A DEEP LEARNING MODEL FOR REDUCING DATA TRANSMISSION IN IOT NETWORK

ZOHAA MUBARAK ABDULLAH AL-QFAIL

UNIVERSITI KEBANGSAAN MALAYSIA

A DEEP LEARNING MODEL FOR REDUCING DATA TRANSMISSION IN IOT NETWORK

ZOHAA MUBARAK ABDULLAH AL-QFAIL

PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF COMPUTER SCIENCE (ARTIFICIAL INTELLIGENCE)

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2021

MODEL PEMBELAJARAN MENDALAM UNTUK MENGURANGKAN PENGHANTARAN DATA DALAM RANGKAIAN IOT

ZOHAA MUBARAK ABDULLAH AL-QFAIL

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN
DARIPADA SYARAT MEMPEROLEHI IJAZAH
SARJANA SAINS KOMPUTER (KECERDASAN BUATAN)

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI
2021

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

12 August 2021

ZOHAA MUBARAK
ABDULLAH AL-QFAIL
GP07230

ACKNOWLEDGEMENT

First and foremost praise be to Almighty Allah for all his blessings for giving me patience and good health throughout the duration of this master research.

Secondly, I am grateful to this great country, Malaysia, for giving me the chance to study under its umbrella.

Moreover, I am very fortunate to have Associate Professor Dr. Zulaiha Ali Othman as a research supervisor. Thank you for her advice, support, and encouragement.

Also, I would like to express my high appreciation to the coursework coordinators, Dr. Kamsuria Ahmed and Dr. Dahlila Putri Dahnil Sikumbang.

Moreover, I am grateful to all my professors at FTSM for helping me to gain knowledge. Thanks to all FTSM and UKM staff,

I would like to thanks My dearest husband, all family, and all friends,

Last but not least, I gratefully acknowledge the financial support provided by Hadhramout University, Seiyun University, and Hadhramout Foundation, Yemen, that they helped me to get a scholarship to study abroad and complete my higher studies,

ABSTRAK

Lebihan komunikasi dalam Rangkaian IoT adalah satu daripada punca kegagalan perlaksanaan bandar pintar berasaskan IoT. Data yang diambil secara berterusan dari sensor ke rangkaian IoT menghasilkan data yang besar. Data yang dikumpulkan sering mempunyai beberapa ruang atau temporal berlebihan dan tidak perlu, yang patut dihilangkan. Dua pendekatan popular untuk masalah ini ialah: pemampatan data dan mengurangkan penghantaran data. Kajian sebelumnya telah mencadangkan kaedah Dual Prediction (DP) dalam korelasi Spatio-temporal pada data sensor di Rangkaian Sensor Tanpa Wayar (WSN) untuk mengurangkan penghantaran data menggunakan teknik Mesin dan Pembelajaran Dalam. Oleh itu, kajian ini bertujuan untuk mengurangkan penghantaran data untuk aplikasi IoT atmosfera menggunakan pembelajaran mendalam. Eksperimen tersebut dilakukan di tiga stesen data Atmosfera Malaysia: Putrajaya, Petaling Jaya, dan Tanjung Malim, di mana setiap stesen terdiri daripada lima atribut: suhu, kelembapan, kelajuan angin, NO2, dan NOx. Tiga algoritma LSTM digunakan untuk menyelidiki model ramalan terbaik untuk setiap atribut, iaitu Stack LSTM, LSTM Bi-directional (BiD LSTM), dan LSTM Konvolusional (ConvLSTM). Eksperimen ini menggunakan kaedah pengesahan berjalan kedepan sepuluh kali untuk setiap atribut menggunakan pengukuran MAPE. Hasilnya menunjukkan algoritma ConvLSTM secara konsisten menunjukkan model terbaik di dataset Putrajaya dan Tanjung Malim. Kemudian, model ramalan ConvLSTM telah digunakan untuk meramalkan data untuk ketiga-tiga stesen. Hasil eksperimen menunjukkan bahawa ConvLSTM telah menurunkan data transmisi suhu untuk Putra Jaya, Petaling Jaya, dan Tanjung Malim masing-masing sekitar 69.5%, 67.31%, dan 70.5%. Ia juga mengurangkan Kelajuan Angin 77.4%, 38.95% dan 73.57% dan Kelembapan 19.8%, 10.9% dan 12.5% masing-masing dengan ambang 0.5. Selanjutnya, ia mengurangkan NOx sekitar 52.9%, 21.86%, 74.6%, dan mengurangkan NO2 dan 73.1%, 69.24%, dan 93.22% untuk Putra Jaya, Petaling Jaya, dan Tanjung Malim masing-masing dengan ambang 0,005. Berdasarkan hasil Putrajaya, dapat disimpulkan bahawa pembelajaran mendalam telah menyumbang untuk mengurangkan penghantaran data IoT dari 77.40%.

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%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
ARIMA	Auto-Regressive Integrated Moving Average
BiD LSTM	Bi-directional LSTM
CH	Cluster Head
CNN	Convolution Neural Network
ConvLSTM	Convolutional LSTM
DC	Data Compression
DP	Dual prediction
DPSs	Dual Prediction Schemes
GW	Gateway
IoT	Internet of Things
LMS	Least Medium Square
LSTM	Long Short Term Memory
MAPE	Mean Absolute Percentage Error
ML	Machine learning
MSD	Mean-Square Derivation
NNs	Neural Networks
OSSLMS	LMS Optimal Step Size
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SPSs	Single Prediction Schemes
TRP	Transmission Reduction Percentage
T-SVD	Truncated-Singular Value Decomposition
WSNs	Wireless Sensor Networks

CHAPTER I

INTRODUCTION

1.1 RESEARCH BACKGROUND

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). 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 collected temporal or spatial correlation readings 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).

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.

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).

There are still needs in this field to develop methods to reduce data transmission thus, reducing congestion of the IoT communications. Therefore, this study will reduce the data transmission between the sensor node (as a sender) and the gateway (as a receiver) by using deep learning models to forecast univariate time series on Malaysia Atmospheric dataset.

1.2 PROBLEM STATEMENT

Many studies handled a vast amount of data produced by sensor nodes in the WSNs by using Dual prediction (DP) or Data compression (DC) schemes to reduce the data transmission and enhance communications.

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) proposed a new WSNs framework with a custom combination. Three techniques: data prediction, compression, and recovery, were simulated in this study. OSSLMS, LMS models were used for prediction, and Principal Component Analysis (PCA) for compression to reduce communication cost. In the same manner, Jarwan et al. (2019), Applied the DP and DC on their proposed framework of WSN to save the lifetime of the sensor and improving communication using NNs and LSTM models for prediction and four DC schemes. However, all of these studies were implanted under the restriction of their WSNs framework.

On the other hand, many traditional models are used in univariate and multivariate time series forecasting in different fields, but they cannot predict a high number of observations as the Deep learning univariate and multivariate time series forecasting models (Tiwari et al. 2020).

For instance, Zhang et al. (2018), Siami-Namini et al. (2018), Jarwan et al. (2019), and Kelany et al. (2020) proved that LSTM models outperformed ARIMA, Logistic Regression, and Random Forest, Neural Networks (NNs), and SVM models in time series prediction.

In short, our problem is almost similar to the previous studies, which aim to reduce data transmission, but this work focuses on IoT networks 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.

1.3 RESEARCH QUESTIONS

The main questions of this study are:

- Which is the best performing model out of Stack LSTM, Bi-directional LSTM, and Convolutional LSTM when they applied on Malaysia Atmospheric dataset?
- How can the best model contribute to reducing the data transmissions using Malaysia Atmospheric dataset?

1.4 RESEARCH OBJECTIVES

The main objectives of this research are:

- To propose three LSTM deep learning models (Stacked LSTM, Bi-Directional LSTM, and Convolutional LSTM) to predict the IoT sensors' of Malaysian atmospheric.
- To evaluate the performance of the best prediction model reduced the IoT data transmission.

1.5 RESEARCH SCOPE

This study focuses on an experiment on reducing data transmission in IoT networks by implementing deep learning univariate time series forecasting model on three stations Malaysia Atmospheric Dataset to reduce the data transmission.

