A DEEP LEARNING MODEL FOR REDUCING DATA TRANSMISSION IN IOT NETWORK

Zohaa Mubarak Al-Qfail, Zulaiha Ali Othman

Center Artificial Intelligence Technology (CAIT),
Fakulty of Information Science & Technology, Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor Malaysia.

gp07230@siswa.ukm.edu.my, Zao @ukm.edu.my

ABSTRACT

Overload communication in the IoT network is one of the reasons a Smart city-based IoT was failed. The continuously capturing data from sensors into the IoT network are generating massive data. The data collected often has some spatial or temporal redundancy and is unnecessary, which can be eliminated. Two popular approaches for this issue: data compression and reduce data transmission. Previous studies have proposed a Dual Prediction (DP) method in Spatio-temporal correlation on sensors' data in the Wireless Sensors Networks (WSNs) to reduce data transmission using Machine and Deep Learning techniques. Therefore, this study aims to reduce data transmission for an atmospheric IoT application using deep learning. The experiments were conducted on three stations Malaysia Atmospheric datasets: Putrajaya, Petaling Jaya, and Tanjung Malim, in which each station consists of five attributes: Temperature, Humidity, Wind Speed, NO2, and NOx. Three LSTM algorithms are used to investigate the best prediction model for each attribute: Stack LSTM, Bi-directional LSTM (BiD LSTM), and Convolutional LSTM (ConvLSTM). The experiment applied ten times run of walk-forward validation method for each attribute using MAPE measurement. The result shows ConvLSTM algorithm has consistently shown the best model on Putrajaya and Tanjung Malim datasets. Later, the ConvLSTM prediction model has been used to forecast data for all three stations. The experiment results showed that ConvLSTM had reduced temperature transmission data for Putra Jaya, Petaling Jaya, and Tanjung Malim by about 69.5%, 67.31%, and 70.5%, respectively. It also reduced Wind Speed 77.4%, 38.95% and 73.57% and Humidity 19.8%, 10.9% and 12.5% respectively with 0.5 threshold. Furthermore, it reduces NOx by about 52.9%, 21.86%, 74.6%, and reduced NO2 and 73.1%, 69.24%, and 93.22% for Putra Jaya, Petaling Jaya, and Tanjung Malim, respectively, with a 0.005 threshold. Based on the best accuracy result of Putrajaya, it can be concluded that deep learning has contributed to reducing IoT data transmission from up to 77.40%.

Keywords: Data transmission reduction, Deep learning techniques, Stack LSTM, Bi-directional LSTM, Convolutional LSTM, Time series forecasting, Wireless Sensors Networks, IoT network

I. INTRODUCTION

The Internet of Things (IoT) is a global network that connects a wide range of hardware and people using advanced information and communication technologies. The smart city is one of the applications of the IoT that aims to improve the quality of human life. Many industrial, health, and service applications can be managed and monitored intelligently within the smart city through Wireless Sensor Networks. WSNs are essential for smart city applications and IoT systems because they are the main data capture source through sensors.

One of the most important applications that use WSNs is Monitoring applications. Through monitoring applications, data around changes are collected in environments such as factories, hospitals, and weather-tracking stations. In monitoring applications, sensor nodes generate vast volumes of critical data that must be transmitted continuously over long periods of time in order to be used for effective action and decision-making. Due to the increase in human needs and the applications of smart cities, the number of sensors within WSNs is constantly increasing, thus collecting larger amounts of data. Therefore, Smart City applications may suffer delays in work or downtime for some time. As a result of the loss or congestion of the large data accumulated on communication lines, which need speed in transmission, processing, and analysis.

Some of these data are related in time and place due to their physical nature, and they are in the form of time series that were collected chronologically or sequentially. Moreover, some of the temporal or spatial correlation readings that are collected do not bear exclusive information or can be extracted from previous readings. Therefore, preventing unnecessary data transfers in a network has a significant impact on reducing the energy consumption of devices and reducing the congestion and the cost of communications in the smart city (He et al. 2014).

Data prediction technique is one way to manage the vast amount of data in WSNs. It means predicting new data from the previous data collected from sensor nodes. However, one key concern is to ensure the accuracy of the prediction within a user-given error bound (Wu et al. 2016). Several studies, such as Azar et al. (2019), highlighted these issues to address them to reduce the congestion communication and increase the efficiency of the IoT by transferring the data in a correct and timely manner.

Dias et al. (2016) simulated a self-managed WSN network that used the AI technique. State-of-the-art prediction algorithm used in this simulation to reduce the data transmission. Wu et al. (2016) applied three techniques (prediction, compression, and recovering) in their proposed framework to reduce data transmission and communication costs using Least Mean Square (LMS) and LMS with Optimal Step Size (OSSLMS) models. Jarwan et al. (2019) applied the DP on their proposed framework of WSN to save the lifetime of the sensor and improving communication using NNs and LSTM models.

In short, our problem is almost similar to the previous studies which aim to reduce the data transmission, but we focus on IoT network in which an intelligent IoT is developed to reduce the transmission of unnecessary data. Therefore, this research aims to develop a sensor forecasting model using deep learning by only allowing data transmission greater than a threshold. This paper aims to propose three LSTM deep learning models (Stacked LSTM, Bi-Directional LSTM, and Convolutional

LSTM) using Malaysia Atmospheric dataset; and to evaluate the models and use the best one for IoT sensor prediction to reduce the data transmission.

