

Multivariate Time Series Analysis and Batch Normalization for Air Quality Prediction in Long Short-Term Memory Networks

Dr. Sridhar Patthi
Professor, CSE (AI&ML)
Institute of Aeronautical
Engineering,
Hyderabad
sridharp35@gmail.com

Ila Chandana Kumari P
Associate Professor, CSE
Hyderabad Institute of
Technology and Management,
Hyderabad
ilachandana@gmail.com

Dr. AVS Pavan Kumar
Assistant Professor, Department
of CSE,
GIET University,
Gunupur
avspavankumar@giuet.edu

Tirumala Manav
Student, B.Tech - CSE (IOT)
Hyderabad Institute of
Technology and Management,
Hyderabad
thirumalamanav123@gmail.com

Abstract: The well-being of every individual hinge on the level of oxygen they breathe. The purity of oxygen is intricately tied to the presence of trees in the vicinity, with forests playing a crucial role in enhancing oxygen quality. However, in urban areas, deforestation contributes to elevated air pollutants, leading to adverse health effects in humans. Understanding the sources of air impurities and the specific toxins responsible is essential for individuals and organizations to comprehend the rate of air contamination. This knowledge aids industries and health organizations in predicting and addressing air pollution, ultimately supporting the World Health Organization's efforts. Research focused on air quality data, particularly utilizing Deep Learning approaches, plays a pivotal role in making informed decisions and conducting suitable analyses. The primary objective is to offer users real-time, effective predictions of air quality data using multivariate time series and a BN-LSTM. This combination enables timely and accurate predictions of air quality through Deep Learning. The incorporation of BN-LSTM has notably improved the average magnitude of errors to 28.06. This advancement facilitates enhanced support for predictions by the World Health Organization and other entities concerned with air quality.

Keywords— Deep Learning, Machine Learning, Univariate Time Series, Long Short-Term Memory (LSTM).

I. INTRODUCTION

Air pollution denotes the presence of harmful substances in the atmosphere, posing adverse effects on human health, the environment, and climate. Primary sources of air pollution encompass vehicle emissions, the combustion of fossil fuels, and industrial activities. Various pollutants characterize the air, including particulate matter, ground-level ozone, nitrogen dioxide, and carbon dioxide. The repercussions of air pollution extend to human health, contributing to respiratory, cardiovascular, and cancer-related issues. Environmental impacts include ecosystem damage, acid rain, and the release of greenhouse gases.

Effectively addressing air pollution necessitates a multifaceted approach, combining regulatory measures, technological advancements, and heightened public awareness. This comprehensive strategy aims to mitigate the far-reaching impacts of air pollution on both human health and the environment.

The prediction and management of air pollution stand as a crucial application of technology and science, harnessing advancements across diverse fields. This multidisciplinary approach combines technology, scientific principles, and

data analysis to deliver timely and precise information for mitigating the effects of air pollution on public health and the environment. Within this domain, deep learning, a subset of machine learning and artificial intelligence (AI), has emerged as a promising tool. Notably, deep learning models, particularly neural networks, demonstrate remarkable proficiency in handling intricate patterns and large datasets, thereby enhancing their utility in forecasting air quality.

Time series forecasting is a predictive modelling technique focused on estimating future values by analysing past observations arranged in chronological order. The main goal of time series analysis is to comprehend the inherent patterns, trends, and seasonality within historical data and leverage this knowledge to forecast future values. This methodology finds application in diverse fields such as finance, economics, weather forecasting, and environmental science.

Time series forecasting holds pivotal significance in decision-making, resource planning, and risk management across diverse industries. The selection of an appropriate model hinges on the distinctive characteristics of the time series data and the targeted forecasting horizon. The BN-LSTM, a variant of recurrent neural networks (RNN), is tailored to capture long-term dependencies and memory within time series data, rendering it well-suited for sequential prediction tasks.

Multivariate time series forecasting is a method that predicts future values of a single variable based on its historical observations. This approach is simpler and proves effective when there are limited relationships between variables or when the objective involves predicting in consideration with multiple variables.

