model (GARCH) [10], based on statistics and probability theory to predict the short-term stock price with a large number of long-term data. However, the uncertainty and high noise characteristics of financial time series cast doubt on the accuracy of using time series models alone, and the relationship between independent and dependent variables is subject to dynamic changes over time, which restricts their further application and expansion [11, 12].

A novel deep learning method (CNN-LSTM) is suggested to forecast the AQI level by examining the time series and correlation of stock price data analysis. This method uses LSTM for data forecasting and CNN for extracting the temporal characteristic from the data. It can fully utilize the data's temporal sequence to produce forecasts that are more accurate. The assessment indices of CNN-LSTM, CNN, RNN, LSTM, and CNN-RNN are compared, and it is demonstrated that CNN-LSTM has higher forecasting accuracy and is a better choice for predicting.

3. PROPOSED CNN-LSTM MODEL

1D-CNN model was used to extract the features from the given input sequences. A 1D convolutional neural network (CNN) is designed to effectively extract the features of the sequence data. The number of CNN layers, the number of neurons in each layer, the size of the filter, and the subsampling factor of each layer are the hyper-parameters used to configure 1D-CNN. The fundamental way that a filter is applied to an input is through the convolution layer. Repeatedly applying the filtering procedure results in a feature map that shows the specific characteristics associated with the data points. Convolution is a linear procedure in which the inputs are multiplied with a set of weights. In this instance, the single-dimensional array weights—also referred to as the kernel—are multiplied by the inputs. Each iteration of training yields a different value and the resultant are referred to as a feature map. Each value is sent to the ReLU activation function once the feature map has been calculated. ReLU is a linear activation function that, in the case that the input is negative, converts it to zero and, in the other case, outputs the original input. The vanishing gradient issue is resolved and the model can learn from the training data more quickly thanks to the ReLU activation function.

When handling 1D signals, 1D CNNs are better than their 2D counterparts because of their advantages. The number of CNN and dense layers/neurons, filter (kernel) size in each CNN layer, subsampling factor in each CNN layer, and selection of pooling and activation functions comprise the configuration of a 1D-CNN. The processes of understanding and feature extraction are combined into a single process that may be tweaked to optimize predicting performance. The main benefit of 1D CNNs is that they can also have low computational complexity because the sole expensive operation is the series of 1D convolutions, which are essentially the linear weighted summing of two 1D arrays. Effectively, such a linear process can be carried out in tandem with the Forward and Back-Propagation operations.

In addition, the extracted features are then processed by the LSTM layers in order to further extract the temporal features. After that, the output features are fed into several fully connected layers. Proposed Approach consists of two networks. First, a 1D-convolutional neural network pre-processes the data and then sends the compact representation to the stacked LSTM network. The output of stacked LSTM is given to an activation function that gives the one week ahead forecast for the given input sequence. CNN is employed in feature engineering because of its propensity to focus on the most noticeable elements in the field of view. LSTM is a popular technique for time series because of its ability to extract essential information from the temporal sequence. The effectiveness of model learning can be increased by significantly reducing the number of parameters through the use of CNN's local perception and weight sharing.

Eq. 1

$$I_t = \tanh(x_t * k_t + b_t)$$

where I_t output of the convolution, \tanh is the activation function, x_t is the input vector, and k_t is the convolutional kernel weight.

3.1 Long-Short Term Memory RNN

Long Short-Term Memory (LSTM) networks (shown in Fig. 3) which enable higher memory retention for prior information, are created by adjusting recurrent neural networks. Here, the vanishing gradient problem of the RNN is resolved. Applications of LSTM made it possible to process, and predict time series which has long term dependencies. Back-propagation is used to train the model. An LSTM network has three gates. Input gate identifies the value from the input should be utilized to change the memory using the input gate. The sigmoid function determines which numbers to pass 0,1. and the \tanh function assigns weights to the given numbers, determining their level of significance from -1 to 1.