This paper consists of five (5) sections. Section I discuss the background of this study. Section II reviews the literature review and related works. Section III introduces the method used in the study. Section IV presents the results of the work and discussion. Lastly, section V concludes the paper with a summary of the findings and recommended future work.

II. LITERATURE REVIEW

A. METHODS TO REDUCE DATA TRANSMISSIONS BASED ON WSN

Two methods, namely Data prediction and Data compression, have been used to reduce the data transmission in clustered WSNs. WSN is located within the first level of the IoT architecture. The Clustered WSN contains several normal sensor nodes and a head per cluster. The Cluster Head (CH) connects the normal sensor nodes with the gateway (GW) (Dias et al. 2016).

Data compression means reduce the size of data. In general, the data compression approach can classify into two main methods: lossless compression, which ensure information is correct during the compression and decompression process, and the lossy data compression algorithms, which lead to some loss of the original data after the decompression operation (Wu et al. 2016). Jarwan et al. (2019) and Wu et al. (2016) have implanted Principal Component Analysis (PCA) to compress and recover previously predicted data.

Data prediction means predicting future values based on previous historical data or inferring lost values in a data set by experimental probability or statistics. (Dias et al. 2016). Two prediction schemes were used to reduce the data transmission in WSNs. One is the Single Prediction Schemes (SPSs); the other is the Dual Prediction Schemes (DPSs) (Zhang et al. 2018). SPSs mean the predictions are made in a single point in WSN like sensor nodes or CHs. CHs can predict the data collected by sensor nodes. Sensor nodes will anticipate changes in their environments to prevent unnecessary measurements, thus avoid their transmissions. In SPSs, each device will determine by itself whether or not to use predictions. In DPSs, the prediction operations are made in two endpoints at the same time. For example, in the sensing node and its CH. The purpose of DPSs is to avoid unnecessary data transmissions. The same prediction algorithm is implemented on both endpoints. So, if the predicted data in the sensing node is of high accuracy and falls below the threshold value, the sensing node prevents the data transmission, and the predicted data on CH is used instead. Whereas, if the data predicted by algorithms fall outside the threshold value, the sensor node will read accurate data from the environment surrounding it and sent the data to CH. (Dias et al. 2016). Dias et al. (2016) applied state-of-the-art algorithms to perform Single Prediction Schemes and Dual Prediction Schemes to reduce data transmission and achieved acceptable results.

B. *TIME SERIES PREDICTION METHODS*

Different models have been used to conduct a time series prediction, like Auto-Regressive Integrated Moving Average (ARIMA), machine and deep learning models. Siami-Namini et al. (2018) have introduced and tested ARIMA and Long Short Term Memory (LSTM) on a collection of financial data. The results were revealing that the LSTM-based algorithm outperformed ARIMA by 85% on average. So, LSTM demonstrated superiority over ARIMA. Similarly, Zhang et al. (2018) proved that LSTM based method produced higher prediction accuracy comparing with ARIMA in terms of time series analysis and forecasting. In the same manner, Kelany et al. (2020) used LSTM, Logistic Regression, and Random Forest to predict future prices stocks. The results showed that the LSTM model is better than other traditional techniques for all stock categories across various time periods. Furthermore, Thai-Nghe et al. (2020) used Simple LSTM and Stack LSTM. The results using RMSE showed that the Stack LSTM model achieved better results compared to the simple model and the LSTM models showed better results than the SVM baseline regression model.

C. *RELATED WORK ON IOT DATA TRANSMISSION BASED ON WSNS*

(Wu et al. 2016) applied three techniques (prediction, compression, and recovering) to reduce data transmission and communication costs in their proposed framework while guaranteeing data prediction and processing accuracy in clustered WSNs. Two dual prediction algorithms were used to predict the data in the sensor node and CH. One is the LMS, and the second is LMS with OSSLMS that minimizing the mean-square derivation (MSD). After that, a centralized PCA technique was used to implement the compression and recovery for the predicted data on the CHs and the Sink in the Clustered WSN. Thus, prevent the spatial redundancy of the sensed data and reduce the communication cost.

State-of-the-art algorithms were applied for predictions to perform SPSs and DPSs. In an improved WSN structure which integrated artificial intelligence technologies. A Reinforcement Learning technique called Q-Learning technology was used in the study semulation. The results reduce data transmissions and improving communication within the WSNs networks by 92%, assuming the evolution of the sensor specifications and keeping data quality (Dias et al. 2016). However, this study did not provide sufficient details about the simulation process, the data used, or the prediction algorithms.

A based on bidirectional LSTM called Multi-Node Multi-Feature (MNMF) was presented by Cheng, Xie, Wu, et al. (2019) to eliminate unnecessary data transfer within the WSN. Temperature, Humidity, light, voltage, date, time, timestamp, and node ID are included among the 2.3 million pieces of sensory data collected from 54 nodes in the collection used in this study. The proposed model was compared to three neural network prediction models: Elman network, a recurrent neural network with local memory units and local feedback connections, GRNN (general regression neural network), and NARX (nonlinear autoregressive exogenous model). The proposed model outperformed the Elman, GRNN, and NARX models on the MAPE measurement, with an error rate of 0.318 compared to 0.698 for Elman, 0.328 for GRNN, and 1.17 for NARX. However, The prediction approach used in this paper depended on multiple nodes and multip features. They used the models to analyze data from Intel Berkeley Research Laboratory's defined distributed WSN network, which comprises 54 sensor nodes.