II. RELATED WORK

In the present era, governments worldwide, spanning developed and developing nations like America, Russia, India, China, and Japan, are increasingly prioritizing the regulation of air quality. Air pollution, arising from both natural and human-induced sources, is exacerbated by factors such as heavy traffic, extensive vehicle usage, and the burning of plastics. The continuous monitoring of air quality levels is considered crucial in the collective effort to reduce air pollution. Statistical models, particularly regression, play a key role in establishing relationships between variables. Logistic regression, in particular, is instrumental in determining whether a given data sample is polluted or not

[1]. For forecasting future values of particulate matter based on historical data, auto-regression is employed. Utilizing existing records regarding particulate matter assists in proactively mitigating its hazardous levels. A comprehensive dataset includes atmospheric details such as temperature, dew point, pollution levels, atmospheric pressure, wind speed, wind direction, and cumulative hours of rain and snowfall.

Prediction constitutes a vital component of air pollution forecasting, entailing the anticipation of future events through the examination of historical records. Within this forecasting framework, the strategic analysis of features and the selection of pertinent ones provide clear advantages by incorporating only essential information. The utilization of a mathematical technique called interpolation becomes crucial for estimating values of an unknown function, $g(x)$, for a specified argument x , typically within an interval $[a, b]$. In the calculation of finely detailed urban air quality predictions, emphasis is particularly placed on feature selection and interpolation as critical considerations.

Each of the three aspects mentioned earlier, namely interpolation, prediction, and feature selection, has been individually tackled through various existing works and diverse models. A noteworthy model in this context is "Deep Air Learning [DAL]," which integrates semi-supervised learning and feature selection [2]. Rigorous experiments conducted with real data from Beijing, China, showcase its effectiveness. A comparative analysis of existing literature indicates that Deep Air Learning surpasses peer models, particularly in the domain of air quality, encompassing weather prediction. This model holds the potential to predict future pollution levels based on atmospheric conditions [3].

In conjunction with deep learning, atmospheric dispersion models play a significant role. These models, collectively known as Atmospheric Dispersion Modelling (ADM), play a key role in addressing challenges related to air quality prediction. Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models, implemented using the open-source software library Tensor Flow, emerge as viable choices for air pollution forecasting [4]. The incorporation of advanced technologies and models emphasizes the continuous endeavours to improve the accuracy and efficiency of predicting air quality and atmospheric conditions.

Principal Component Analysis (PCA) and the Ordinary Least Squares (OLS) Model have revealed a consistent finding, indicating that a specific gas, namely NO₂, is the primary contributor to imperfections and pollution in the air. These models have successfully pinpointed the key parameter influencing air pollution [5].

The utilization of feedforward artificial neural networks, employing principal components as inputs, has streamlined complexity and eliminated data collinearity in the model. This approach has successfully predicted the ozone concentration for the next day based on the data from the current day [6].

III. DESIGN METHODOLOGY

A. Input Data:

The dataset, sourced from Kaggle, a data service provider, was acquired with the aim of training and predicting a target value. Spanning five years, this dataset

concentrates on air pollution forecasting, encompassing data on air pollution levels and atmospheric weather conditions in Beijing. The initial dataset comprises diverse features including row number, year, month, day, hour, PM2.5 concentration, dew point, temperature, pressure, wind direction, wind speed, and cumulative hours of snow and rainfall. Currently, all these data points are considered as raw data and will undergo additional processing for subsequent analysis and modelling.

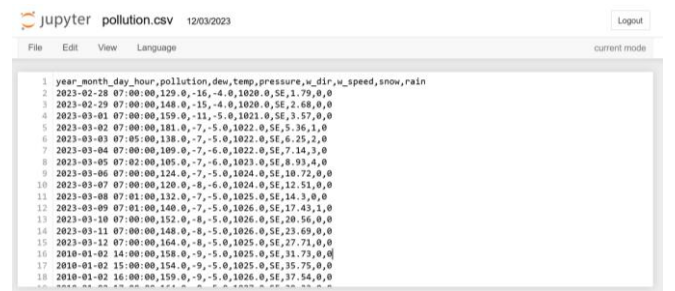
The dataset also incorporates values obtained from a hardware device designed for recording. Specifically, data recorded for one month, spanning from 28 February 2023 to 12 March 2023, is included, representing observations in Kukatpally, Hyderabad. The model is tested across all days starting from 7:00 AM. The recordings gathered from the MQ Series Sensor model (refer to Fig 1), designed for this purpose, are integrated into the dataset obtained from Kaggle, a data service provider.

