Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.98549453

TIME SERIES BASED AIR QUALITY FORECASTING USING AUTOREGRESSIVE RECURRENT NETWORK

Mr.K. Azhahudurai
Research Scholar,Department of Computer and Information Science,
Annamalai University, Annamalai Nagar , Email: jkmaz477@gmail.com

Dr. V. Veeramanikandan
Assistant Professor,Department of Computer Science,Thiru Kolanjiappar
Government Arts College, Vridhachalam. , Email: klmvmani@gmail.com

ABSTRACT
According to a research by the Swiss technology company IQAir, India's cities made up 12 of the 15 cities with the highest pollution levels and the country's cities had the eighth-most contaminated air overall. India's average PM2.5 concentration, weighted by population, was 53.3 micrograms per cubic metre (g/m3) in 2022 compared to the WHO's suggested yearly recommendation level of 5 g/m3. IQAir's World Air Quality Report uses PM2.5, or particulate matter with a diameter of less than 2.5 microns, as the primary air quality measure. Additionally, the average determined by IQAir exceeded the permissive level of 40 g/m3 established by India's Central Pollution Control Board. The wireless sensor networks are becoming an active research field because of their wide application. This research study shall focus on developing an air pollution forecasting approach using the DeepAR, an autoregressive recurrent network using the data collected from a wireless sensor network. DeepAR has a high prediction ability as the working principle is combined with deep learning and conventional probabilistic prediction.
Keywords: PM2.5 concentration, air quality, DeepAR, autoregressive recurrent network, wireless sensor network

1. INTRODUCTION

India's air quality is in a terrible state, as evidenced by the 5th Yearly Global Air Quality Assessment published by IQAir [1]. 12 of the 15 most polluted towns in South and Central Asia are located in India, which ranks eighth on the list of nations with the lowest air quality index. In around 60% of the Indian cities examined for the paper, annual PM2.5 levels were no less than seven times greater than recommended by the WHO. Only somewhat lower than the median PM2.5 level of 58.1 pg/m3 in 2021, 53.3 pg/m3 was recorded in 2022. The highly polluted metropolis in India was determined to be Bhiwadi in Rajasthan, with frightening PM levels of 92.7; New Delhi was determined to be the highly polluted city among the world's major cities, with PM levels of 92.6.

Asthma, cancer, lung illness, heart disease, and early death are just a few of the health disorders that are brought on or made worse by exposure to air pollution. The most vulnerable groups affected by air pollution are those who are already at risk, such as elderly people, women, children, and outdoor laborers. 89+% of all pollution-related premature deaths happen in nations with low and medium incomes. Poor air quality is blamed in the study for 93 billion sick days and more than six million deaths globally each year. Over 1.2 million deaths are attributed to air pollution in India each year, according to the World Health Organization [2]. When the figures are converted to dollars, they become even more startling; the analysis estimates that India loses 150 million dollars annually in the economy. This is particularly alarming when taking into account India's sizable vulnerable population.

We suggested a wireless sensor network air pollution monitoring system in this paper to gather data on air pollution. In fact, over the past few decades, air pollution has significantly increased due to a growth in industries, automobiles on our roads, and fast urbanization. Over the subcontinent, industries have expanded and infrastructure projects have been completed. The air quality had declined as a result of combustion in industrial processes, thermal power plants, and other chemical businesses. Monitoring air pollution is thought to be a highly difficult undertaking, but it is nonetheless crucial. In the past, data collection devices were used for gathering data on a regular basis, but this was both time- and money-consuming. The use of WSN can simplify air pollution monitoring and enable more immediate results [3, 4]. The people should be promptly informed about the extent of the air pollution and any changes to the environment so that they can take precautions to ensure their safety. Consequently, a variety of forecasting models are used to forecast the pollution level in advance, but more accurate models are still needed [5].