1.6 SIGNIFICANCE OF STUDY

With the increase in data generation from smart city applications and the Internet of things, it has become imperative to find solutions to manage and address the congestion problems on communication channels on the IoT. In order to avoid the problems of delay in the arrival of the critical and necessary data on time, and thus maintain the performance of applications and take appropriate decisions. In contrast, many studies have applied many traditional, machine, and deep learning models to reduce data transfers; but these studies applied regarding the WSNs.

Hence, it is a great opportunity to reduce data transfers and improve communications on the Internet of things by applying data prediction processing using deep learning models on the IoT sensor node and the gateway with a prior dataset.

1.7 METHODOLOGY

The methodology of this study is constituted of four stages. The first stage prepares the datasets. The second stage aims to apply three deep learning models based on LSTM used in Shastri et al. (2020) using Putrajaya Station, Petaling Jaya, and Tanjung Malim station in Malaysia Atmospheric Dataset and choose the best model. The third stage aims to evaluate the models on data of three stations to choose the best model. The last stage aims to calculate the data transmission reduction. The phases are discussed in more detail in Chapter III.

1.8 PROJECT REPORT ORGANIZATION

This project has five chapters, summarized as follows:

Chapter I contains the basics of the research. These include a background of the study, problem statement, research questions, research objectives, research scope, significance, and the methodology of the study. This chapter also gives the organization of the thesis.

Chapter II introduces a glance at the IoT and WSNs structure and presents the related work to reduce data transmission. It reviews the studies and projects that have been conducted in the past about this topic with providing the strengths and limitations of these studies.

Chapter III discusses the research methodology that will answer the research questions and objectives of the research mentioned in Chapter I. This research methodology contains datasets profile and deep learning models used in our study to achieve the research objectives described earlier.

Chapter IV presents the results and findings obtained.

Finally, Chapter V provides the conclusion of the research. The chapter concludes the work carried out in the study by presenting the contribution taken from all the experiments done together with the recommended future work.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter gives a brief introduction to IoT networks. An overview of some methods related to reducing the data transmission based on WSNs. A quick review of some techniques applied to reduce data transmission based on WSNs and some related works.

This chapter is structured as follows: Section 2.2 introduces the IoT network. An overview of some methods related to reducing the data transmission based on WSNs is presented in Section 2.3. In Section 2.4, some time series prediction techniques applied are viewed. Related works about reducing data transmission based on WSNs are presented in Section 2.5. Section 2.6 provides a summary of this chapter.

2.2 IOT NETWORK

The advances in communication and information technologies led to the appearance of the IoT. It is a technology concept created as a global network that aims to connect a large number of physical objects and digital devices provided with several sensing and computing capabilities to interact with each other through the Internet.

The basic architecture of the IoT network can be dividing into three levels, which are the IoT data collecting like the sensors, the gateways node which responsible for the processing data, and the cloud Azar et al. (2019) as shown in Figure 2.1.

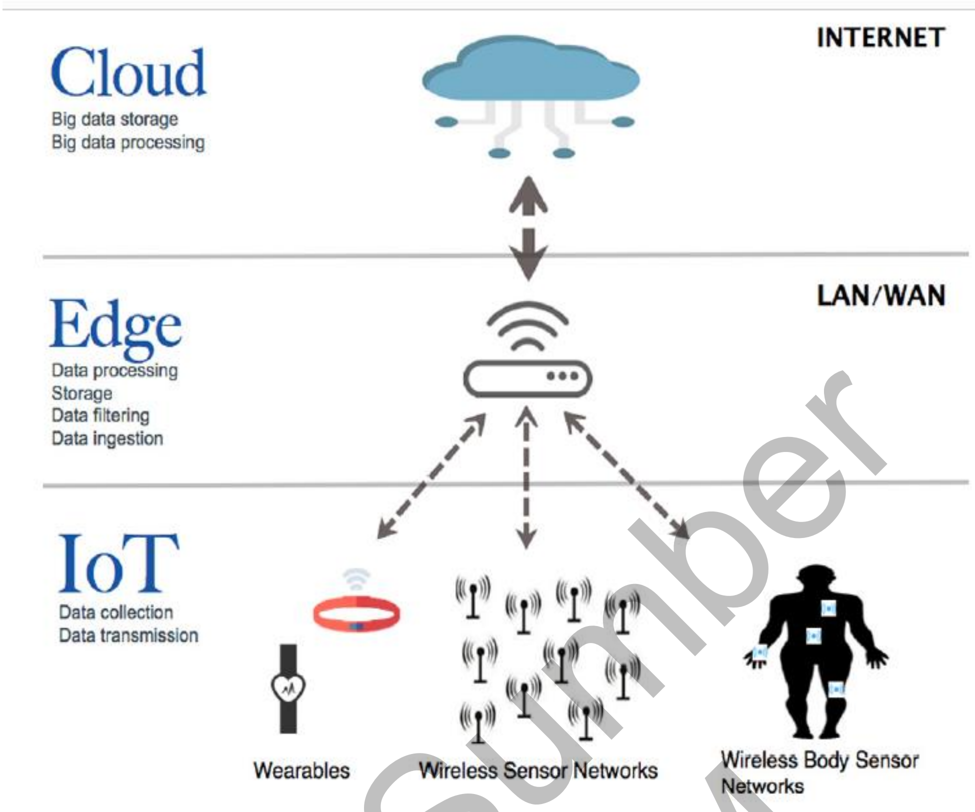


Figure 2.1 IoT Network Architecture

The IoT-based smart city includes billions of devices on the IoT network. The smart city is defined as a complex characterized by the extensive use of communication and information technology, aiming to enhance the quality of life and make cities more attractive and sustainable (Mehmood et al. 2017).

Many industrial and consumer applications in the smart city, such as transportation, health, electricity, homes, and the climate, can be managed and monitored intelligently. The wireless sensor networks are the primary source of data for these applications (Gaur et al. 2015).

WSNs consist of sensing and communication elements that can monitor and measure a specified area. Today's WSNs are essential to IoT systems, which are used to capture data for several applications. For instance, WSNs collect information such as Humidity, Temperature, light, air quality, wind speed, and vital signs. Monitoring is one of the important WSNs applications that has proven to be of interest in environmental, industrial, healthcare, government, and military applications (Jarwan et al. 2019).

Due to advances in monitoring applications and smart city services based on the IoT, the number of sensors is constantly increasing to measure and read many of the surrounding environment variables. Since the sensed data is generated in large quantities, this may cause congestion on the communication channels. Therefore, causing a delay in transferring the data to the gateways or servers in the electronic cloud, which need to process and analyze this data to make appropriate decisions (He et al. 2014).

2.3 METHODS TO REDUCE DATA TRANSMISSIONS BASED ON WSN

Some studies have focused on reducing data transmission in WSNs. Two methods, namely Data prediction and Data compression, have been used to reduce the data transmission. However, before explaining those schemes, the WSN organization should be explained to understand some terms used to describe the prediction schemes.

2.3.1 WSN Organization

WSN is located within the first level of the IoT architecture. It contains many sensor nodes and a central workstation known as a gateway (GW) that connects the sensors with the users over the Internet.

In some cases, the number of sensor nodes in the WSN can be large. Therefore, the WSN is organized internally by grouping the sensor nodes into groups called clusters, according to their location or the relationship between their measurements, to prevent the loss or collision of packets within the network and preserve network energy.