Eq. 2

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

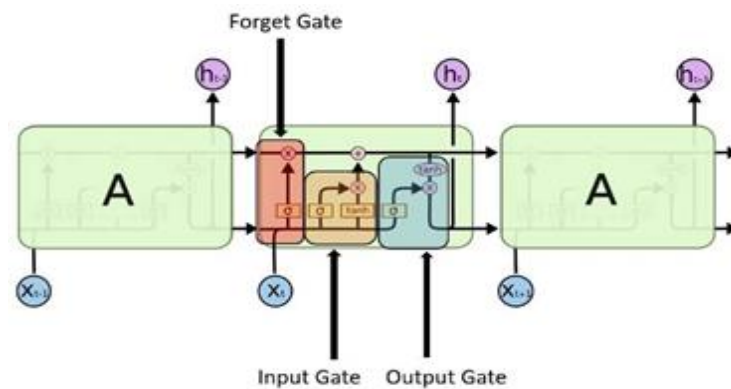


Fig. 3 Architecture of LSTM Unit

The details to be removed from the block are identified by the forget gate. The sigmoid function makes that determination. For each number in the cell state C_{t-1} , it produces a number between 0 and 1 by examining the previous state (h_{t-1}) and the content input (x_t).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{Eq. 3}$$

The block's memory and input are used to determine the output. The sigmoid function determines which numbers to pass 0,1. and the \tanh function multiplies the Sigmoid output with the given values to determine the relevance of each item on a scale from -1 to 1.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad \text{Eq. 4}$$

$$h_t = o_t * \tanh(C_t)$$

RNN adds new information by completely changing the pre-existing information and applying a function. The entirety of the information is transformed as a result of "vital" information and "not so important" information not being taken into account. LSTMs, on the other hand, only add and multiply the data insignificantly. With LSTMs, the data is transmitted through a method known as cell states. This selective recall or selective forgetting capability of LSTMs is useful when processing a time series data with long term dependency.

4. EXPERIMENT AND RESULTS

In the experiments a 1D-CNN with 02 convolution layers, 01 maximum pooling layer, stacked LSTM layers, and 03 fully connected layers were utilized. Sets of convolutional and pooling layers followed by stacked LSTM layers and fully connected layers make a different architecture when compared to the traditional design of a 1D CNN. Each convolutional layer employs a small 1D filter to capture the local feature of the input data. For non-linear data transformation, the output of convolutions is passed via an activation function. To create a hierarchical representation of the data, pooling is used. The architecture's configurable parameters are all learned using a feed-forward and back-propagation technique that minimizes a cost function.

The experiments were performed on the dataset containing the PM2.5 of the five Chinese cities including the meteorological data for each of the city over the duration Jan 01, 2010 and Dec 31, 2015. Dew point, temperature, humidity, atmospheric pressure, cumulated precipitation, and wind speed were used for predicting the value of the PM2.5 [13].

Table 1. Dataset Description

Item	Description
Characteristics	Time Series and Multivariate
Number of samples	43824
Number of Attributes	06 (weather and atmospheric parameters)
Period of Collection	2010 – 2015
Frequency	Hourly Data

When processing the raw input data, the convolution layers provide a compact representation of it, which is then given to the LSTM layers for further processing. Thus, feature extraction and forecasting are merged into a single process that is carried out repeatedly until the forecasting is observed to be satisfactory. The 1D convolutions may be thought of as the linear weighted sums of the 1D raw input and the 1D convolution kernel, the 1D-CNN is computationally less expensive. Both the forward propagation and the backward propagation can parallelize the linear weighted sum process. The difference between the expected and actual output is utilized to calculate the model's prediction error. Utilizing the mean square error function and the ADAM optimizer, the CNN parameters are optimized. With a learning rate of 0.0001, the model's parameters are tuned by reducing the error. With a mini-batch size of 16, the CNN was trained for a maximum of 150 epochs.

Table 2 Hyperparameter Values of 1D-CNN

1D-CNN Parameter	Value
Loss Function	MSE
Optimizer	ADAM
Learning Rate	0.0001
Batch Size	16
Epochs	150

Table 3 Filter and Kernel Details

Layer	Value
No of Filters in Convolution 1	32
No of Filters in Convolution 2	64
Max Pooling size	02
Dense Layer 1 No of Neuron	64
Dense Layer 2 No of Neuron	32
Dense Layer 3 No of Neuron	16
Output Layer No of Neuron	01
No. of Trainable Parameters	209665