The people should be promptly informed about the extent of the air pollution and any changes to the environment so that they can take precautions to ensure their safety. There are also many forecasting models in use to anticipate the level of pollution, but a more precise mathematical model is still needed for estimating the level of pollution and the quality index of the air, which has a detrimental effect on human health. In India, the amount of air pollution has gotten so bad [6] that it has replaced smoking as one of the main causes of cancer, heart disease, and several respiratory ailments. The impacts of pollution in the air are concerning since exposure to NO2, SPM, and other pollutants might harm our lungs and respiratory systems [7]. Everyone needs to be made aware of this hazard indicator in that scenario. It is exceedingly challenging to deploy effective sensors everywhere to assess the pollution levels and warn people in advance due to federal economic budget constraints. Therefore, it is preferable to create a model that can predict pollution levels in advance without relying just on real-time sensor data [8]. Global pollution of the air is a serious issue, but it is particularly difficult for emerging nations like India which needs more rapid sustainable development. To tackle this problem properly, cities' air pollution levels must be measured and monitored continuously. By examining the underlying trends included in the managed historical pollutant data, government agencies/ department can forecast the pollutant concentrations specific to a given geo-location. As a result, the main goal of this study is to forecast the values of the parameters that were measured using historical data in order to assess the growing degree of air pollution over a given period of time. The DeepAR method is used to forecast the level of air pollution using a time series forecasting approach. This forecasting technique can provide predicted values on a regular basis to manage air pollution issues efficiently.

2. LITERATURE SURVEY

In recent years, there has been an increase in interest in using artificial neural networks (ANNs) to estimate and predict ambient air pollution. A greater demand is placed on policy-makers and urban city planners to provide quick and economical solutions to mitigate the effects of air pollution because poor air quality in urban areas has been linked to chronic diseases and early deaths of vulnerable people in the general population. Numerous applications for both short- and long-term forecasting have effectively used ANNs in recent years [9-11]. Nevertheless, notwithstanding its effectiveness in many applications, ANN model construction and interpretation involve some challenges. In general, data-driven models are problem-specific, thus there is obviously no one-size-fits-all method for creating them. However, numerous writers claimed that it is still necessary to construct more broad and uniform protocols that describe every step in the creation of ANN models [12–13].

The Weather Research and Forecast combined with the Community Multi-scale Air Quality modelling system and WRF with Chemistry are two common traditional atmospheric models [14–15, 31]. Random Forest and Artificial Neural Networks have recently emerged as cutting-edge techniques for predicting pollutants in the atmosphere or toxins in water [16–18]. A unique technique called Recurrent Neural Network (RNN), a time-dependent intelligent deep learning computer, is utilized to forecast air pollutants first. Random Forest and RNN are more precise, considerably faster to run, and less expensive than classic atmospheric models [14–16], whose accuracy must be guaranteed by regularly updated pollution inventories. This is because pollution inventories are not required with Random Forest and RNN[32][35].

Convolutional neural networks are used in the work in [27] to propose a probabilistic forecasting framework for multiple related time series forecasting. Both parametric and non-parametric situations can use the framework to estimate probability density. The method described in [26, 30] utilized a long short-term memory network (LSTM) that underwent global training to take advantage of the non-linear demand linkages present in an E-commerce product assortment hierarchy. A single prediction model is generated across all of the accessible time series in LSTM-MSNet, a globally trained Long Short-Term Memory network (LSTM), in order to take advantage of the cross series knowledge in a collection of related time series. Furthermore, to complement the LSTM learning process, the methodology outlined in [25][33] incorporates a number of cutting-edge multi-seasonal decomposition algorithms. Every kernel of a multi-kernel convolution layer is fitted to a collection of time series to extract short-term features in nearby regions in the neural network architecture proposed in [23, 24, 34,28]. Convolution-LSTM layer is applied after the output of the convolution layer to combine trends and detect long-term patterns in broader regional areas. An unsupervised pre-trained de-noising auto-encoder rebuilds the output of the model in a fine-tuning step to produce a robust prediction when faced with missing data. We think that only data with appropriate length and high frequency over time can give a precise estimate of the magnitude of the air pollution and characterize its trend and pattern due to the high unpredictability and confounding effects of meteorological conditions. These set the analysis described in [22,29,39,40] apart from current exploratory short-term investigations[36][38].

The majority of forecasting techniques currently in use were created for use in forecasting single or tiny sets of series of samples. In this method, parameters of the model are separately calculated from historical observations for every single series that is presented. DeepAR is a deep learning method which consists of recurrent neural network-based forecasting technique that builds an universal model from past data for every time series in the training samples. This approach extends earlier research on deep learning for time-series information [19] and adapts a recurrent neural architecture design based on LSTM cells [20][41][42] to the forecasting of probabilities problem.