B. Data Preprocessing:

In certain instances, datasets might present inaccuracies or incomplete information, introducing challenges, especially during model training. These issues can have detrimental effects on the overall system performance and the accuracy of predictions. To mitigate these concerns, data pre-processing becomes a critical step, intending to uphold data quality and ensure a cohesive dataset. Typically conducted before data cleaning, this process encompasses two main types: automated pre-processing and manual pre-processing.

Automated pre-processing employs computational techniques and algorithms to manage tasks such as imputing missing data, scaling, normalization, and feature engineering. These methods prove efficient and time-saving, especially when handling large datasets.

In contrast, manual pre-processing involves human intervention to execute tasks like data validation, outlier detection, and feature selection. Human expertise is essential for making subjective decisions that automated algorithms may find challenging, particularly when dealing with data requiring domain-specific knowledge.



	year	month	day	hour	pollution	dew	temp	pressure	w_dir	w_speed	snow	rain
1	2023	02	28	07:00:00	129.0	-15.0	4.0	1020.0	SE	1.75	0.0	0.0
2	2023	02	29	07:00:00	148.0	-15.0	4.0	1020.0	SE	2.68	0.0	0.0
3	2023	03	01	07:00:00	159.0	-11.0	5.0	1021.0	SE	3.57	0.0	0.0
4	2023	03	02	07:00:00	181.0	-7.0	5.0	1022.0	SE	5.36	1.0	0.0
5	2023	03	03	07:05:00	138.0	-7.0	5.0	1022.0	SE	6.25	2.0	0.0
6	2023	03	04	07:00:00	109.0	-7.0	6.0	1022.0	SE	7.14	3.0	0.0
7	2023	03	05	07:02:00	105.0	-7.0	6.0	1023.0	SE	8.93	4.0	0.0
8	2023	03	06	07:00:00	124.0	-7.0	5.0	1024.0	SE	10.72	0.0	0.0
9	2023	03	07	07:00:00	120.0	-8.0	6.0	1024.0	SE	12.51	0.0	0.0
10	2023	03	08	07:01:00	132.0	-7.0	5.0	1025.0	SE	14.30	0.0	0.0
11	2023	03	09	07:01:00	140.0	-7.0	5.0	1026.0	SE	17.43	1.0	0.0
12	2023	03	10	07:00:00	152.0	-8.0	5.0	1026.0	SE	20.56	0.0	0.0
13	2023	03	11	07:00:00	148.0	-8.0	5.0	1026.0	SE	23.69	0.0	0.0
14	2023	03	12	07:00:00	164.0	-9.0	5.0	1025.0	SE	27.71	0.0	0.0
15	2010	01	02	14:00:00	158.0	-9.0	5.0	1025.0	SE	31.73	0.0	0.0
16	2010	01	02	15:00:00	154.0	-9.0	5.0	1025.0	SE	35.75	0.0	0.0
17	2010	01	02	16:00:00	159.0	-9.0	5.0	1026.0	SE	37.54	0.0	0.0

Fig. 1. Recordings of MQ series sensor model with multivariate data

Automated and manual pre-processing methods work in tandem to improve the overall quality of the dataset, contributing to the effectiveness of machine learning models. The overarching objective is to establish a clean, well-structured dataset that fosters accurate and reliable model training and predictions.

C. Basic Data Preparation:

The initial datasets require preparation before they can be subjected to further analysis or application. The initial step involves consolidating individual date and time information

into a unified date-time format. In pandas, these date-time values serve as an index for the dataset. Within the first 24 hours, certain data fields contain "NA" values, which are subsequently replaced with "0" values to ensure data consistency. A Python script is utilized to load the raw datasets and parse them into a format suitable for analysis. Furthermore, any columns lacking values are removed from the dataset.