The Clustered WSN contains several normal sensor nodes and a head per cluster. The Cluster Head (CH) connects the normal sensor nodes with the GW. As shown in Figure 2.2

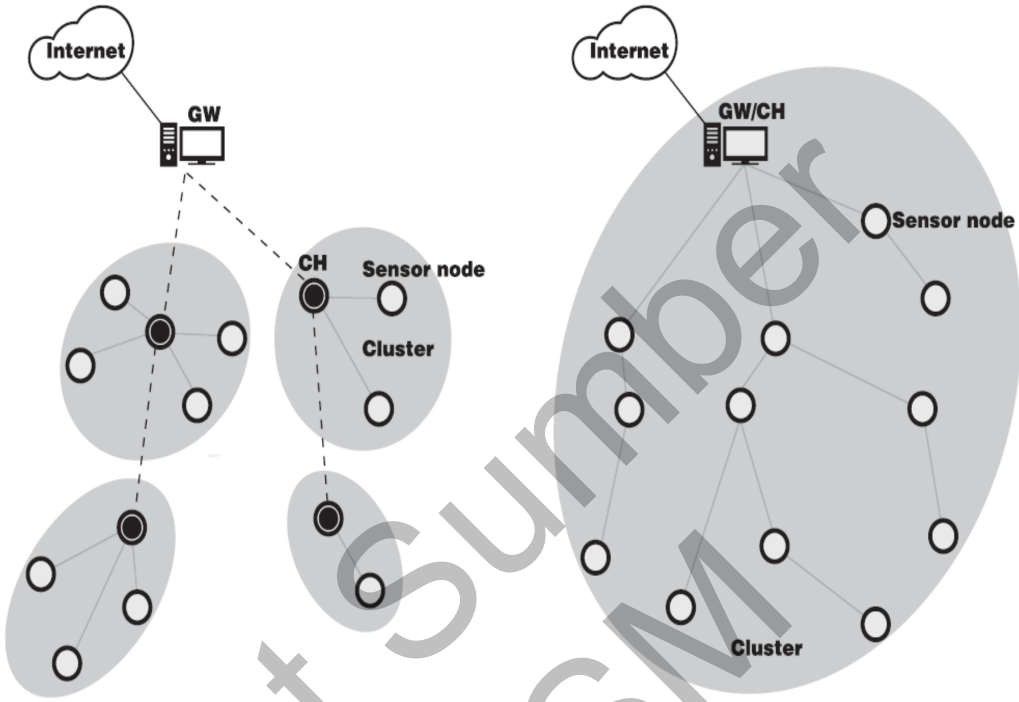


Figure 2.2 Clustered WSNs

2.3.2 Data Compression

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)

Several techniques have been used to compress the data, such as Principal Component Analysis (PCA). PCA is a classic method for analyzing data with multiple variables and linearly converting them from a collection of linked variables to noncorrelated variables. PCA can compress the data while preserving the original data set's characteristics without missing too much information. For instance, researchers in both studies Jarwan et al. (2019) and Wu et al. (2016), used this technique to compress and recover previously predicted data on CHs in Clustered WSN and Sink to lower communication costs and increase energy conservation WSNs.

2.3.3 Data Prediction

The prediction process means forecasting future values based on previous historical data or inferring lost values in a data set by experimental probability or statistics. Two important prediction schemes were used in some studies to reduce the data transmission in WSNs, thus, the congestion of the IoT communications. One is the Single Prediction Schemes; the other is the Dual Prediction Schemes (Zhang et al. 2018).

a. Single Prediction Schemes (SPSs)

Single prediction means that the predictions are made in a single point which can be in sensor nodes or CHs. For example, CHs can predict the data collected by sensor nodes. On the other hand, Sensor nodes will anticipate changes in their environments to prevent unnecessary measurements, and as a result, avoid their transmissions. The main benefit of SPSs is that each device will determine by itself whether or not to use predictions, and there is no need for them to coordinate or synchronize with other devices.

b. Dual Prediction Schemes (DPSs)

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 the sensing node and its CH. 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 real data from the environment surrounding it and sent the data to CH. Additionally, in both situations, the new data should be adding in the historical data to forecast the next data (Dias et al. 2016).

Many approaches applied the data prediction schemes to forecasting new data in different fields using time series data collected by the IoT sensors. For example, 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.

2.4 TIME SERIES PREDICTION METHODS

A time series is a sequence of measurements that are performed in sequential order over some period of time; sensors can generate time-series data (Jia et al. 2016). The prediction process means forecasting future values based on previous historical data or inferring lost values in a data set by experimental probability or statistics. (Dias et al. 2016)

Different models have been used to conduct a time series prediction. For example, traditional models like Auto-Regressive Integrated Moving Average (ARIMA), machine and deep learning models have been discussed here due to some studies such as Siami-Namini et al. (2018) and Cerqueira et al. (2019) that concluded that machine and deep learning models achieved better than the traditional.

2.4.1 Traditional Statistical and Machine learning Models

Traditional statistical models like Auto-Regressive Integrated Moving Average (ARIMA) are one of the most popular and widely used time series models that have enjoyed useful applications in forecasting. The popularity of the ARIMA model is due to its statistical properties as well as to the well-known Box–Jenkins methodology, which is a mathematical model designed to forecast data ranges based on inputs from a specified time series. In addition, the ARIMA model can implement various exponential smoothing models (Khashei et al. 2012).

Machine learning (ML) is a subfield of Artificial Intelligence based on the idea that computers can learn from data, recognize patterns, and make judgments with little or no human intervention. ML is a data analysis method that automates analytical model building.

ML algorithms have been used to analyze data in various service sectors such as healthcare, financial, and technology services. This is due to the continuous increase in the generation of data through infrastructures, smart cities, and communication networks. The ML tasks are dependent on the nature of the data it is trained on. In short, training is the act of learning the ML algorithms to achieve a specific goal by discovering potential correlations between input data and output data.

Generally, ML is divided into four primary categories which are: Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. When training supervised learning models, input and output data are presented to the algorithms. Therefore, this type can be used in classification, regression, and prediction. Unsupervised learning, on the other hand, involves training the system on unlabeled data. This is to explore the relationships between data and group them into similar structures.

Moreover, Semi-supervised learning uses labeled and unlabeled training data for the same applications as supervised learning. Finally, reinforcement learning is educated on environment data by applying reinforcement learning algorithms to it in order to determine the optimal strategies for a specific factor in different environments (Chen et al. 2019).

Different models have been used to conduct a time series prediction. For example, ARIMA and some machine learning models like Logistic Regression, Random Forest, Neural Networks (NNs), SVM, etc. Whereas some studies proved that the deep learning models outperformed these machine learning.

Siami-Namini et al. (2018) 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) presented a systematic LSTM-based method for time series analysis and forecasting. The used 33 sensor data collected for three months from a central pump in a power station to simulate the working conditions of the power stations. Root Mean Square Error was used to assess the prediction accuracy. Then they compared the LSTM model with the ARIMA model. The evaluation showed that LSTM produced higher prediction accuracy, and the results of the forecast were stable.

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) the main objective of this study is to develop a prediction model that uses deep learning. In order to provide expected future values of indicators that measure water quality for mariculture. Simple LSTM and Stack LSTM were used in this study. Six datasets were used in the experiment. 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.

2.4.2 Deep learning models

Deep Learning is a subset of Machine Learning. There are three different types of neural networks used in deep learning: (1) Artificial Neural Networks (ANN), (2) Convolution Neural Networks (CNN), and (3) Recurrent Neural Networks (RNN).

ANN, or sometimes known as Feed-Forward Neural networks, is one of the most popular machine learning algorithms. ANN contains a large number of interconnected processing elements (called neutron) at each layer that works together to solve a problem. The processing in ANN is performed in the forward direction. One of the problems that ANN can be used in is time series prediction.

Convolution Neural Networks is a type of feedforward neural network that can be used in image processing, natural language understanding, and forecast time series with great success. The convolution layer and the pooling layer are the two fundamental components of CNN. There are several convolution kernels in each convolution layer. The features of the data are retrieved after the convolution operation of the convolution layer, but the extracted feature dimensions are quite large; thus a pooling layer is added after the convolution layer to minimize the feature dimension and thus reduce the cost of training the network (Lu et al. 2020).