(Jarwan et al. 2019), tried to prevent unnecessary data transfer to save energy and bandwidth in WSNs. To achieve their goal, the researchers applied Dual Prediction and Data Compression schemes on a clustered WSN. OSSLMS, Long Short-Term Memory networks (LSTMs), and Neural Networks (NNs) models were used as Time-series prediction algorithms to implement the prediction scheme. Readings for temperature values with 30-second intervals were included in the data set used in this experiment. The simulations were run on the first 600 temperature values obtained. The result revealed that both NNs and LSTMs perform better than the OSSLMS algorithm regarding Transmission Reduction Percentage (TRP). TRP means how many data point transmissions were not carried because they were accurately predicted. the TRP for the total nodes in 54 WSN are as following: LSTM =51:2%, for NNs=51:1%, and 48:3% for OSSLMS. Whereas the average MSE is 0.00372 for all three algorithms. Moreover, the study proved that deep learning models perform better than OSSLMS in prediction processes. However, since the dual predictions relied on the correlation of data, the performance of these models may vary from one environment to another. On the other hand, the hyperparameters of the chosen model significantly affect the model's performance.

Briefly, most of the studies reviewed above have used different methods to reduce data transmission using time series prediction in WSNs. So, these studies were restricted to their WSNs frameworks.

III. METHOD

The research design that has been followed in this paper is provided in this section. By applying these steps, the study objectives have been achieved. The main stages of this research approach that have been taken are shown below:

- Stage 1: Preparing datasets.
  - Dataset Description.
  - Datasets Preprocessing.
- Stage 2: Implementation.
- Stage 3: Evaluation and Comparison Results.
- Stage 4: Data Transmission Reduction.

A. Preparing datasets

i) Dataset Description

The dataset that used in this paper is Malaysia Atmospheric Dataset for the year 2016. It is time-series data that represented hourly sensor readings. It is a cleaned data. The dataset contains data for three stations representing three places: Putrajaya, Petaling Jaya, and Tanjung Malim. Each station includes five attributes of hourly sensor readings, which are: Temperature, Humidity, Wind Speed, Nitrogen Oxides (NOx), and Nitrogen Dioxide (NO2). The total number of instances for each attribute in all stations is (8784).

ii) Dataset Preprocessing

Preprocessing data is a data mining technique for transforming raw data into a usable and efficient format. Data normalization, splitting the dataset, and dataset restructuring were applied to prepare the dataset forecasting processing in this study.

a. Dataset Normalization

Data normalization is one of the preprocessing approaches. The main purpose of data normalization is to ensure the quality of the data before it is given to any learning algorithm. Data normalization can be used to minimize bias within the learning models. Data normalization can also speed up training time by starting the training process within the same data scale. In this study, the data was scaled in a range between 0 and 1 (Brownlee 2017).

b. Splitting Dataset

In this paper, the station's datasets are divided into 80% as a training set and 20% as a testing set. The total instance number of the train data for each station attribute is (7027) and (1756) for testing.

c. Restructure the dataset

The time-series data is arranged in a connected and sequential manner. This study will use univariate time series forecasting as a supervised learning issue in this research. Therefore, to implement deep learning and machine learning algorithms, the data must be reframed using the lag method or sliding window. Therefore, the data will be reorganized to present sequential reads as the input x and the next read as output y. Then arrange the data again, in order, the value y will be as a part of the input x sequentially, and the next value will be as the output y, as shown in the following Figure 1 (Kotriwala et al. 2018).

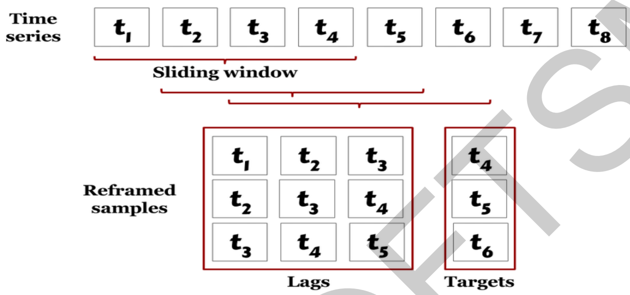


Figure 1 Restructured the Datasets

B. Implementation

Three models based on LSTM: Stacked LSTM, Bi-directional LSTM, and Convolutional LSTM, were used due to the accuracy of these models that was achieved in (Shastri et al. 2020) study. Three stations Putrajaya, Petaling Jaya, and Tanjung Malim datasets, have been used in the experiment. Each model applied on all the five attributes one by one for each station dataset.

i) Stacked LSTM

Stack LSTM is multiple LSTM layers that are fully connected structure. As several LSTM layers are combined, this leads to greater model complexity and increased model depth. Each intermediate LSTM layer generates sequential vectors, which are fed into the next LSTM layer, as shown in Figure 2.

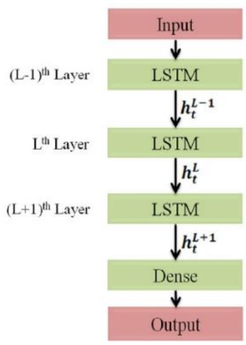


Figure **Error! No text of specified style in document.** Stacked LSTM

ii) Bi-directional LSTM (BiD LSTM)

Traditional RNNs have a limitation. They only process information in one direction and pay no attention to future processed data. The notion of a Bi-directional RNN was proposed by (Bidirectional Recurrent Neural Networks). Bi-directional RNN can use different hidden layers as forwarding and backward layers, to process information in both directions states simultaneously. Bi-directional LSTM combines Bi-directional RNN and LSTM cells. Bi-directional LSTM was presented by (Graves et al. 2005). The Bi-directional LSTM structure was shown in Figure 3.