D. Multivariate LSTM Forecast Model and LSTM Data Preparation:

To ready the air pollution dataset for Long Short-Term Memory (LSTM) modelling, several crucial steps are involved. First and foremost, structuring the dataset as a supervised learning problem is necessary, and the normalization of input variables is essential. The *series_to_supervised()* function proves useful for transforming the dataset into a supervised learning format. Here's a breakdown of the process:

1) Supervised Learning Problem Transformation:

Utilize the *series_to_supervised()* function to structure the dataset as a supervised learning problem. This entails arranging the data into input-output pairs that are appropriate for training an LSTM model.

2) Loading Pollution Dataset and Integer Encoding:

Load the 'pollution.csv' dataset, ensuring it encompasses the requisite features for analysis. Carry out integer encoding, particularly for the wind speed feature, to numerically represent categorical data.

3) Normalization of Input Variables:

Normalize the dataset by rescaling the attributes to conform to the necessary range. This step is vital to ensure that all features contribute equally to the model training.

4) Transformation into Supervised Learning Format:

Lastly, convert the normalized dataset into a supervised learning problem, where each observation incorporates both input features and the corresponding output label. This transformation is essential for effectively training the LSTM model.

Normalization is a standard procedure in the data preparation for machine learning models, ensuring that features are on a similar scale and preventing any particular feature from dominating the training process. This systematic approach establishes the foundation for the utilization of LSTM, a type of recurrent neural network well-suited for sequential data such as time series.

E. Evaluation of a Model:

After the model has been trained on the complete test dataset and is considered suitable, the subsequent step is to perform forecasting (refer to Fig 2). This entails generating predictions based on the learned patterns of the model. Following this, the forecasts are integrated with the original test dataset. To maintain consistency, inverse scaling is applied to the test dataset, particularly for the pollution numbers, restoring them to their original units.

The Root Mean Square Error (RMSE) plays a crucial role in evaluating the model. RMSE is utilized to measure the disparity between predicted and observed values. A smaller RMSE signifies higher accuracy in the prediction system, indicating that the model's forecasts closely match the actual observed values.

Essentially, the RMSE functions as a metric to assess the predictive performance of the model. It offers a measure of the average magnitude of errors, providing insights into how effectively the model captures and reproduces the patterns present in the test dataset.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\text{predicted}_i - \text{actual}_i)^2}{N}}$$

where,

predicted_i = The predicted value for the i th observation

actual_i = The observed / actual value for the i th observation

N = Total number of observations

IV. METHODOLOGY

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is designed to address the vanishing gradient problem associated with traditional RNNs. The vanishing gradient problem occurs when the gradients of the loss function become extremely small during the training of deep neural networks, making it challenging for the network to learn and retain information over long sequences. LSTM networks were introduced to overcome this issue by incorporating a memory cell that can maintain information over extended periods. The architecture includes three main components: a cell state, an input gate, and an output gate Fig 2.

Cell State:

The cell state acts as a conveyor belt, allowing information to flow unchanged when needed. This helps in preserving long-term dependencies in the data.

Input gate:

The input gate regulates the flow of information into the cell state. It decides which values from the input should be updated and added to the cell state.

Output gate:

The output gate determines the next hidden state based on the current input, the previous hidden state, and the information from the cell state. It decides what information the LSTM will output.

LSTM networks are particularly effective for sequential data and time series analysis, making them well-suited for tasks like natural language processing, speech recognition, and, as mentioned, air pollution forecasting. They can capture and remember patterns in data over long sequences, making them suitable for modelling complex temporal dependencies.

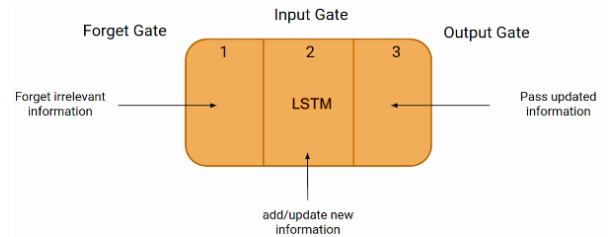


Fig. 2. Block Diagram of LSTM

Multivariate Time Series:

A multivariate time series (MTS) is a collection of multiple time series that are observed or measured over the same time intervals. In simpler terms, it's a sequence of data points collected or recorded at different time points, where each data point consists of multiple variables or features. These variables are observed simultaneously, making it a multivariate time series.