3. METHODOLOGY

Traditional approaches for the prediction of air quality is extremely simple and does require a pre and post processing steps for achieving better performance. Current time series models typically gain knowledge from prior observations and forecast future values entirely based on recent history. These models include Simple Exponential Smoothing (SES) and Autoregressive Integrated Moving Average (ARIMA). Furthermore, cross-learning's potential benefits are not taken advantage of since the parameters are calculated separately for each time series. In order to select the most appropriate model for every time series or group of time series and to account for some significant factors, such as autocorrelation structure, patterns, cyclical nature, and other explanatory variables, approaches demand us to define heuristics, manual efforts, and adjustments steps. In the proposed air quality prediction approach, the first step is to design a sensor network to capture the air quality indicators information.

A WSN has been designed which follows a clustered routing protocol for data aggregation from the sensor nodes. This data has been used for evaluating the performance of the model with respect to a real time data. The problem of forecasting the air quality indicators has been formulated as a regression task and solved using recurrent network based DeepAR model which has the ability to scale using the multiple covariates. By utilizing covariates, DeepAR is able to capture intricate and cluster dependent associations. The time and effort required to choose and prepare variables as well as the model section heuristics generally utilized with conventional forecast models are reduced. The block diagram presented in Fig. 1 depicts the flow of process and data during the model training and inference phase.

By randomly selecting a few training samples from every one of the time series in the data set used for training, DeepAR builds a model. A pair of neighbouring context and forecast windows with specified length makes up each training sample. The context_length hyperparameter regulates the network's ability to look back in time, while the prediction_length hyperparameter regulates the network's ability to make predictions about the future. Elements of the training set that include time-series data that are less than a certain prediction length are ignored by the algorithm during training. DeepAR also automatically inputs lagged data from the target time series to capture seasonality trends.

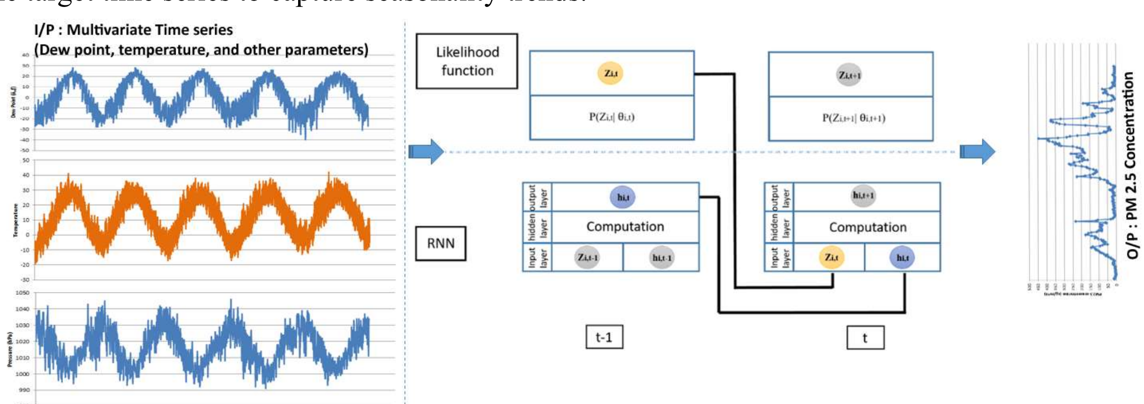


Fig. 1 Schematic view of PM2.5 Concentration prediction using Deep AR

3.1 Dataset

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 [22].

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

3.2 ARCHITECTURE OF DeepAR

Let us consider that $z_{i,t}$ represent a value at time *t* of a series, then the conditional distribution can be modeled by $P(z_{i,t_0:T}|z_{i,1:t_0-1}, x_{i,1:T})$ of the forthcoming values in the series $[z_{i,t_0}, z_{i,t_0+1}, \dots, z_{i,T}]$ with respect to the historical data $[z_{i,1}, z_{i,t_0-2}, \dots, z_{i,t_0-1}]$ where t_0 represents the step from which the value of dependent variable is unknown. The covariates $x_{i,1:T}$ are available for every time steps. The time ranges called as the condition range, and prediction range are represented as $[1, t_{0-1}]$ and $[t_0, T]$. The schematic view of the model is given in the Fig.2

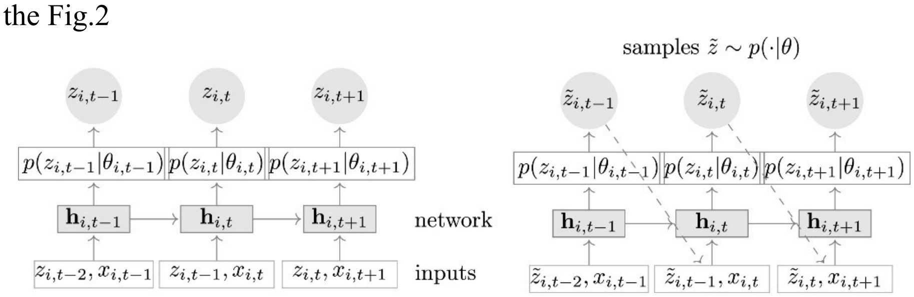


Fig. 2 DeepAR Model Architecture [21]