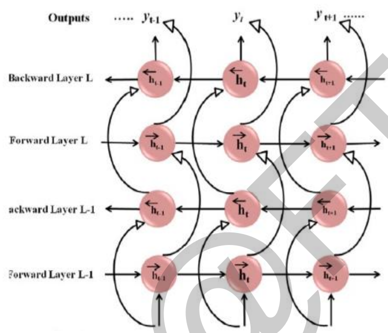


Figure 3 Bi-directional LSTM

iii) Convolutional LSTM (ConvLSTM)

One of the most popular deep neural networks is CNN. It gets its name from the linear mathematical process between matrixes called convolution. Shi et al. (2015) proposed ConvLSTM as a type of RNN that performs CNN convolutions as part of the LSTM for each step. The ConvLSTM uses convolutional structures in both the inputs and past states of its local neighbors to predict the future state of a cell in the grid see Figure 4

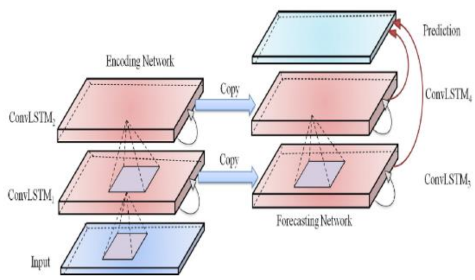


Figure 4 Convolutional LSTM

Moreover, depending on Shastri et al. (2020), the three models used some of the same configuration parameters and differed in other parameters according to the structure for each model.

All models used 100 neurons in each layer, ReLu activation function, 40 epochs, 0.2 for validation split, Adam used as an optimizer, Mean Square Error as loss function to evaluate the model, and the same three sizes of the input data. In contrast, The models differed in the following parameters, as illustrated in Table 1.

Table 1 Parameters of The LSTM Models

Stack LSTM	Bi-directional LSTM	Convolutional LSTM
▪ Two layers of Stack LSTM were used. ▪ Return sequences = true. ▪ The verbose = 1	▪ A single hidden layer was used. ▪ The verbose = 2	▪ A single hidden layer was used. ▪ 64 filters ▪ kernel size as (1, 2) ▪ The verbose = 2

Additionally, this experiment will implement ten times for each model with each different input data among each station data to compare the performance of the models and choose the best one. The best model will determine by comparing the average of (MAPE) values and the accuracy calculated. The experiments will be carried out using Spyder python 3.8.5 with open source libraries like Tensorflow. Keras version 2.4.0, Pandas, and Numpy. The experimental setup is based on a Laptop computer having Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz with 8.00 GB under 64-bit Windows 10 Pro Operating system version 20H2.

C. Evaluation

1) Walk Forward Validation Strategy

The traditional approaches used in machine learning, such as k-fold cross-validation, are ineffective with time-series data. Instead of that, a walk-forward validation strategy can help solve this problem. The Walk-forward validation is a strategy in which the prediction is performed using expanding window methods. The size of the forward window is determined sequentially dependent on the sampling frequency of the time series. Training and validation sets are progressively time-shifted to integrate recent observations after each time step prediction. Then renew the forecast window (depending on the horizon) after each iteration, as shown in Figure 5 (Suradhaniwar et al. 2021)

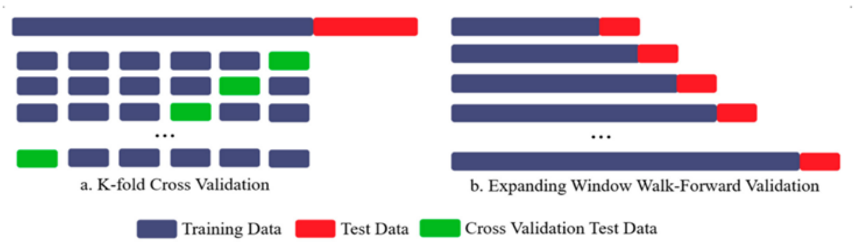


Figure 5 Walk Forward Validation vs K-fold Validation

2) Evaluation Metrics

Tow metrics have been selected, which are Mean Absolute Percentage Error and Root Mean Squared Error to provide a comparison of performance between the models used in this paper.

i) Mean Absolute Percentage Error (MAPE)

MAPE is one of the most popular metrics for evaluating forecasting performance. MAPE is expressed as a percentage. It also benefits from being scale-independent, making it useful for comparing forecast performance across different datasets (Martínez-Álvarez et al. 2015). The following formula gives it.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

ii) Root Mean Squared Error (RMSE)

The RMSE is a metric for comparing estimated and measured values. RMSE is always positive by definition, and a lower value indicates greater accuracy. The value of RMSE benefits from being scale-dependent; consequently, it is appropriate for comparing different models for the same dataset but not for different datasets (Jiang et al. 2020). The following formula gives it.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2}$$