For example, consider a scenario where you are monitoring the weather at a particular location. A univariate time series might track only temperature changes over time. On the other hand, a multivariate time series could include temperature, humidity, wind speed, and precipitation, all observed at the same time intervals. Key characteristics of multivariate time series:

Multiple variables: It involves more than one variable or feature measured at each time point.

Temporal structure: The data is ordered chronologically, with observations recorded at regular time intervals.

Interdependencies: Variables in a multivariate time series may be interdependent, meaning the value of one variable at a given time can be influenced by the values of other variables at the same or previous time points.

Analysing multivariate time series data often involves advanced statistical and machine learning techniques, such as vector autoregression (VAR), dynamic time warping, recurrent neural networks (RNNs), or long short-term memory networks (LSTMs). These methods are designed to capture the temporal dependencies and relationships between the variables in the time series data.

Batch Norm (BT) LSTM Model:

The deep learning model is designed using the Keras library, particularly utilizing the Sequential model, which is well-suited for organizing a sequential stack of layers. This model is specifically crafted for tasks involving sequence prediction or time series forecasting. It leverages Long Short-Term Memory (LSTM) layers, a type of recurrent neural network architecture renowned for its efficacy in capturing long-term dependencies within sequential data.

The initial layer is an LSTM layer comprising 100 units, set to return sequences by virtue of the `return_sequences=True` parameter. This layer plays a pivotal role in capturing temporal patterns within the input data. The input shape is defined as `(X_train.shape[1], X_train.shape[2])`, indicating the count of time steps in each sequence and the number of features, respectively.

To mitigate overfitting, a dropout layer is incorporated with a rate of 0.3 following the initial LSTM layer. This layer selectively omits a proportion of input units during training, serving as a regularization measure. Subsequent to the dropout layer, a batch normalization layer is introduced to normalize the activations of the LSTM layer. This normalization step contributes to stabilizing and expediting the training process.

The model proceeds with a second LSTM layer, featuring 100 units and configured to return sequences. Subsequently, another dropout layer and batch normalization layer are added, replicating the structure implemented after the initial LSTM layer. Following this, a

third LSTM layer is introduced, also comprising 100 units but without the `return_sequences` parameter. This layer is designed to capture higher-level abstractions from the sequence.

Just like the preceding LSTM layers, the third layer is coupled with a dropout layer and batch normalization layer, ensuring regularization and normalization. The ultimate layer is a dense layer with a single unit, tailored for regression tasks where the model predicts a continuous value. The selection of the Adam optimizer and mean squared error loss function during compilation adheres to standard practices for regression problems, striving to minimize the squared difference between predicted and actual values.

To train the model, the `fit` method is utilized on the training data (`X_train` and `Y_train`) for 50 epochs, employing a batch size of 1024. Following the successful training, the model is saved as 'AirPollutionMultivariate.h5'.

In overview, the model architecture is thoughtfully crafted, featuring a strategic combination of LSTM layers complemented by dropout regularization and batch normalization. This design reflects a thoughtful approach to effectively handle sequential data, addressing common challenges encountered in training deep neural networks. The training process spans 50 epochs, and post-training, the model is saved for potential future use or deployment. For fine-tuning, one can explore adjustments to hyperparameters and conduct further experimentation, tailoring the model to the specific characteristics of the dataset and the intricacies of the forecasting task.

Compared to the original LSTM model, the BT-LSTM model exhibits notable enhancements. The layer count has expanded from 3 to 5, incorporating Batch Normalization after each layer. Additionally, the dropout rate has been elevated from 0.2 to 0.3 for each layer. The LSTM units have been doubled, escalating from 50 to 100 for every layer. The epoch count remains consistent at 15 for each layer, while the batch size experiences a substantial increase, soaring from 32 to 1024. Importantly, the dense layer, optimizer, and input shape are retained as constants in this modified architecture.