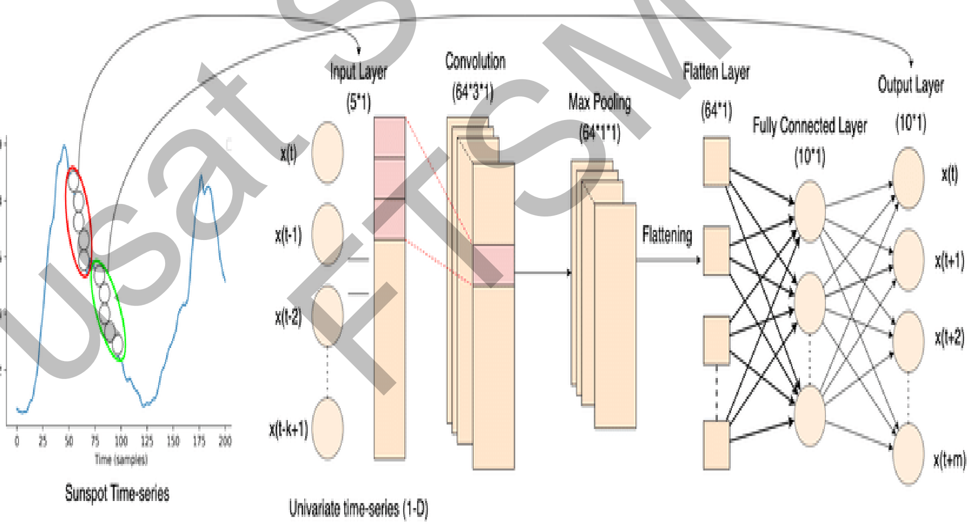


Figure 2.3 One-dimensional CNN for time series prediction

Recurrent Neural Networks are a class of feedforward neural networks, as shown in Figure 2.4. RNN is helpful in modeling sequence data like time series. RNN has an input layer, output layer, and many hidden recurrent units which have memory gates. Therefore, it can remember significant things about the input they received, which allows them to predict what is coming next. Therefore, RNN is used to find sequential correlations in time series prediction (Shastri et al. 2020).

Thai-Nghe et al. (2020) presented an architecture with a forecasting model based on LSTM. The purpose is to monitor water quality in fisheries and aquaculture that using IoT systems. The dataset which used contains some indicators such as Temperature, dissolved oxygen, salinity. Every day, these indicators are gathered and automatically transferred to a cloud server for analysis and forecasting. The Experimental results in this study revealed that the LSTM models could be a good method to improve the forecasting performance.

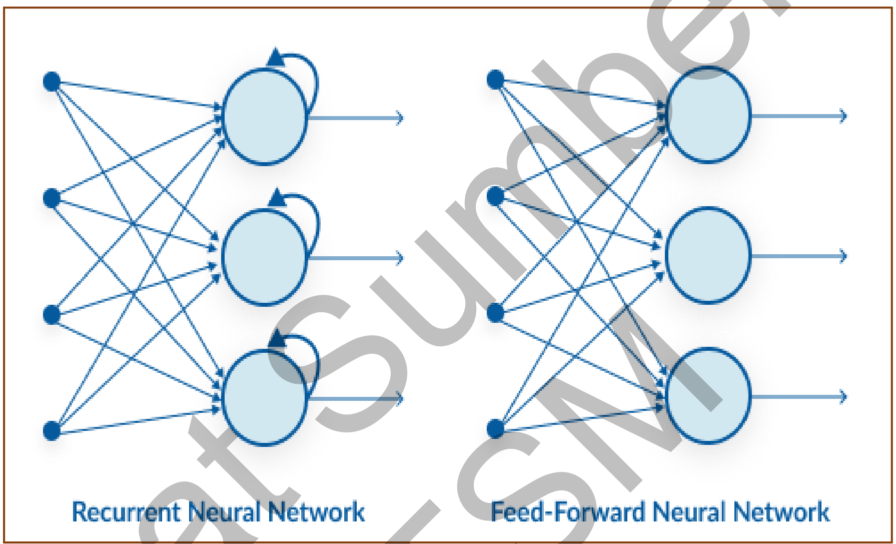


Figure 2.4 RNN and Feed-Forward Neural Networks

i. Long Short Term Memory (LSTM)

Long Short Term Memory is a type of RNN. LSTM can learn long-term dependencies by replacing the hidden layers of RNN with memory cells, as shown in Figure 2.4. Therefore, it is easy to remember past data. Additionally, Different gate units, such as input gate (i), output gate (o), forget gate (f), along with the activation function, are used to model LSTM and learn the behavior of temporal correlations.

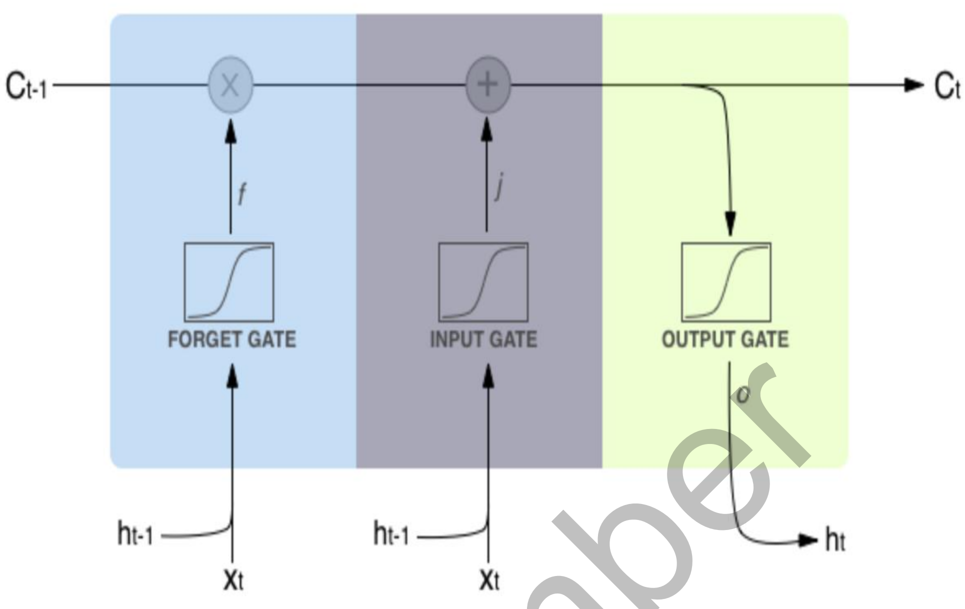


Figure 2.5 LSTM Cell

Three types of models based on LSTM, which are: Stacked LSTM, Bi-directional LSTM, and Convolutional LSTM, were used by Shastri et al. (2020) to forecast the future conditions of Covid-19 for the next month. The datasets of confirmed and death cases of Covid-19 cases in India and the USA. The researchers compared the performance of these three models. The results showed that all models achieve good accuracy, whereas the Convolution LSTM outperformed the other two models and predicted the Covid-19 cases. These three models will be used in our study to achieve our goal.

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 2.6.

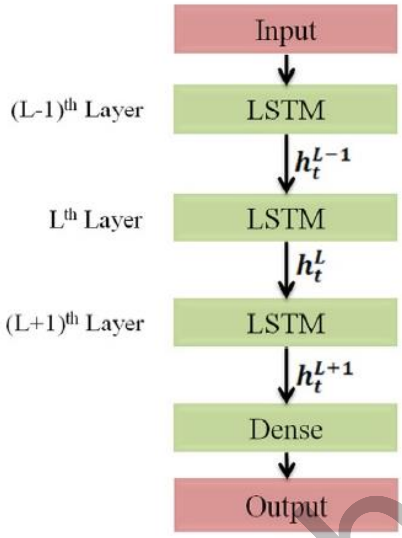


Figure 2.6 Stacked LSTM

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 simultaneously process information in both directions. 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 2.7.

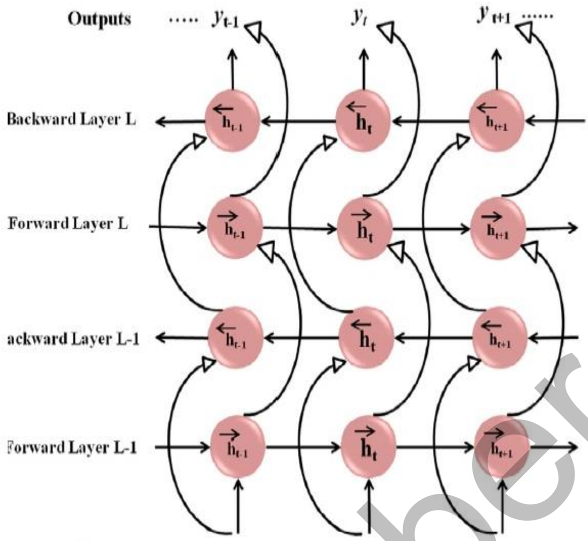


Figure 2.7 Bi-directional LSTM

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 2.8.