D. Data Transmission Reduction

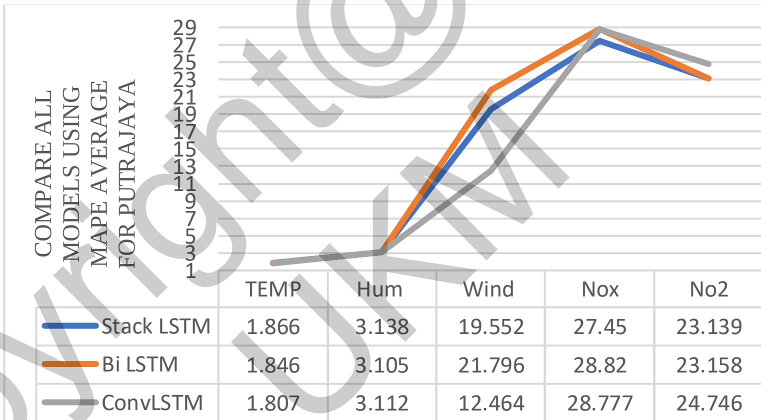
Three important points are used for data transmission reduction. The first one is the best model which is produced from the evaluation stage. The second point is the best accuracy from the ten times implementations based on MAPE measurement. The last point is the value of a threshold. The threshold means the maximum acceptable error (Forecasting Error) that defines how accurate the data is needed (Jarwan et al. 2019). In this paper, 0.5 was chosen as a threshold for Temperature as (Jarwan et al. 2019), Humidity, and Wind Speed. Whereas 0.005 was chosen as a threshold for the NOx and NO2 because the range of their values is small between (0 to 0.94). Based on those three points, the number of accurately predicted values will be calculated by using the Transmission Reduction Percentage (TRP). The TRP means the percentage of data stated under the threshold, so there is no need to transfer it (Jarwan et al. 2019), as shown in this formula :TRP = (Accurate Data / Total Number of Raws) * 100

IV. RESULTS & DISCUSSION

A. EXPERIMENT RESULTS TO CHOOSE THE BEST MODEL

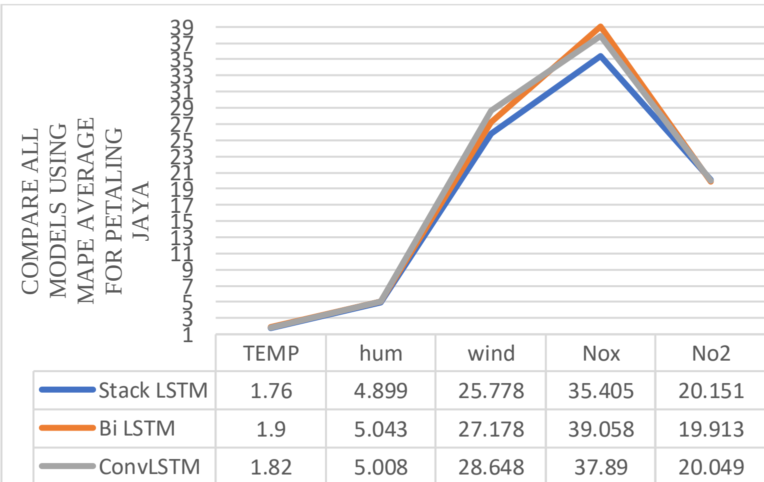
The experiment has applied using Python programming language. The total number of experiments was ten times of implementation for the model on each attribute in each station dataset.

Figure 6 shows the results of the performance of the three models is competitive on all attributes except the Wind Speed attributes with the ConvLSTM model. ConvLSTM was dealing better in Wind Speed with the smallest MAPE = 12.4, compared with the other models, which have 19.5 MAPE for Stack LSTM and 21.7 for BiD LSTM. Figure 7 shows the results of the implementation of all the models are competitive performance on Temperature, Humidity, and NO2; however, it is clear the Stack LSTM is the best model performance with the smallest MAPE = 25.7 on Wind Speed and 35.4 for Nox.



	TEMP	Hum	Wind	Nox	No2
Stack LSTM	1.866	3.138	19.552	27.45	23.139
Bi LSTM	1.846	3.105	21.796	28.82	23.158
ConvLSTM	1.807	3.112	12.464	28.777	24.746

Figure 6 Putrajaya Results



	TEMP	hum	wind	Nox	No2
Stack LSTM	1.76	4.899	25.778	35.405	20.151
Bi LSTM	1.9	5.043	27.178	39.058	19.913
ConvLSTM	1.82	5.008	28.648	37.89	20.049

Figure 7 Petaling Jaya Result

Figure 8 shows that the performance results for the three models is closed to each other on all attributes except the with Wind Speed the ConvLSTM is clearly has the best MAPE =16.0 as shown in an ellipse. Furthermore, the ConvLSTM was clearly dealing better on Putrajaya and Tanjung Malim, as shown in Figures 6 and 8. So, we can choose the ConvLSTM to calculate the data transmission reduction amount.

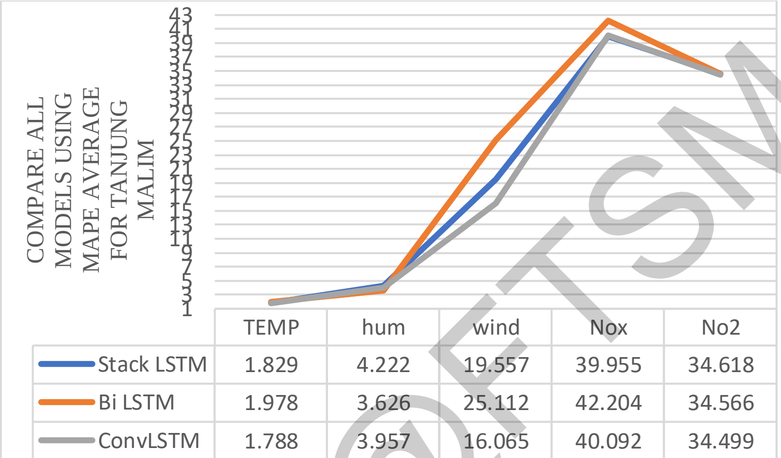


Figure 8 Tanjung Malim Result

B. *EXPERIMENT RESULTS OF THE DATA TRANSMISSION REDUCTION*

Data transmission reduction means reduce data transfer between two endpoints in the IoT network by applying the proposed model in terms of Single or Dual Prediction techniques as explained in the previous sections. The first experiment was performed on the Putrajaya. The outcomes are stated in Table 2 below.