For every time step $'t'$ the covariates $x_{i,t}$ are given as input to the network, $z_{i,t-1}$ is the value to be predicted at the earlier time step, and $h_{i,t-1}$is the earlier output of the model. The parameters of the likelihood $p(z|\theta)$ are estimated by using the output $h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$. To predict the future values, past values in the series $z_{i,t}$ where $t < t_0$ is given as input to the network. A sample $\hat{z}_{i,t} \sim p(.\,|\theta_{i,t})$ is given as input for the prediction of the next value in the series till all items in the range $t = t_0 + T$ are predicted. The model distribution $Q_\Theta(z_{i,t_0:T}|z_{i,1:t-1}, x_{i,1:T}) = \prod_{t=t_0}^{T} p(z_{i,t}|\theta(h_{i,t}, \Theta))$ represented as product of likelihood that can be parameterized by the output $h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta)$ where h represents a function approximated by a stacked layers of recurrent architecture consisting of

LSTM cells which are parameterized by Θ. The model is auto-regressive as it consumes the last time step $z_{i,t-1}$ and recurrent since the previous model output is given as input for the next step. The likelihood follows a fixed distribution having parameters denoted by $\theta(h_{i,t}, \Theta)$.

The joint samples $\tilde{z}_{i,t_0:T} \sim Q_{\Theta}(z_{i,t_0:T} | z_{i,t_0-1}, x_{i,1:T})$ can be determined based on the model parameters Θ. Initially the value of h_{i,t_0-1} is determined for the time range $[1, \dots \dots, t_0]$. Sample $\hat{z}_{i,t} \sim p(. | \theta(\tilde{h}_{i,t}, \Theta))$ with respect to $\tilde{h}_{i,t} = h(h_{i,t-1}, \tilde{z}_{i,t-1}, x_{i,t}, \Theta)$.

3.3 **Design of WSN for Data Collection**

Instruments that can detect changes in the environment are called sensors. The sources of inputs including light, temperature, motion, and strain might vary. If sensors are connected to a network, they can communicate data with intelligent devices and control devices to create useful information. Wireless sensor networks are interconnected networks that monitor the environment. Nodes aim to be low power and de-centralized in their distribution. At this time, there also exist significant technological barriers to implementing encryption that take into account resource constraints such memory, computing power, lifespan of batteries, and bandwidth. Various security breaches could put your privacy, device accessibility, or availability at risk. Although PM includes a variety of particle kinds, including shapes, sizes, and compositions, it may be easily divided into distinct sub-categories based on the specifics of the particle size. Particles with a median diameter of 2.5 micrometres are the most common type of PM2.5 particle.

The aim of installing WSN is to gather environmental data from sensors or to transmit data to the environment (actuators). Cheap cost environment monitoring systems are appropriate for using the cheap cost silicon sensors in an array form. This array could be improved by adding temperature, pressure, and humidity sensors to track air quality concentrations as well as other physical characteristics for better calibration. The calibration is done in two parts in the first step. Choosing the measurement unit's thickness is also important. It's crucial to use "zero air" to determine a sensor's zero value. However, there is no agreed-upon definition of what constitutes a sensor. "zero air" All sensors are therefore initialized in the morning in clean air. 6LoWPAN is used as the communication protocol which permits the transmission and retrieval of IPv6 packets from networks based on IEEE 802.15.4. It can make sense at a distance of 45 to 90 metres by consuming a fair amount of energy.

A variety of sensing devices and a transmission mechanism are needed for the presented wireless sensor network air pollution monitoring system (WAPMS) in order to send the data to the server. Data is automatically collected by the sensor nodes and transmitted to one or more control stations, which then send it to a sensor network server. The nodes are encouraged to transmit data without any central oversight because the framework instructs them to do so in order to collect the data. In order to best manage the enormous volume of data acquired by the system and to organize the various components involved, the approach divides the region of concern into numerous smaller sections.

One cluster head is deployed in each zone, resulting in clusters with nodes in each area that collect, aggregate, and send data back to the sink from them. The sensor nodes are then randomly placed in the various zones. This arrangement will gather the data and transmit it via multi-hop routing to the cluster head in the corresponding region. There would be a number of sinks that would collect aggregates from the cluster heads and transfer them to the gateway.