Given the significant gap in the input data, the z-score standardization approach is used to standardize the data in order to improve model training. This can be seen in the following formula:

$$y_i = \frac{x_i - \bar{x}}{s} \quad \text{Eq. 5}$$

where y_i is the standardized output, x_i is the raw input data, \bar{x} is the average of the input sequence, and s is the standard deviation of the input sequence. The error in forecasting is estimated by comparing the output value with the raw input data. During training update each layer's weight and bias,

propagate the estimated error in the opposite way, and continue training the network. The mean absolute error (MAE), root mean square error (RMSE), and R-square (R²) are used as the methodologies' evaluation criteria in order to assess the forecasting effectiveness of CNN-LSTM.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad \text{Eq. 6}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad \text{Eq. 7}$$

$$R^2 = 1 - \frac{(\sum_{i=1}^n (\hat{y}_i - y_i)^2)/n}{(\sum_{i=1}^n (\bar{y}_i - \hat{y}_i)^2)/n} \quad \text{Eq. 8}$$

where \hat{y}_i is the predicted value, y_i is the ground truth value, and \bar{y}_i is the average value. The value of R^2 ranges between 0, and 1. The forecasting accuracy increases with the distance between the predicted and real values from zero, as indicated by the values of RMSE and MAE. The better the model's degree of fitting, the closer R^2 is to 1.

Table 4. Comparison of proposed model's performance

Model	MAE	RMSE	R ²
MLP	26.485	37.457	0.89
CNN	19.985	31.574	0.91
RNN	27.624	30.851	0.93
LSTM	25.862	28.458	0.94
CNN-RNN	25.416	26.349	0.96
CNN-LSTM	24.438	24.869	0.98

Following the training of MLP, CNN, RNN, and LSTM utilizing the processed training set data the model finished by training is used to forecast the test set data for CNN-RNN and CNN-LSTM, respectively. The real value is compared with the predicted value to estimate performance metrics (tabulated in Table 3). MLP has the lowest degree of broken line fitting, whereas CNN-LSTM has the highest degree, nearly corresponding with each other. After the CNN layer, the CNN-LSTM suggested in this work has lower MAE and RMSE than LSTM; R² has somewhat improved; MAE lowers by 4.0%, RMSE decreases by 3.2%, and R² increases by 0.2%. The accuracy of the model during the training and validation is presented in Fig. 3. It demonstrates that using CNN to extract data features can significantly enhance the predicting performance of LSTM. In this research, the CNN-LSTM model outperforms the other four comparison models in terms of error value and fitting degree. It accurately forecasts the AQI reading for the upcoming week.

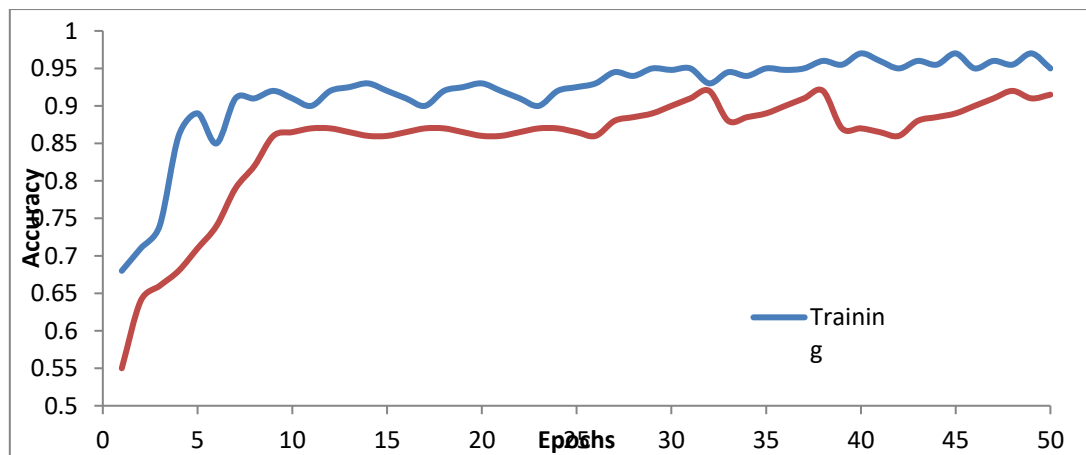


Fig.4 Accuracy plot (Training and validation)