Table 2 Putrajaya Data Transmission Reductions Results

Column Name	**Total Number of Rows**	**Data Under the Threshold**	**Data Over the Threshold**	**TRP**	**MAPE**	**Accuracy**	**RMSE**	**Threshold**
Temp	1756	1220	536	69.5%	1.75%	98.24%	0.846	0.5
Humidity	1756	348	1408	19.8%	2.996%	97.003%	3.683	0.5
WindSpeed	1756	1359	397	77.4%	13.058%	86.941%	0.562	0.5
NOx	1756	929	827	52.9%	27.771%	72.228%	0.0095	0.005
NO2	1756	1284	472	73.1%	23.525%	76.474%	0.00527	0.005

Table 2 shows the data transmission reduction for the Putrajaya station dataset. The table consists of eight columns. The first one is the total number of raws (1756) used in the forecasting to calculate the data transmission reduction. The second column shows how much data will be prevented from sending from the sensor to the gateway because it is under the threshold so, it is accurate. The total number of the data under the threshold for the Temperature, Humidity, Wind Speed, NOx, and NO2 are 1220, 348, 1359, 929, and 1228. In contrast, column three represents the data that is over the threshold. So, it should send from the sensor to the gateway as the actual reading of the sensor, not the predicted data. The total number of the data over the threshold for the Temperature, Humidity, Wind Speed, NOx, and NO2 are 536, 1408, 397, 827, and 472. Column four shows the TRP. TRP is the percentage of data that were not carried to transfer because they were accurately predicted. It is clearly noticed that the TRP that was achieved for the attributes shows as 69.5% for the Temperature, 19.8% for Humidity, 77.4% for Wind Speed, 52.9% for NOx, and 73.1 for NO2. Column five shows the MAPE for the best iteration of the ten implementations. The MAPE for the Temperature, Humidity, Wind Speed, NOx, and NO2 are 1.75%, 2.99%, 13.05%, 27.77%, and 23.5%. However, the next column is the accuracy calculated based on the MAPE as shown in the formula below. It shows the accuracy for the Temperature, Humidity, Wind Speed, NOx, and NO2 as 98.24%, 97.00%, 86.94%, 72.22%, and 76.47%.

Accuracy = 100 – MAPE

Column seven is RMSE measurement. It shows prediction error for the Temperature, Humidity, Wind Speed, NOx, and NO2 as 0.846, 3.683, 0.562, 0.0095, and 0.00527. Finally, the last column represents the threshold. In this study, the threshold means the maximum difference between the actual and prediction data (Forecasting Error). The threshold value is 0.5 for Temperature, Humidity, and Wind Speed; however, 0.005 for NOx and NO2. The threshold value affects the TRP. For example, with Humidity data, the MAPE of it is 2.9; thus, the accuracy is about 97%. Whereas the TRP is 19.8%, this is because the threshold seems not sensible regarding the nature of the data and the range of the sequential of the Humidity data. So, the threshold value needs the understanding of the user who knows the nature of his data to put the sensible threshold for his application.

Similar to Table 2 for the data transmission on Putrajaya, Table 3 shows Petaling Jaya data transmission reduction results, and Table 4 shows Tanjung Malim data transmission reduction results. In addition, there is an issue that needs to explain when comparing the three tables 2, 3, and 4. Figures 9 and 10 are showing it. Figure 9 shows that Putrajaya is the best one depending on the accuracy 98.24% for the Temperature, 97.00% for Humidity, 86.94% for Wind Speed, 72.23% NOx, and 76.47% for NO2. Then Tanjung Malim accuracy 98.19% for the Temperature, 96.28% for Humidity, 83.97% for Wind Speed, 62.99% NOx, and 69.04% for NO2. Finally, Petaling Jaya accuracy 98.20% for the

Temperature, 95.14% for Humidity, 74.87% for Wind Speed, 66.00% NOx, and 69.24% for NO2. In comparison, Tanjung Malim is clearly better than Petaling Jaya on Humidity and Wind Speed only.