Each drain will be assigned a set of cluster heads. The gateway would compile the information from the sinks, transmit it to the database, and then provide it to our submission.

4. EXPERIMENTS AND RESULTS

The autoregressive characteristics of the deepAR network denotes that both input given to the model and output generated by the network scale with the observations of the network, but the non-linear nature of the model poses a limit in the range of operation. The network must learn to scale the input to a range and invert the same at the output layer. This is addressed in this study by dividing the input by a scaling factor and conversely increasing the likelihood parameters. Selecting an optimal value of the scaling factor impacts the overall performance of the network, this study uses the average value $ov_i = 1 + \frac{1}{t_0}\sum_{t=1}^{t_0} z_{i,t}$ which is a heuristic approach that yield better result. The hyperparameters used during the training of DeepAR model was given in Table 2.

Table 2. DeepAR model Hyperparameter Values

Parameter	**Value**
Learning rate	0.001
Likelihood	Gaussian
Mini Batch size	256
Number of layers	04
Dropout rate	0.15
Time Frequency	Hourly
Prediction Length	48
Epochs	200
Weight Decay	1e-6

Weight Decay or L2 regularization offers a method to lessen a deep learning neural network model's overfitting to the training data and enhance the model's performance on test samples. A regularization technique called dropout excludes input and recurrent connections to LSTM units probabilistically from weight and activation updates during network training. As a result, overfitting is decreased and model performance is enhanced. To parameterize a Gaussian likelihood function, DeepAR uses LSTMs. Specifically, to estimate the Gaussian function's parameters.

A stochastic optimization process that uniformly chooses training examples at random extracts a time series with big scales resulting in such time series which cause the model underfit. This imbalance in the data is the second factor to consider. This could be particularly troublesome in the context of demand forecasting, where high-velocity products may behave qualitatively differently from low-velocity ones and where precise forecasts for high-velocity items may be

more crucial for achieving certain business goals. By randomly selecting samples from the examples during training, we can combat this impact. In specifically, our weighted sampling method establishes a window selection probability proportional to v_i. This sampling plan is straightforward but successful in reducing the skew. The standardized covariates have a mean value of zero and a variance value as one by applying the proper normalization. Considering a time series dataset used for training $\{z_{i,1:T}\}_{i=1,\ldots,N}$ along with its covariates $x_{i,1:T}$; such that the value of the $z_{i,t}$ in the prediction range is known, then the Θ parameters of the model can be determined maximizing the log-likelihood $L = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p(z_{i,t}|\theta(h_{i,t}))$. It can be optimized using a SGD by estimating the gradients with reference to Θ.
In all tests, we employ the ADAM optimizer with early halting and conventional LSTM cells with a forget bias set to 1.0, and 200 samples are taken from our decoder to produce predictions. The optimum setting for the hyper-parameters item output embed dimension and # of LSTM nodes is found via a grid-search. To do this, the data prior to the prediction start time are divided into two partitions and utilized as the training set. We fit our model on the training set's first partition, which contains 90% of the data, for every hyper-parameter candidate, then choose the one with the lowest negative log-likelihood for the remaining 10% of the data. The evaluation metrics are assessed on the test set, or the data that comes after the forecast start time, after identifying the optimal values of hyperparameters.

The ND and RMSE are metrics used for assessing the performance of the prediction model.

$$ND = \frac{\sum_{i,t} |z_{i,t} - \hat{z}_{i,t}|}{\sum_{i,t} |z_{i,t}|} \quad \text{Eq. 1}$$

$$RMSE = \frac{\sqrt{\frac{1}{N(T-T_0)} \sum_{i,t} (z_{i,t} - \hat{z}_{i,t})^2}}{\frac{1}{N(T-T_0)} \sum_{i,t} |z_{i,t}|} \quad \text{Eq. 2}$$