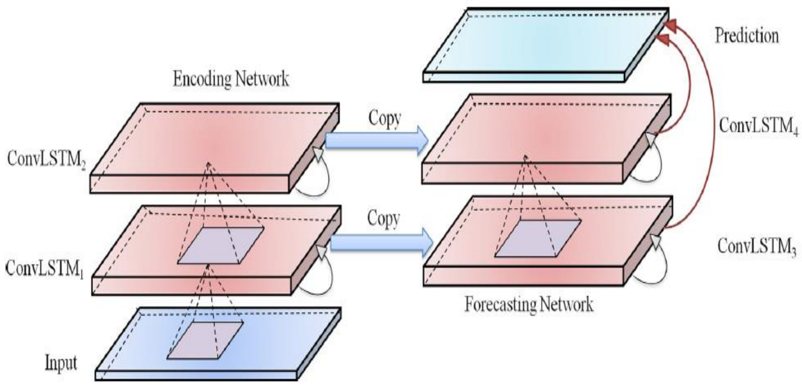


Figure 2.8 Convolutional LSTM

Several studies focused on Time series prediction using deep learning. For this reason, the authors of this paper, Chandra et al. (2021), presented an evaluation study comparing the performance of seven deep learning models on seven datasets: ACI-finance, Sunspot, Lazer, Henon, Lorenz, Mackey-Glass, Rossler. The authors evaluated the performance of the models using the mean RMSE over ten steps, then ranked the models on all the datasets. The results showed that the models ranked as BiDerc-LSTM, EncDec-LSTM, LSTM, CNN, RNN, FNN-Adam, finally FNN-SGD. They were recommended to study the performance of deep learning models on time series with Spatio-temporal correlation, such as predicting the behavior of storms and other applications such as air pollution problems.

In the same manner, (Siami-Namini et al.) compared the performance of the LSTM and ARIMA algorithms as time series forecasting models. The experiment used monthly historical time series data for the period 1985 to 2018 that was collected from the Yahoo Finance Website1. This data contains 12 stocks, with a total number of time series observations ranging from (368- 1698). Each stock data is divided as 70% for training and 30% for testing. The results revealed that LSTM outperformed the ARIMA model, where LSTM reduces the average error rates in the range from 84-87% compared with ARIMA.

Since our study focuses on time series forecasting to reduce the data transmission, three deep learning model-based LSTM (Stack LSTM, Bi-directional LSTM, and ConvLSTM) will be used to achieve our goal.

2.4.3 Walk Forward Validation Strategy

The traditional approaches used in machine learning, such as train test splits and k-fold cross-validation, are ineffective with time-series data (Brownlee 2017). The reason for this is that traditional models assume that the observations are mutually independent. In contrast, time-series datasets are correlated in time and often have temporal interdependencies. As a result, it is not correct to split them into groups randomly. The data should be divided in such a way that saves the sequence in which the values were observed, as shown in Figure 2.9. A walk forward validation methodology helps to solve this kind of problem.

A 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 2.9 (Suradhaniwar et al. 2021).

Many studies applied to walk forward validation, such as Siami-Namini et al. (2018), (Żbikowski 2015), (Gao et al. 2021), and others.

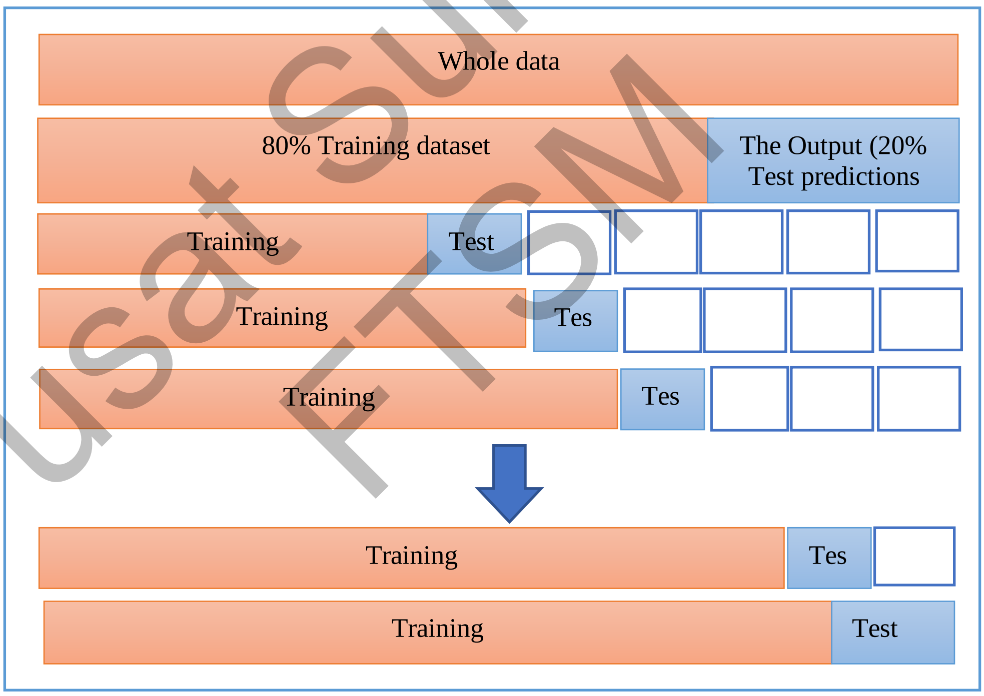


Figure 2.9 Walk Forward Validation Strategy

2.4.4 Evaluation Metrics

In general, the performance measures focus on methods for evaluating prediction values depending on the actual values. Time series prediction measures summarise the skills and capabilities of the forecasting models. Two metrics common evaluation metrics for forecasting have been selected: Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), to compare the performance between the models used in this study

a. Mean Absolute Percentage Error (MAPE)

The mean absolute percentage error 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 2.1 gives it.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \qquad \ldots (2.1)$$

MAPE indicates the percentage of error in predicting compared with the actual value. Mean Absolute Percentage Error is calculated using the summation of the absolute forecasting error ($|y_i - \hat{y}_i|$) in each period divided by the actual values for that period (Yi). Then, multiply the recap by 100 divided by the number of all actual observations.

b. 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 2.2 gives it.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \qquad \ldots(2.2)$$

RMSE is calculated using the square root of the summation of the squared absolute forecasting error ($(y_i - \hat{y})^2$) in each period.

2.5 RELATED WORK ON IOT DATA TRANSMISSION BASED ON WSNS

WSNs continuously monitor some natural phenomena like Humidity, Temperature, etc., over a large geographical area. Restricted computational power, narrow bandwidth, memory limitations, and energy supplied; present sensor nodes limits that should be considered when using wireless sensor networks for real-time data transmission. The data prediction technique is beneficial for increasing network lifetime and reducing network traffic by reducing data transmission, particularly by exploring data correlation among the sensory data (Cheng, Xie, Shi, et al. 2019).

Dias et al. (2016) have improved the WSN network structure by integrating it with artificial intelligence technologies to make them self-managed networks that benefit from cloud services and the management of data transmission during it. In addition to making the WSN networks more efficient, the researchers in this study simulated and tested the structure of this network and applied state-of-the-art algorithms for predictions to perform Simple Prediction Schemes (SPSs) and Dual Prediction Schemes (DPSs) to reduce data transmission. The researchers used a Reinforcement Learning technique called Q-Learning technology. They concluded through their results that the prediction algorithms could reduce data transmissions and improving communication within the WSNs networks by 92%, assuming the evolution of the sensor specifications and keeping data quality.

Although, this study did not provide sufficient details about the simulation process and the data used. But it opened up broad horizons for research into applying different prediction algorithms and different mechanisms such as (Sigle or Dual) within the WSN networks to reduce transmissions.

In this study, Wu et al. (2016), three techniques: data prediction, compression, and recovery, were combined in one framework. This framework aims to reduce the communication cost while guaranteeing data prediction and processing accuracy in clustered WSNs. This study used two dual prediction algorithms to predict the sensor node and cluster head (CH) data. One is the Least Mean Square (LMS), and the second is LMS with Optimal Step Size (OSSLMS) that minimizing the mean-square derivation (MSD). After that, a centralized Principal Component Analysis (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.

The authors used the simulation method to apply the techniques to the proposed framework, and the errors created during the application were theoretically assessed on real-world data. The researchers used Intel Berkeley Research Lab to collect data through 30 days for a temperature reading. The samples were taken every 31 seconds from 54 sensors deployed in the laboratory. The results showed that the OSSLMS outperforms the traditional LMS in prediction accuracy, convergence speed, and communication reduction. Additionally, the data compression algorithm achieves the saving of communication cost when sensor data is spatially correlated.