Batch Normalization:

Batch normalization is a supervised learning technique employed to normalize the intermediate outputs between layers within a neural network. This normalization process resets the distribution of the output from the preceding layer, facilitating the subsequent layer in more effectively analysing the data.

V. IMPLEMENTATION AND RESULTS

In the data preparation phase, shown in Fig. 3, a pivotal step involves consolidating date-time information into a unified date-time format, subsequently utilized as an index in the pandas library. Additionally, columns containing values labelled as "no" are removed from the dataset (Fig.4). Furthermore, occurrences of "NA" values are handled by replacing them with the value "0." These procedures collectively contribute to the creation of a well-structured and usable dataset as depicted in Fig. 5 and Fig. 6, thereby facilitating subsequent analysis and modelling processes as described in Fig. 7.

```

jupyter AirPollutionMultivariate Last Checkpoint: 11/30/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

In [3]: # Data cleaning
dataset.drop('No', axis=1, inplace=True)
dataset.columns = col_names
dataset['pollution'].fillna(0, inplace=True)
dataset = dataset[24:] # drop the first day
print(dataset.head(5))
dataset.to_csv('pollution.csv') # save new CSV

year_month_day_hour  pollution dew temp pressure w_dir w_speed snow rain
2018-01-02 00:00:00    129.0 -16 -4.0  1020.0 SE  1.79  0  0
2018-01-02 01:00:00    148.0 -15 -4.0  1020.0 SE  2.68  0  0
2018-01-02 02:00:00    159.0 -11 -5.0  1021.0 SE  3.57  0  0
2018-01-02 03:00:00    181.0 -7 -5.0  1022.0 SE  5.36  1  0
2018-01-02 04:00:00    138.0 -7 -5.0  1022.0 SE  6.25  2  0

```

Fig. 3. Preparing dataset of multivariate

```

In [4]: # Load dataset
df = pd.read_csv('pollution.csv', header=0, index_col=0)
df.describe()

Out[4]:

```

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

Fig. 4. Loading data including all variates

```

jupyter AirPollutionMultivariate Last Checkpoint: 11/30/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

In [14]: # Splitting the dataset
n_train = 24*365
X_train, X_test = X[n_train:], X[:n_train]
print('X_train', X_train.shape)
print('X_test', X_test.shape)

Y_train, Y_test = Y[n_train:], Y[:n_train]
print('Y_train', Y_train.shape)
print('Y_test', Y_test.shape)

X_train (35036, 4, 8)
X_test (8768, 4, 8)
Y_train (35036, 1)
Y_test (8768, 1)

```

Fig. 5. Splitting the dataset for training and testing

```

jupyter AirPollutionMultivariate Last Checkpoint: 11/30/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

In [28]: # Train the model
model.fit(X_train, Y_train, epochs = 50, batch_size = 1024)

Epoch 1/50
35/35 [=====] - 1s 13ms/step - loss: 9.4087e-04
Epoch 2/50
35/35 [=====] - 0s 9ms/step - loss: 8.7663e-04
Epoch 3/50
35/35 [=====] - 0s 9ms/step - loss: 8.9734e-04
Epoch 4/50
35/35 [=====] - 0s 9ms/step - loss: 9.2554e-04
Epoch 5/50
35/35 [=====] - 0s 9ms/step - loss: 8.4957e-04
Epoch 6/50
35/35 [=====] - 0s 9ms/step - loss: 8.7828e-04
Epoch 7/50
35/35 [=====] - 0s 8ms/step - loss: 8.5369e-04

```

Fig. 6. Training the model with epoch 50 in multivariate

```

jupyter AirPollutionMultivariate Last Checkpoint: 11/30/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (pykernel)

return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

In [37]: from sklearn.metrics import mean_squared_error
mse = mean_squared_error(Y_tested, Y_predicted)
print('Mean squared error', mse)
print('RMSE', np.sqrt(mse))
print('Mean of Test data', np.mean(Y_tested))

RMSE 28.698801
Mean of Test data 96.041435