5.CONCLUSION

This research suggests a CNN-LSTM to predict the AQI level based on the temporal features of weather and climate data. The approach fully utilizes the time sequence aspects of the data by using meteorological and climatic data, as well as changes in the data, as input. CNN calculations are done in parallel hence they are supposed to be quicker than LSTMs, which require sequential processing because each step depends on the preceding one. By concentrating on the most crucial features, the CNN network positioned in the first layer of the suggested model can result in a number of complexity reductions. The tensor's size is reduced by using convolutional layers, and this reduction is further achieved by using pooling layers. The CNN's output is used as the LSTM's input in this work. As a result, the CNN's learnt features from the input data can be learned by the LSTM. The CNN's input will be fed into the LSTM's output in upcoming tests. By doing this, the CNN will be able to extract features from the LSTM's output. We will also investigate the performance of the parallel design, in which the fully linked layer receives concatenated outputs from the CNN and LSTM, which process the input data independently.

6.REFERENCES

- [1] Prof.A. Mohamed Azharudheen, Mrs. A. Kumudham, & Ms. S. Kalaivani. "A Scalable Deep Learning-Optimized Data Security Architecture for High-Availability Big Data Environments." *International Journal of Engineering Research & Technology (IJERT)*, ISSN 2278-0181, Vol. 14, Issue 12, December 9, 2025. DOI: 10.17577/IJERTV14IS120127
- [2] A. Mohamed Azharudheen and Dr.V. Vijayalakshmi, "Improvement of data analysis and protection using novel privacy-preserving methods for big data application" *The Scientific Temper* Vol. 15, no. 2, pp. 2181-2189, 2024.
- [3] A. Mohamed Azharudheen and Dr.V. Vijayalakshmi, "Analyze the New Data Protection Mechanism to Maximize Data Availability without Having Compromise Data Privacy" *Educational Administration: Theory and Practice*, Vol.30. No.5, pp. 3911-3922, 2024.
- [4] Abbasimehr, Hossein, and Reza Paki. "Improving time series forecasting using LSTM and attention models." *Journal of Ambient Intelligence and Humanized Computing* (2022): 1-19.
- [5] Kirisci, Melih, and Ozge Cagcag Yolcu. "A new CNN-based model for financial time series: TAIEX and FTSE stocks forecasting." *Neural Processing Letters* 54.4 (2022): 3357-3374.
- [6] R. Qiao, "Stock prediction model based on neural network," *Operations Research and Management Science*, vol. 28, no. 10, pp. 132-140, 2019.
- [7] C. Jung and R. Boyd, "Forecasting UK stock prices," *Applied Financial Economics*, vol. 6, no. 3, pp. 279-286, 1996.
- [8] W. Bleessner and P. Liicoff, "Predicting stock returns with bayesian vector autoregressive," *Data Analysis, Machine Learning and Applications*, vol. 1, pp. 499-506, 2005.
- [9] A. Adebisi, A. Adewumi, and C. Ayo, "Stock price prediction using the ARIMA model," in *Proceedings of the 2014 UKSimAMSS 16th International Conference on Computer Modelling and Simulation*, IEEE, Cambridge, UK, March 2014.
- [10] C. Zhang, X. Cheng, and M. Wang, "An empirical research in the stock market of Shanghai by GARCH model," *Operations Research and Management Science*, vol. 4, pp. 144-146, 2005.
- [11] Q. Yang and C. Wang, "A study on forecast of global stock indices based on deep LSTM neural network," *Statistical Research*, vol. 36, no. 6, pp. 65-77, 2019.
- [12] K.-S. Moon and H. Kim, "Performance of deep learning in prediction of stock market volatility," *Economic Computation And Economic Cybernetics Studies And Research*, vol. 53, no. 2, pp. 77-92, 2019.
- [13] Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. *Proceedings of the Royal Society A*, 471, 20150257.
- [14] V. Veeramanikandan and M. Jeyakarthis, "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.
- [15] Veeramanikandan, Varadharajan & Jeyakarthis, M.. (2020). Parameter-Tuned Deep Learning Model for Credit Risk Assessment and Scoring Applications. *Recent Advances in Computer Science and Communications*. 13.
- [16] Veeramanikandan, V., and M. Jeyakarthis. "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.
- [17] Veeramanikandan, V., and M. Jeyakarthis. "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.