In the same manner (Jarwan et al. 2019), the authors of this study found that WSNs may generate spatial and temporal correlation data that most can derive from each other. Therefore, this study tried to prevent unnecessary data transfer to save energy and bandwidth in WSNs. To achieve their goal, the researchers applied Dual Prediction (DP) and Data Compression (DC) 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 DP 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, For example, the number of layers in NNs and the number of neurons. Therefore, the chance of applying other models with other parameters may give better results than the results obtained in this study.

MNMF Multi-Node Multi-Feature (MNMF) is a method presented by Cheng, Xie, Wu, et al. (2019) to eliminate unnecessary data transfer within the WSN. It is based on bidirectional LSTM. 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. 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.

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.

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.

Additionally, the deep learning models showed better performance than the traditional and machine learning approaches depending on the previous studies reviewed.

Therefore, our study focused on applying deep learning forecasting models to reduce the data transmission between the IoT sensor and the gateway using a prior dataset for one year of sensed data from the Malaysia Atmospheric dataset to make the forecasting.

Table 2.1 illustrates the previous work conducted on time series prediction and data transmission reduction and summarizes studies related to our work with their strengths and limitations. It is clear that there were some gaps in these studies that have not been fulfilled.

Table 2.1 Literature of the Related Work

No.	Year	Author	Description	Strengths	Limitations
1	2016	(Wu et al.)	Three techniques (prediction, compression, and recovering) were applied in the proposed framework to reduce data transmission and communication costs.	The experimental results showed that the OSSLMS algorithm offered better performance than the traditional LMS.	Applied the algorithms in simulations, and the analysis was implemented theoretically. Furthermore, the models used in this study were traditional machine learning models, not deep learning models.
2	2016	(Dias et al.)	This study simulated a self-managed WSN network that used the AI technique. Additionally, state-of-the-art prediction algorithms are used to reduce data transmission.	The result showed that prediction algorithms could reduce the data transmissions within the WSNs networks by 92%.	This study did not provide sufficient details about the simulation process, the data used, or the prediction algorithms.
3	2019	(Jarwan et al.)	The NNs, LSTM, and OSSLMS models were used to apply the dual prediction to save the lifetime of the sensor and improving communication.	The result revealed that the LSTM deep learning model had outperformed the other two models.	The study applied the algorithms in simulations. The chosen model parameters significantly affect the model's performance. Therefore, it is a chance to apply the models with other parameters.
4	2019	(Cheng, Xie, Wu, et al.)	MNMF (Multi-Node Multi-Feature) model was proposed to reduce unnecessary data transfer in a special WSN structure.	The results proved that the proposed model outperformed the other three models using the MAPE measurement,	They did not calculate the amount of data transmission reduction using their proposed model.

… to be continued

… continuation

			Also compared MNMF with another three neural network prediction models using 54 sensor nodes data collected by Intel Berkeley Research Laboratory.	with an error rate of 0.318 compared to 0.698 for Elman, 0.328 for GRNN, and 1.17 for NARX.	
5	2018	(Siami-Namini et al.)	This study compared the performance of the LSTM and ARIMA algorithms as time series forecasting models. The experiment used monthly historical time series data for the period 1985 to 2018 that was collected from the Yahoo Finance Website1.	The results revealed that LSTM reduced the average error rate by 87% comparing with ARIMA.	Very small numbers of time series observations that used in the implementation.
6	2020	(Thai-Nghe et al.)	This study aims to develop a time series prediction model using deep learning, so LSTM was used with RMSE to measure the quality of water. Six data sets were used in the experiment.	The results showed that the Stack LSTM model outperforms the Simple LSTM, and in general, both the LSTM models achieved better results than the SVM baseline model.	The authors used a simple LSTM, and they recommended using sophisticated deep learning models in further research.

… to be continued

… continuation

7	2019	(Udeh et al.)	This paper used LSTM and MLP to model a time series prediction. Hourly weather data from New York State Mesonet stations were used in this study.	The findings indicate LSTM AND MLP predictive ability in the problem, exceeding the baseline model.	This study implemented a multivariate time-series method.
8	2021	(Chandra et al.)	This study aims to compare the performance of seven deep learning models using seven benchmark datasets.	The results showed that the models ranked as BiDerc-LSTM, EncDec-LSTM, LSTM, CNN, RNN, FNN-Adam, finally FNN-SGD based on RMSE measurement.	Some of the benchmark datasets that were used are not a real-time series data it is a simulated data. Moreover, the number of time series points is small, like the Sunspot dataset has 2000 data points. The ACI-finance contains 800 data points. The Lazer consists of only 500 points.
9	2019	(Siami-Namini et al.)	This study aims to compare the performance of (BiLSTMs) with regular unidirectional LSTM. Two datasets were used in this study. One dataset has (368) time-series observations. However, another one has (82,205) time-series observations.	The results revealed that additional data training and hence BiLSTM-based modeling gives 37.78% better predictions than normal LSTM-based models on average. However, BiLSTM models appeared more slowly than LSTM-based models in terms attain equilibrium.	The time-series data that are used are not seasonal time series data which has correlated circle data.

2.6 SUMMARY

This chapter provides the literature review and the related works. It introduces an overview of. Furthermore, the chapter also discusses the clustered WSNs and the concepts around reducing the data transmissions in WSNs. An explanation for two techniques used to reduce the data transmissions. These techniques are data compression and data prediction.

Additionally, Some studies that used machine and deep learning techniques to apply time-series data prediction were reviewed, and these studies proved that deep learning techniques outperformed machine learning techniques. Also, previous studies that used data prediction techniques using time series were reviewed in order to reduce data transmissions within the WSNs.

Most of the studies that tried to reduce data transmission used machine learning techniques. According to what the researchers stated in Jarwan et al. (2019), no one before them used LSTM technology in this regard. According to their results, LSTM outperformed NNs and OSSLMS techniques. On the other hand, most of the studies tried to reduce the data transmissions within the WSNs.

Therefore, the processes of reducing data transmission are still confined within the WSNs. Therefore, this work aims to reduce the data transmission between the IoT sensor and the gateway by applying deep learning forecasting models with a prior dataset.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This chapter outlines the research methodology of the project to achieve the goals of the study. The steps of the proposed work have been described in the experimental design used in this study.

This chapter is organized as follows: Section 3.2 illustrates the experimental design that contains all stages of the methodology used. Section 3.3 presents the datasets preparing in this study. Then, Section 3.4 offers the deep learning models for time series forecasting-that are used in this project. After that, Section 3.5 gives a brief about the walk-forward validation method and the evaluation metrics to evaluate the performance of models used in this study. Section 3.6 explains the data transmission reduction. Finally, a brief summary of the chapter is provided at the end of the chapter in Section 3.7.

3.2 EXPERIMENTAL DESIGN

The research design that has been followed in this project 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.
- Stage 2: Implementation.
- Stage 3: Evaluation.
- Stage 4: Data Transmission reduction.