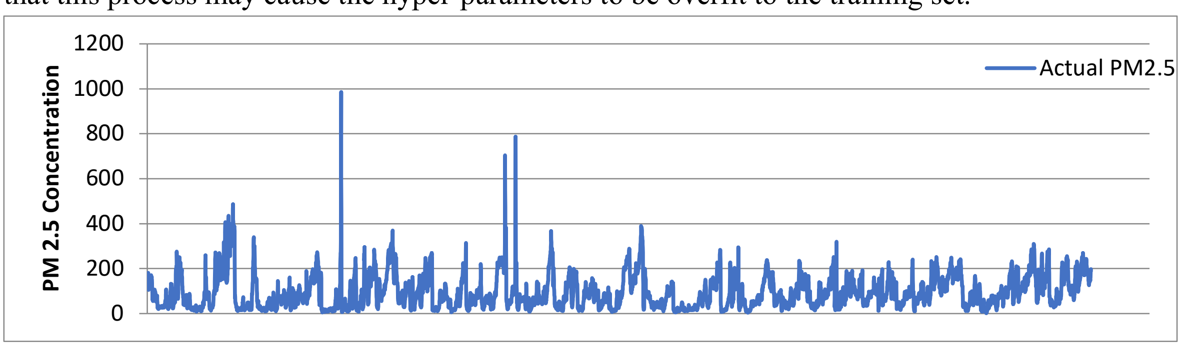
On the datasets, DeepAR performs better than other approach. The outcomes highlight the value of modelling these datasets with a count distribution, while rnn-gaussian produces less accurate findings. The lack of scaling and weighted sampling has a detrimental impact on forecast accuracy overall. For evaluating the performance of the model the following metrics: normalized deviation (ND) and normalized root-mean-square error (NRMSE) were used. The findings indicate that MatFact is underperforming DeepAR. The creation of a model of time series using the ARIMA model is made easier by using Auto ARIMA. The most effective parameter values (p, d, and q) are automatically generated using Auto ARIMA. The model will produce reliable forecast results using the best values that were generated. Also the performance of the DeepAR model is compared with the Auto ARIMA model. The creation of a model of time series using the ARIMA model is made easier by using Auto ARIMA. The most effective parameter values (p, d, and q) are automatically generated using Auto ARIMA. The model will produce reliable forecast results using the best values that were generated. Various techniques, such as log transformation, differencing, removing the rolling mean, taking the square root, etc., can be used to solve this issue if the time series is not stationary. The key benefit of auto ARIMA is that it first does a number of checks to determine whether or not the time series is stationary. Additionally, it makes use of a smart grid search technique to establish the ideal values for p, d, and q.

Table: 3 Comparison of model performance

	Normalized Deviation	Normalized RMSE
Auto-ARIMA	0.158	1.23
MatFact	0.14	1.08
DeepAR	0.085	0.96

The predictive performance of the DeepAR model is presented in Fig. 3 with reference to the actual values of the PM2.5 concentration. The average error in prediction is 14.3 whereas for the MatFact model the average error was observed to be 29.6. The DeepAR model was able to capture the trend and seasonality in the PM2.5 series. The regularization parameter to control model overfitting issues weight decay. In order to prevent overfitting, a larger value of weight decay achieves greater regularization; but, if fixing it too high, the model will be too constrained to learn anything. The weight decay value used in our experiments is 1e-6.

The optimum setting for the hyper-parameters item, such as the number of LSTM nodes and layers, is found using a grid-search. To do this, the data prior to the prediction start time are divided into two divisions and utilized as the training set. We fit our model on the training set's first partition, which contains 90% of the data, for every hyper-parameter candidate, then choose the one with the lowest negative log-likelihood for the remaining 10% of the data. The evaluation metrics are assessed on the test set, or the data that comes after the forecast start time, once the optimal collection of hyper-parameters has been identified. It should be noted that this process may cause the hyper-parameters to be overfit to the training set.