```

Fig. 7. Results of RMSE and mean test data

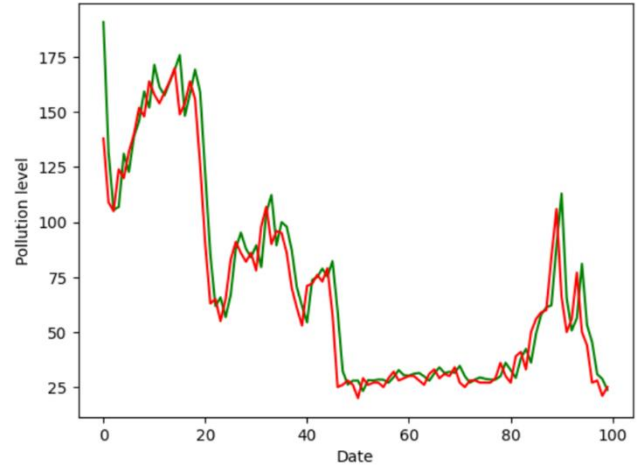


Fig. 8. Accuracy of prediction in multivariate LSTM

The RMSE values for the univariate models as shown in Fig. 8 is consistently high, indicating a high level of accuracy when comparing predicted and actual values. The red line shows actual pollution level and green line shows predicted pollution level. Therefore, we can confidently conclude that the designed model provides accurate results without exhibiting bias, particularly in the univariate context.

The training time for the dataset falls within the range of 1 ms to 10 ms, showcasing significant improvement compared to the existing model. The proposed BN-LSTM model's performance is validated by assessing the RMSE values, demonstrating a reduction from 29.03 to 28.06. Additionally, the mean for the test data is recorded at 96.04.

This accomplishment was realized using a model with 5 layers. However, upon increasing the number of layers further to minimize RMSE, it was observed that the model became overfit. Consequently, the paper presents an optimal RMSE value in relation to the number of layers for forecasting air pollution.

VI. CONCLUSION

The central emphasis is on tackling air quality issues and understanding the health risks linked to polluted air, such as heart and lung problems. It becomes crucial to raise awareness about air quality. Deep learning platforms play a vital role in predicting and forecasting air pollution. The use of Univariate time series is instrumental in minimizing errors and enhancing prediction accuracy. Prior to forecasting, time series data undergoes pre-processing through auto-encoders. The forecasting model, specifically utilizing Batch Norm - Long Short-Term Memory (BN-LSTM) for multivariate time series, predicts air pollution trends based on input variables.

To boost the efficiency of the system, it is recommended to integrate it with IOS and Android platforms. This integration facilitates seamless alerts to the public regarding atmospheric air quality levels, employing Multivariate Time Series Analysis. Furthermore, extending the application to wearable devices proves valuable, offering individuals practical tools to safeguard the environment and living beings from the harmful impacts of air pollution.

REFERENCES

- [1] Aditya C R, Chandana R Deshmukh, Nayana D K, Praveen Gandhi Vidyavastu, "De-tecton and Prediction of Air Pollution using Machine Learning Models", International Journal of Engineering Trends and Technology, Vol 59, No. 4, 2018.

- [2] Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li, Zhongfei Zhang, "Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-grained Air Quality", arXiv:1711.00939, Vol 2, 2018.
- [3] Xia Xi, Zhao Wei, RuiXiaoguang, Wang Yijie, Bai Xinxin, Yin Wenjun, Don Jin, "A Comprehensive Evaluation of Air Pollution Prediction Improvement by a Machine Learning Method", IBM CRL Beijing, China, SOLI/ICT4ALL, 2015.
- [4] Yi-Ting Tsai, Yu-Ren Zeng, Yue-Shan Chang, "Air pollution forecasting using RNN with LSTM", Taiwan, Vol 7, Issue 5, 2019.
- [5] Shi, J.P., Harrison, R. M., "Regression modelling of hourly NOX and NO₂ concentrations in urban air in London", Atmospheric Environment, No 31, 1997.
- [6] Sousa, S.I.V., Martins, F.G., Alvimö Ferraz, M.C.M., Pereira, M.C., "Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations", Environmental Modelling and Software, Vol 22, 2007.