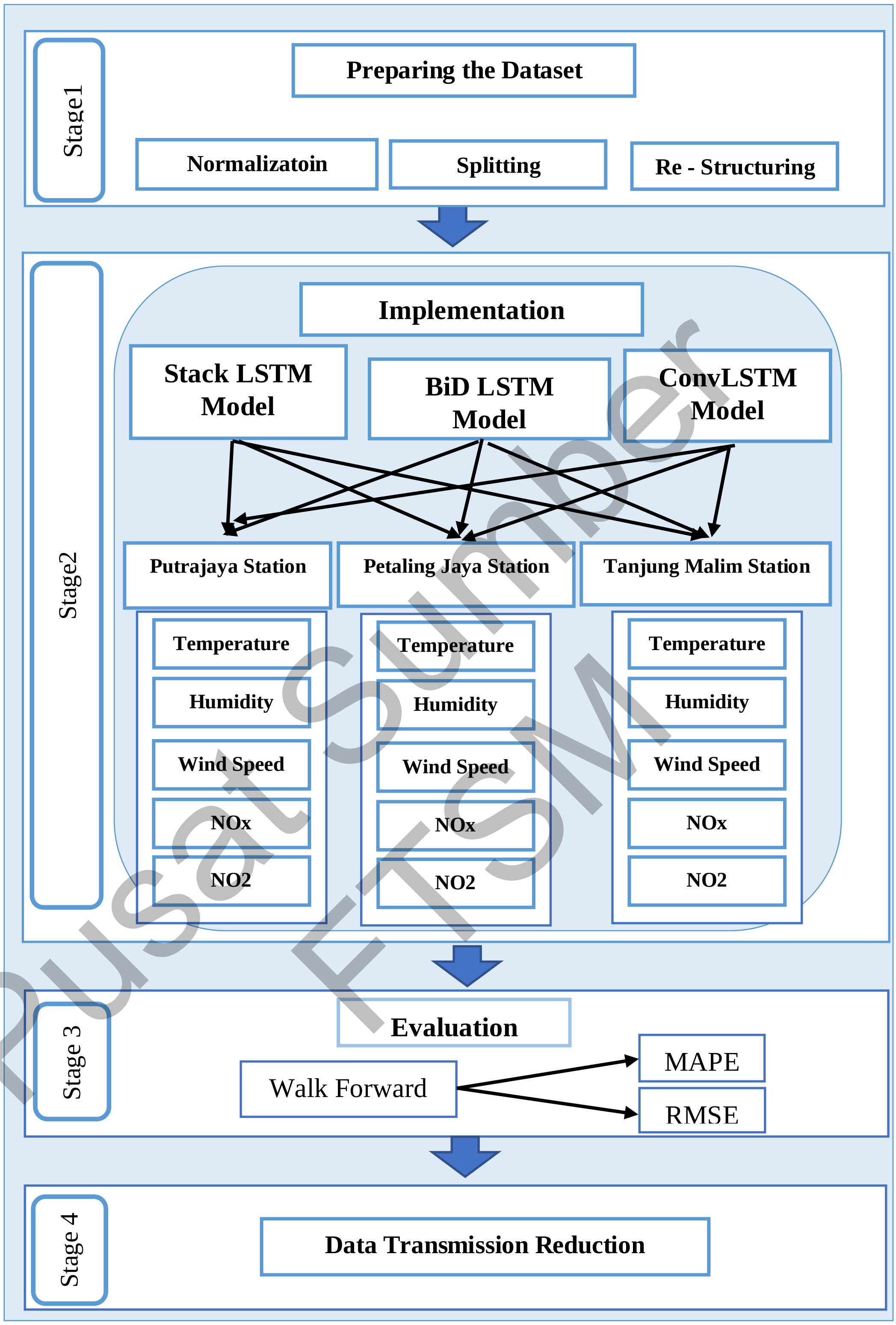


Figure 3.1 Experimental design for DL models

Figure 3.1 above gives a full view of the research design used in this study. The details of the research phases will be explained later in the following sections.

3.3 PREPARING THE DATA SET

The aim of Stage 1 is to prepare the datasets to input in models. The following paragraphs provide brief information about the dataset. Moreover, the preprocessing for the dataset has also been explained.

3.3.1 The Dataset

The dataset that used in this study 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). Figure 3.2 below shows a sample of the Putrajaya dataset, and Table 3.1 shows more details.

	A	B	C	D	E	F	G	H	I
1	Year	Month	Day	Hour	Temp	NOx	NO2	Humidity	WindSpeed
2	16	1	1	1	24.1	0.057	0.028	86	3.5
3	16	1	1	2	23.9	0.029	0.024	84	3.4
4	16	1	1	3	23.8	0.023	0.019	87	2.6
5	16	1	1	4	23.7	0.016	0.013	89	3.7
6	16	1	1	5	23.1	0.015	0.011	89	1.6500001
7	16	1	1	6	23	0.013	0.01	89	3.5
8	16	1	1	7	23	0.014	0.01	90	1.7
9	16	1	1	8	23.1	0.015	0.011	92	2.2
10	16	1	1	9	25	0.023	0.016	91	1.7
11	16	1	1	10	28.4	0.02	0.013	88	2.9
12	16	1	1	11	31.1	0.024	0.017	83	1.45
13	16	1	1	12	32.4	0.016	0.01	73	1.7
14	16	1	1	13	32.1	0.011	0.007	68	1.5
15	16	1	1	14	30.6	0.011	0.007	68	1.5
16	16	1	1	15	28.6	0.013	0.009	72	1.4000001
17	16	1	1	16	28.5	0.012	0.008	72	1.8
18	16	1	1	17	29	0.012	0.008	69	1.1
19	16	1	1	18	27.9	0.012	0.009	69	3.3
20	16	1	1	19	26.4	0.012	0.009	75	1.6500001
21	16	1	1	20	25.5	0.012	0.008	78	2.5
22	16	1	1	21	25.4	0.02	0.012	81	2.9

Putrajaya(2016)

Figure 3.2 Sample of Putrajaya Station DataSet

Table 3.1 Dataset Details

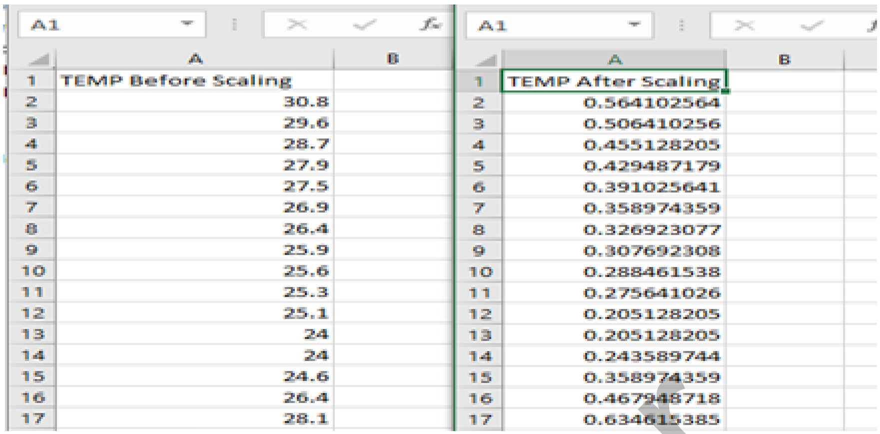
	Data range	Putrajaya	Petaling Jaya	Tanjung Malim
Temperature	Min	20.8	22.6	20
	Max	37.9	38.4	39.7
Humidity	Min	55	34	39
	Max	96	100	109
Wind Speed	Min	0.9	1	0.9
	Max	11.2	18.8	16.4
NOx	Min	0	0.001	0.001
	Max	0.122	0.201	0.069
NO2	Min	0	0	0
	Max	0.062	0.094	0.031

3.3.2 Preprocessing Dataset

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

Firstly, 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). Figure 3.3 shows a Sample of Temperature Attribute Before and After Normalization in Putrajaya Data



	A	B		A	B
1	TEMP Before Scaling		1	TEMP After Scaling	
2	30.8		2	0.564102564	
3	29.6		3	0.506410256	
4	28.7		4	0.455128205	
5	27.9		5	0.429487179	
6	27.5		6	0.391025641	
7	26.9		7	0.358974359	
8	26.4		8	0.326923077	
9	25.9		9	0.307692308	
10	25.6		10	0.288461538	
11	25.3		11	0.275641026	
12	25.1		12	0.205128205	
13	24		13	0.205128205	
14	24		14	0.243589744	
15	24.6		15	0.358974359	
16	26.4		16	0.467948718	
17	28.1		17	0.634615385	

Figure 3.3 Sample of Temperature Attribute Before and After Normalization in Putrajaya Data

b. Splitting Data

In this project, 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, the data must be reframed using the lag method or sliding window to implement deep learning and machine learning algorithms. 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 3.4 (Kotriwala et al. 2018).

Figure 3.5 shows a sample of converting data from series or sliding window (like 0.2514 0.2514 0.3684 …0.8596 0.9239 0.9005 0.8538 0.8538 0.8128 0.7309) to lags and targets as showing in Figure 3.4. Moreover, it is clear using arrows in Figure 3.5 that the output y in the first sequence becomes a prat in input data x in the next sequence, as shown and explained in Figure 3.4.