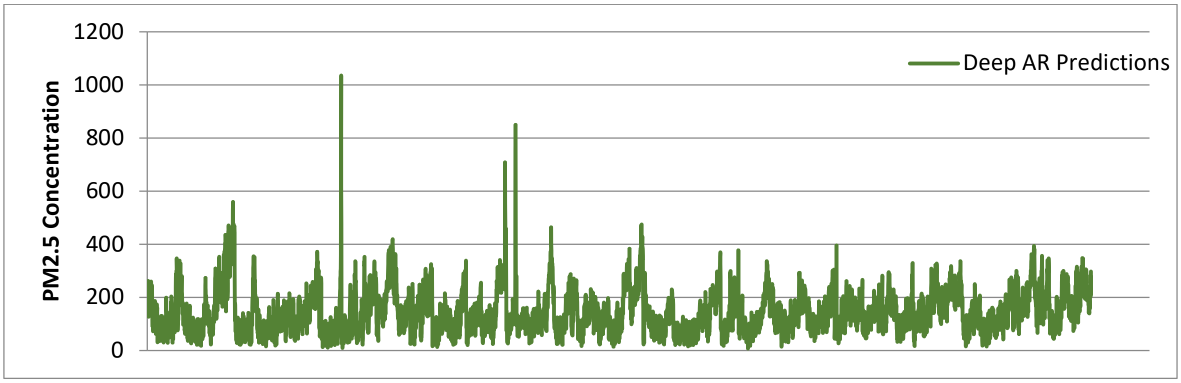


Fig. 3 Analysis of Predictive Performance of DeepAR

CONCLUSION

The forecasting ability of the DeepAR model was demonstrated on a benchmarked dataset and its performance was compared to the contemporary approaches to infer that deep learning techniques can significantly increase forecast accuracy compared to existing prediction methods. Our suggested DeepAR model generates calibrated probabilistic predictions with high accuracy, is efficient at developing a model from associated temporal series, can analyze data having values at varying scales through a scaling and sampling technique, and can learn highly non-linear patterns including seasonality and uncertainty over time varying data. This model can be used for prediction where a few training samples are available and performs well over a wide range of datasets when the model hyperparameters are tuned. In the experiments the hyperparameters value are estimated using a grid search approach. A better method is to estimate the model weights and assess the negative log-likelihood on not overlapping time intervals in addition to distinct windows. For each dataset, we manually adjust the learning rate, keeping it constant during hyper-parameter tweaking.

REFERENCES

1. https://www.iqair.com/world-air-quality-report
2. https://openknowledge.worldbank.org/server/api/core/bitstreams/550b7a9b-4d1f-5d2f-a439-40692d4eedf3/content.
3. Montrucchio, Bartolomeo, et al. "A densely-deployed, high sampling rate, open-source air pollution monitoring WSN." *IEEE Transactions on Vehicular Technology* 69.12 (2020): 15786-15799.
4. Kingsy Grace, R., and S. Manju. "A comprehensive review of wireless sensor networks based air pollution monitoring systems." *Wireless Personal Communications* 108 (2019): 2499-2515.
5. Samal, K. Krishna Rani, et al. "Time series based air pollution forecasting using SARIMA and prophet model." *proceedings of the 2019 international conference on information technology and computer communications*. 2019.
6. A. Kumar and P. Goyal, “Forecasting of air quality in delhi using principal component regression technique,” Atmospheric Pollution Research, vol. 2, no. 4, pp. 436–444, 2011.
7. J. S. Pandey, R. Kumar, and S. Devotta, “Health risks of no2, spm and so2 in delhi (india),” Atmospheric Environment, vol. 39, no. 36, pp. 6868–6874, 2005.

8. L. Chen, J. Xu, L. Zhang, and Y. Xue, “Big data analytic based personalized air quality health advisory model,” in Proc. 13th IEEE Conf. on Automation Science and Engineering (CASE), 2017, pp. 88–93.
9. Biancofiore, F., Busilacchio, M., Verdecchia, M., Tomassetti, B., Aruffo, E., Bianco, S., … Di Carlo, P. (2017a). Recursive neural network model for analysis and forecast of PM10 and PM2.5. Atmospheric Pollution Research, 10, 8–15.
10. Lightstone, S. D., Moshary, F., & Gross, B. (2017). Comparing CMAQ forecasts with a neural network forecast model for PM2.5 in New York. Atmosphere, 8(9).
11. Rahimi, A. (2017). Short-term prediction of NO2 and NO x concentrations using multilayer perceptron neural network: a case study of Tabriz, Iran. Ecological Processes, 6(1), 4.
12. Galelli, S., Humphrey, G. B., Maier, H. R., Castelletti, A., Dandy, G. C., & Gibbs, M. S. (2014). An evaluation framework for input variable selection algorithms for environmental data-driven models. Environmental Modelling & Software, 62, 33–51.
13. Wu, W., Dandy, G. C., & Maier, H. R. (2014). Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. Environmental Modelling & Software, 54, 108–127.
14. Feng, Rui, et al. "Ethylene, xylene, toluene and hexane are major contributors of atmospheric ozone in Hangzhou, China, prior to the 2022 Asian Games." *Environmental Chemistry Letters* 17 (2019): 1151-1160.
15. Xu, R., Tie, X., Li, G., Zhao, S., Cao, J., Feng, T., Long, X., 2018. Effect of biomass burning on black carbon (BC) in South Asia and Tibetan Plateau: the analysis of WRFChem modeling. Sci. Total Environ. 645, 901e912.
16. Kaminska, J., 2019. A random forest partition model for predicting NO2 concentrations from traffic flow and meteorological conditions. Sci. Total Environ. 651 (1), 475e483.
17. Debnath, A., Majumder, M., Pal, M., 2015. A cognitive approach in selection of source for water treatment plant based on climatic impact. Water Resour. Manag. 29 (6), 1907e1919.
18. Zhang, Y., Zhang, X., Wang, L., Zhang, Q., Duan, F., He, K., 2016. Application of WRF/ chem over East asia: Part I. Model evaluation and intercomparison with MM5/ CMAQ, 124. Atmospheric Environment, pp. 285e300.
19. Dieleman, Sander, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).
20. Hochreiter S., Schmidhuber J. Long short-term memory Neural Compututation, 9 (8) (1997), pp. 1735-1780.
21. Salinas, David, et al. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." *International Journal of Forecasting* 36.3 (2020): 1181-1191.
22. 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.