Table 3 Petaling Jaya Data Transmission Reduction Results

Column Name	Total Number of Rows	Data Under the Threshold	Data Over the Threshold	TRP	MAPE	Accuracy	RMSE	Threshold
Temp	1756	1182	574	67.31%	1.79%	98.2%	0.79	0.5
Humidity	1756	192	1564	10.9%	4.855%	95.14%	5.25	0.5
WindSpeed	1756	684	1072	38.95%	25.12%	74.87%	1.25	0.5
NOx	1756	384	1372	21.86%	33.9%	66.0%	0.023	0.005
NO2	1756	1216	540	69.24%	19.47%	80.52%	0.0061	0.005

Table 4 Tanjung Malim Data Transmission Reduction Results

Column Name	Total Number of Rows	Data Under the Threshold	Data Over the Threshold	TRP	MAPE	Accuracy	RMSE	Threshold
Temp	1756	1238	518	70.5%	1.8%	98.19%	0.83	0.5
Humidity	1756	220	1536	12.5%	3.720%	96.279%	4.138	0.5
WindSpeed	1756	1292	464	73.57%	16.026%	83.97%	0.681	0.5
NOx	1756	1310	446	74.6%	37.00%	62.99%	0.00616	0.005
NO2	1756	1637	119	93.22%	30.956%	69.04%	0.00269	0.005

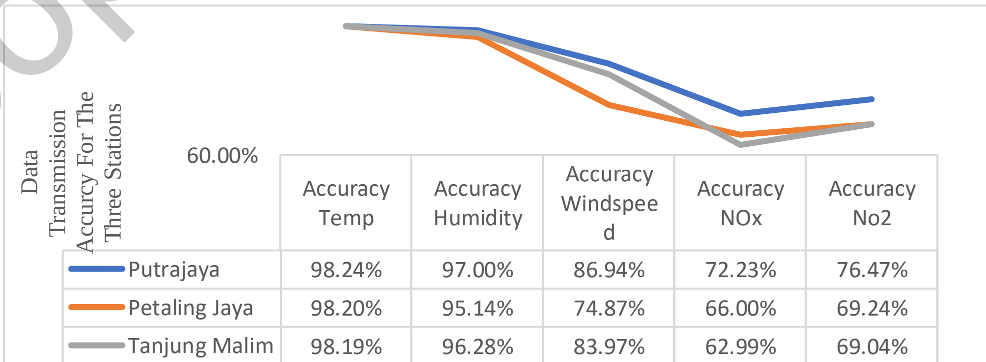
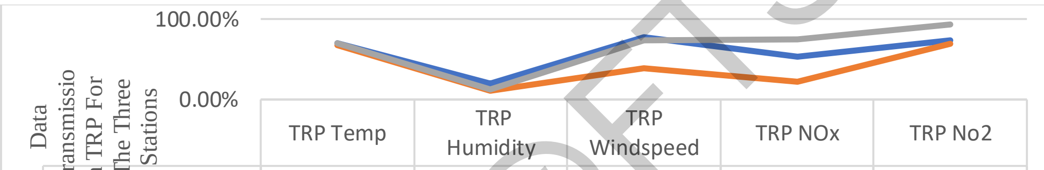


Figure 9 Data Transmission Accuracy for The Three Stations

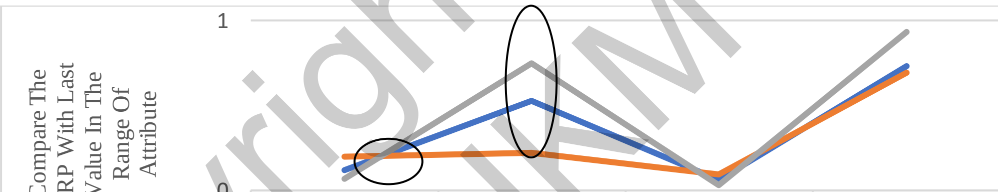
Figure 10 shows that the TRP of Putrajaya is the best one on Temperature 69.50%, Humidity 19.80%, and 77.40% for Wind Speed, then the TRP of Tanjung Malim Temperature 70.50%, Humidity 12.50%, and 73.5% for Wind Speed. Then TRP of Petaling Jaya Temperature 67.31%, Humidity

10.90%, and 38.95% for Wind Speed, this seems similar to the accuracy. However, the TRP with NOx and NO2 dealing inconsistent with the accuracy; Tanjung Malim has the best TRP 74.60% for NOx and 93.22% for NO2 instead of Putrajaya and Petaling Jaya. Putrajaya has 52.90% for NOx and 73.10% for NO2, and Petaling Jaya has 21.86% for NOx and 69.24% for NO2. The reason behind that is the range of data and the value of the threshold. When the last range value is small, like NO2 = (0.31) in Tanjung Malim, this caused the most predicted value to be under the threshold, so the TRP is big (93.22%), as shown in Figure 11. Whereas, last range value for Putrajaya NO2 is 0.062, and its TRP is 73.10%. Also, in the same way, the last range value for Petaling Jaya NO2 is the biggest rang, 0.094, and its TRP is 69.24%. Similarly, with NOx. For this reason, it is important to understand the nature of data to put the sensible threshold.



	TRP Temp	TRP Humidity	TRP Windspeed	TRP NOx	TRP No2
Putrajaya	69.50%	19.80%	77.40%	52.90%	73.10%
Petaling Jaya	67.31%	10.90%	38.95%	21.86%	69.24%
Tanjung Malim	70.50%	12.50%	73.57%	74.60%	93.22%

Figure 10 Data Transmission TRP for The Three Stations



	NOx last value	TRP NOx	NO2 last value	TRP NO2
Putrajaya	0.122	52.90%	0.062	73.10%
Petaling Jaya	0.201	21.86%	0.094	69.24%
Tanjung Malim	0.069	74.60%	0.031	93.22%

Figure 11 Compare The TRP With Last Value In The Range Of Attribute

To sum up, Putrajaya data produced the best accuracy of the data transmission reduction. The TRP value and prediction accuracy are affected by the threshold value and the range of data.