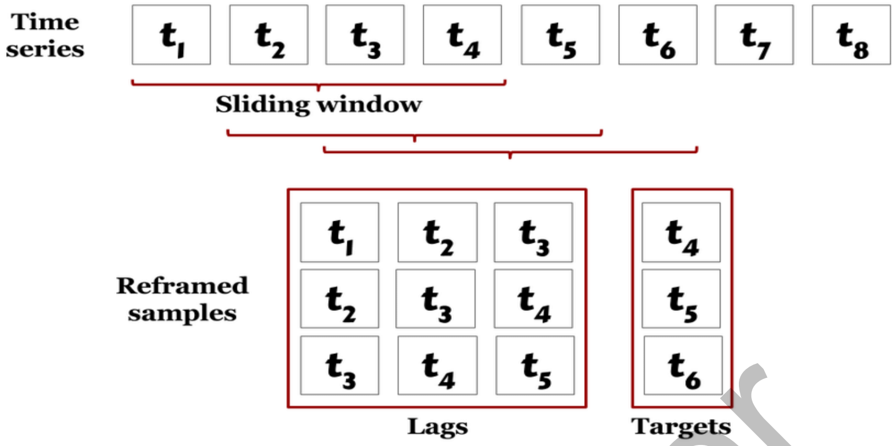


Figure 3.4 Restructured the Datasets

...	...	...	...	...	...	...	...
7022	0.251462	0.251462	0.368421	...	0.859649	0.923977	0.900585
7023	0.251462	0.368421	0.538012	...	0.923977	0.900585	0.853801
7024	0.368421	0.538012	0.660819	...	0.900585	0.853801	0.853801
7025	0.538012	0.660819	0.783626	...	0.853801	0.853801	0.812865
7026	0.660819	0.783626	0.859649	...	0.853801	0.812865	0.730994

Figure 3.5 Restructured the Dataset Sample

3.4 IMPLEMENTATION

Stage 2 aims to show the model implementation in this study. Three types of models based on LSTM: Stacked LSTM, Bi-directional LSTM, and Convolutional LSTM, were used (see chapter II) due to the accuracy of these models that was achieved in (Shastri et al. 2020) study. Three stations datasets have been used in the experiment. Each model applied on all the five attributes one by one for each station dataset, as shown in Figure 3.1.

Furthermore, three different sequences representing 8,24, and 73 hours have been chosen as input data for each attribute. Input data means forecast one new hour after sequence consists historical 8 hours, 24 hours, or 72 hours as shown in Figure 3.6.

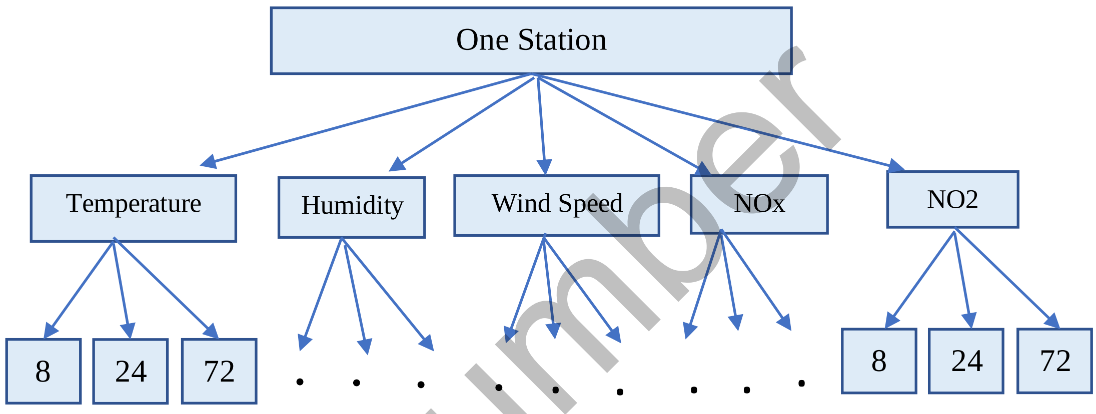


Figure 3.6 Implementation Steps for Each Station Based Different Sequences

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 3.2.

Table 3.2 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 be determined by comparing the average of (MAPE) values and the accuracy calculated duo to the MAPE cause it is a scale-independent measurement.

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.

3.5 EVALUATION

In stage 3, the evaluation process is carried out. Walk forward validation strategy with two measurements RMSE and MAPE, were used to evaluate the performance of models in this study.

3.5.1 Walk Forward Validation

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 strategy was explained in section 2.4.3.

3.5.2 Evaluation Metrics

RMSE and MAPE are standard evaluation metrics for forecasting. RMSE is a scale-dependent measurement that can use to compare different models on one dataset. However, MAPE is scale-independent that can use to compare one model on different datasets. More detail is explained in section 2.4.4.

3.6 DATA TRANSMISSION REDUCTION

In this stage, 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 study, 0.5 was chosen as a threshold for Temperature (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) as shown in Tabel 3.1.

Based on those three points, the number of accurately predicted values will calculate by using the same data of the three stations for the same year 2016 by 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 formula 3.1.

$$TRP = (\text{Accurate Data} / \text{Total Number of Raws}) * 100 \qquad \ldots(3.1)$$

3.7 SUMMARY

This chapter has illustrated the research methodology applied for this study. The research methodology included four main stages: (1) Preparing datasets, (2) Implementation, (3) Evaluation and comparison results, and (4) Data transmission reduction. In the first stage, the datasets used in this study have been overviewed. Additionally, some preprocesses have been done on the datasets to prepare them for final processing. In the second stage of the research methodology, the models used in this study have been detailed. Besides, the values of the parameters used in these algorithms or models have been stated. Tow evaluation metrics have been mentioned in the third stage to evaluate our models to choose the best one. In the final stage, explained how to calculate the data transmission reduction.

CHAPTER IV

ANALYSIS AND DISCUSSION

4.1 INTRODUCTION

This chapter presents the results obtained from implementing the three deep learning models: Stack LSTM, BiD LSTM, and ConvLSTM using three stations from Malaysia Atmospheric Dataset for the year 2016. The results obtained from the models are compared to choose the best model. After that, the best model was used to calculate the data transmission reduction results.

Section 4.2 presents the experimental results that have been implemented on the three different atmospheric station datasets in Malaysia. Section 4.3 provides the experiment to reduce data transmissions. Finally, a summary of the chapter is provided.

4.2 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. Stack LSTM, BiD LSTM, and ConvLSTM models were used. The results for each model have been recorded. The experiment of this study has been applied to three different station datasets, Putrajaya, Petaling Jaya, and Tanjung Malim station. Each station has five dimensions represented the Temperature, Humidity, Wind Speed, NOx, and NO2 of the Malaysia Atmospheric.

In this study, the univariant time series forecasting performance has been evaluated using accuracy measurements like RMSE and MAPE. The results obtained from this experiment are shown in the next section.

4.2.1 Tuning Parameters

Three LSTM based models have been used to achieve our goals. These models were used previously in (Shastri et al. 2020). The same configurations of that models were used as explained in chapter III, except the number of epoch and the input data.

First, Shastri et al.'s study used 500 epochs regarding their datasets, whereas we used a small epoch to be suitable to our dataset. The experiment has been applied BiD LSTM to choose the best epoch from 10 to 100. The increasing step is 10 and 30 times of implementation for each step. The results that were stated in Table 4.1 showed that the lowest mean of RMSE is with the epochs 40, so it is used in this study.

Table 4.1 Epoch Parameter

Epochs	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Mean of RMSE	1.137	0.993	0.987	**0.956**	1.066	7.467	1.217	1.67	1.28	3.968

Second, the data in our study is time-series data that represented hourly sensor readings. Therefore, the input data represented the historical data used to predict the next new hour, as shown in Figure 3.4. Three different input data sequences, 8, 24, and 72 hours were used as input data, as shown in Figure 3.6. These sequences were chosen depending on the correlation of the attributes data to achieve a good prediction accuracy, thus choosing the best input data sequence. The average of the MAPE for ten iterations was used to evaluate the performance of the models. The results were shown in Figures 4.1, 4.2, and 4.3 for the Putrajaya, Figures 4.4, 4.5, and Figure 4.6 for the Petaling Jaya, and Figures 4.7, 4.8, and 4.9 for the Tanjung Malim.

4.2.2 Putrajaya Results

Figures 4.1, 4.2, and 4.3 showed the prediction performance between the three models in terms of the MAPE average for ten times implementations for the Putrajaya data station.