23. Asadi Reza and Regan Amelia C.. 2020. A spatial-temporal decomposition based deep neural network for time series forecasting. *Applied Soft Computing* 87 (2020), 105963.
24. Ballestra Luca Vincenzo, Guizzardi Andrea, and Palladini Fabio. 2019. Forecasting and trading on the VIX futures market: A neural network approach based on open to close returns and coincident indicators. *International Journal of Forecasting* 35, 4 (2019), 1250–1262.
25. Bandara Kasun, Bergmeir Christoph, and Hewamalage Hansika. 2020. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
26. Bandara Kasun, Shi Peibei, Bergmeir Christoph, Hewamalage Hansika, Tran Quoc, and Seaman Brian. 2019. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *Proceedings of the International Conference on Neural Information Processing*. Springer, 462–474.
27. Kumar, M. Sunil, et al. "Deep Convolution Neural Network Based solution for Detecting Plant Diseases." Journal of Pharmaceutical Negative Results (2022): 464-471.
28. Ganesh, D., et al. "Implementation of AI Pop Bots and its allied Applications for Designing Efficient Curriculum in Early Childhood Education." International Journal of Early Childhood 14.03: 2022.
29. Kumar, M. Sunil, et al. "Applying The Modular Encryption Standard To Mobile Cloud Computing To Improve The Safety Of Health Data." Journal of Pharmaceutical Negative Results (2022): 1911-1917.
30. Prasad, Tvs Gowtham, et al. "Cnn Based Pathway Control To Prevent Covid Spread Using Face Mask And Body Temperature Detection." Journal of Pharmaceutical Negative Results (2022): 1374-1381.1911-1917.
31. P. Sai Kiran. "Power aware virtual machine placement in IaaS cloud using discrete firefly algorithm." Applied Nanoscience (2022): 1-9.
32. Peneti, S., Sunil Kumar, M., Kallam, S., Patan, R., Bhaskar, V. and Ramachandran, M., 2021. BDN-GWMNN: internet of things (IoT) enabled secure smart city applications. Wireless Personal Communications, 119(3), pp.2469-2485.
33. Malchi, Sunil Kumar, et al. "A trust-based fuzzy neural network for smart data fusion in internet of things." Computers & Electrical Engineering 89 (2021): 106901.
34. Sangamithra, B., P. Neelima, and M. Sunil Kumar. "A memetic algorithm for multi objective vehicle routing problem with time windows." 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE). IEEE, 2017.
35. Sunil Kumar, M., and A. Rama Mohan Reddy. "An Efficient Approach for Evolution of Functional Requirements to Improve the Quality of Software Architecture." Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, New Delhi, 2016. 775-792.
36. Kumar, T. P., & Kumar, M. S. (2021). Optimised Levenshtein centroid cross-layer defence for multi-hop cognitive radio networks. IET Communications, 15(2), 245-256.

37. Natarajan, V. Anantha, et al. "Segmentation of nuclei in histopathology images using fully convolutional deep neural architecture." 2020 International Conference on computing and information technology (ICCIT-1441). IEEE, 2020.
38. Chen Yitian, Kang Yanfei, Chen Yixiong, and Wang Zizhuo. 2020. Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing* 399 (2020), 491–501.
39. 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.
40. Veeramanikandan, V., 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)*. IEEE, 2019.
41. Veeramanikandan, Varadharajan, and Mohan Jeyakarthic. "Parameter-Tuned Deep Learning Model for Credit Risk Assessment and Scoring Applications." *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)* 14.9 (2021): 2958-2968.
42. 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.