V. CONCLUSION

Overload communication in the IoT network caused problems of delay in Smart city-based IoT applications. The continuously capturing data from sensors into the IoT network are generating massive data. The data collected often has some spatial or temporal redundancy and is unnecessary, which can be eliminated. One popular approach for this issue is data transmission reduction. Some researchers used Machine and Deep Learning techniques to reduce data transmission in the WSNs by applied a

Dual Prediction (DP) method using sensors data. Therefore, this study aims to reduce data transmission for an atmospheric IoT application using deep learning. The experiments were conducted on three stations Malaysia Atmospheric datasets: Putrajaya, Petaling Jaya, and Tanjung Malim, in which each station consists of five attributes: Temperature, Humidity, Wind Speed, NO2, and NOx. Three LSTM algorithms are used to investigate the best prediction model for each attribute: Stack LSTM, BiD LSTM, and ConvLSTM. The result shows ConvLSTM algorithm has consistently shown the best model on Putrajaya and Tanjung Malim datasets. Later, the ConvLSTM prediction model has been used to forecast data for all three stations. The experiment results showed that Putrajaya has the best accuracy result; it can be concluded that deep learning has contributed to reducing IoT data transmission from up to 77.40%. In the future, it is recommended to re-apply this study using other models, use several years of data from the Malaysia Atmospheric dataset. Re-test the data transmission reduction with different thresholds. Re-implement the models were used in this study on a new dataset or to solve another time series forecasting problem. Finally, re-apply this study on hardware.

ACKNOWLEDGEMENT

The authors would like to thanks, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia and University of Malaya by giving the authors an opportunity to conduct this research.

REFERENCE

Azar, J., Makhoul, A., Barhamgi, M. & Couturier, R. J. F. G. C. S. 2019. An Energy Efficient Iot Data Compression Approach for Edge Machine Learning. 96(168-175.

Brownlee, J. 2017. *Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future.* Machine Learning Mastery.

Cheng, H., Xie, Z., Wu, L., Yu, Z., Li, R. J. E. J. O. W. C. & Networking. 2019. Data Prediction Model in Wireless Sensor Networks Based on Bidirectional Lstm. 2019(1): 1-12.

Dias, G. M., Bellalta, B. & Oechsner, S. 2016. Using Data Prediction Techniques to Reduce Data Transmissions in the Iot. 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), hlm. 331-335.

Dias, G. M., Bellalta, B. & Oechsner, S. J. a. C. S. 2016. A Survey About Prediction-Based Data Reduction in Wireless Sensor Networks. 49(3): 1-35.

Graves, A. & Schmidhuber, J. J. N. N. 2005. Framewise Phoneme Classification with Bidirectional Lstm and Other Neural Network Architectures. 18(5-6): 602-610.

He, B., Li, Y., Huang, H., Tang, H. J. K. & Systems, I. 2014. Spatial–Temporal Compression and Recovery in a Wireless Sensor Network in an Underground Tunnel Environment. 41(2): 449-465.

Jarwan, A., Sabbah, A. & Ibnkahla, M. J. I. J. O. S. a. I. C. 2019. Data Transmission Reduction

Schemes in Wsns for Efficient Iot Systems. 37(6): 1307-1324.

Jiang, H., Lu, N., Huang, G., Yao, L., Qin, J. & Liu, H. J. a. E. 2020. Spatial Scale Effects on Retrieval Accuracy of Surface Solar Radiation Using Satellite Data. 270(115178.

Kelany, O., Aly, S. & Ismail, M. A. 2020. Deep Learning Model for Financial Time Series Prediction. 2020 14th International Conference on Innovations in Information Technology (IIT), hlm. 120-125.

Kotriwala, A. M., Hernandez-Leal, P. & Kaisers, M. 2018. Load Classification and Forecasting for Temporary Power Installations. 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), hlm. 1-6.

Martínez-Álvarez, F., Troncoso, A., Asencio-Cortés, G. & Riquelme, J. C. J. E. 2015. A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting. 8(11): 13162-13193.

Shastri, S., Singh, K., Kumar, S., Kour, P., Mansotra, V. J. C., Solitons & Fractals. 2020. Time Series Forecasting of Covid-19 Using Deep Learning Models: India-USA Comparative Case Study. 140(110227.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. & Woo, W.-C. J. a. P. A. 2015. Convolutional Lstm Network: A Machine Learning Approach for Precipitation Nowcasting.

Siami-Namini, S., Tavakoli, N. & Namin, A. S. 2018. A Comparison of Arima and Lstm in Forecasting Time Series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), hlm. 1394-1401.

Suradhaniwar, S., Kar, S., Durbha, S. S. & Jagarlapudi, A. J. S. 2021. Time Series Forecasting of Univariate Agrometeorological Data: A Comparative Performance Evaluation Via One-Step and Multi-Step Ahead Forecasting Strategies. 21(7): 2430.

Thai-Nghe, N., Thanh-Hai, N., Chi Ngon, N. J. I. J. O. a. C. S. & Applications. 2020. Deep Learning Approach for Forecasting Water Quality in Iot Systems. 11(8): 686-693.

Wu, M., Tan, L. & Xiong, N. J. I. S. 2016. Data Prediction, Compression, and Recovery in Clustered Wireless Sensor Networks for Environmental Monitoring Applications. 329(800-818.

Zhang, W., Guo, W., Liu, X., Liu, Y., Zhou, J., Li, B., Lu, Q. & Yang, S. J. I. A. 2018. Lstm-Based Analysis of Industrial Iot Equipment. 6(23551-23560.