EDGE AI IN LORA BASED MESH NETWORK

NG XIN HAO

UNIVERSITI TUNKU ABDUL RAHMAN

EDGE AI IN LORA BASED MESH NETWORK

NG XIN HAO

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Electrical and Electronic Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2022

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Ng Xin Hao

ID No. : 1806864

Date : 12th September 2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"EDGE AI IN LORA BASED MESH NETWORK"** was prepared by **NG XIN HAO** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Honours) Electrical and Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ______________________________

Supervisor : Ir Ts Dr Tham Mau Luen

Date : 12th September 2022

Signature : ______________________________

Co-Supervisor : ______________________________

Date : ______________________________

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Ir Ts Dr Tham Mau Luen for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement.

Finally, I would also like to show my appreciation to all my supportive friends and coursemates. Their support and presence themselves have continuously given me inspiration and motivation that led to the success of this project.

ABSTRACT

Natural disasters such as floods frequently occur in Malaysia. Internet of Things (IoT)-based flood early warning systems can forecast the cataclysmic flood event and subsequently inform the public to take evacuation action earlier. However, the issue of disseminating critical information remains an open issue if the communication network is broken. This project aims to develop a lightweight Artificial Intelligence (AI) disaster forecasting and a vicinity communication infrastructure, a resilient NerveNet mesh network with Wi-Fi and LoRa. It will disseminate the information about forecasted flood events ahead of time reliably to the designated recipients even if the base station is destroyed due to a flood. Using the NerveNet Hearsay daemon, texts and images can be synchronised wirelessly in multiple NerveNet nodes' databases. Experimental results validate the AI model, network, and database synchronisation performance. The project findings can serve as the guideline for designing an AI flood early warning system in real life.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AC	Alternating Current
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Access Point
DC	Direct Current
DNN	Deep Neural Network
GPS	Global Positioning System
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
IoT	Internet of Things
IR	Intermediate Representation
LAN	Local Area Network
LoRa	Long-Range
LoRaWAN	Long-Range Wide Area Network
LPWAN	Low-Power Wide Area Network
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
NMEA	National Marine Electronics Association
OTAA	Over-The-Air Activation
PTMGR	Path Tree Management Generation
PV	Photovoltaic
QoS	Quality of Service
R^2	R Squared
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SVM	Support Vector Machines
WLAN	Wireless Local Area Network

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Flood forecasting models have been researched in the hydrological engineering area for many years. Recently, there has been increased research interest in river flood prediction and modelling, defined as data-driven approaches. The ANN model is the most famous usual data-driven approach. Most conventional statistical methods require a lot of data for their models, and they can generate no assumptions for both linear and non-linear systems. Hence, the data-driven approach, ANN, is an alternative to hydrological flood forecasting instead of the existing methods (Kişi, 2011).

Artificial Intelligence made essential development in modelling hydrological forecasting and dynamic hydrological issues. With the advancement of information technology, the application of ANN models in many aspects of science and engineering is increasingly becoming common due to its simplicity of structure. Diverse neural network modelling approaches have been applied, like implementing the model approaches individually or combining process-based approaches to minimise mistakes and increase the models' forecasting accuracy. The study by (Yaseen et al., 2015) applied AI models to forecast river flow for 15 years starting from 2000 since there are many advantages of ANN models in hydrological modelling and forecasting related to hydrology fields (Yaseen et al., 2015).

However, there are also some limitations of the ANNs model. One of them is lacking understanding of watershed processes. Furthermore, the limitation of memory in calculating sequential data exposes the disadvantages of the ANNs model. The breakthrough in computational science has recently increased the interest in DNN approaches that rely on ANN. In addition, the most recent DNN applications, such as the LSTM and GRU neural networks, have been efficiently implemented in diverse areas and fields, such as time sequence problems. Those models can apply to machine translation, speech recognition, tourism field, language modelling, rainfall-runoff simulation, stock prediction and river flow forecasting.

On 11th March 2011, around 29000 cellular towers were damaged in the East Japan Great Earthquake. These damages have restricted the broadcast of evacuation notices and the collection of historical information for disaster forecasting. Hence, it can be known that the resilience of a network remains an open issue in the deployment of the fault-tolerant network during an emergency disaster. Fortunately, a disaster-resilient mesh-topological network called NerveNet was developed by Japan NICT. Each NerveNet node is independent and tolerant to system failure and link disconnection due to its mesh structure.

In this project, a flood forecasting model is proposed. In the study area, rainfall and river water levels collected at hydrological stations are input as the dataset for the training and testing process of the AI models. Then, the forecasted flood water level will be processed to generate the flood warning message and graph. It will be sent through the NerveNet Wi-Fi and LoRa mesh network. After that, the flood warning message and graph are synchronised to every node. Finally, the performance of the AI model, NerveNet Wi-Fi and LoRa mesh network and database synchronisation are evaluated.

1.2 Importance of the Study

This project meets specific needs with appropriate consideration for public health and safety. An efficient and effective flood forecasting model can reduce the effect of flood events, which increases the public's safety. Besides that, effective flood forecasting systems will be able to record the rainfall and river water levels to produce essential forecasted river water levels in the future. In addition, a reliable NerveNet mesh network can disseminate critical information reliably to the designated recipients even if the base station is destroyed due to a flood so that the public can prepare to retreat from the coming flood-affected area. According to a study by (Faruq et al., 2020), flood forecasting models are frequently useful in flood warning and management systems (Faruq et al., 2020). The availability of flood forecasting systems was associated with a substantial decrease in the likelihood of public-facing floods. Apart from that, several situational factors also appear to be associated with the likelihood of flood forecasting systems being available and useful. This project shows that implementing a flood forecasting system will help safeguard the public to a higher degree.

1.3 Problem Statement

Over the past decades, several studies have been conducted in various areas, such as monitoring, detection, early warning systems, and so on, to minimise the impact of flooding by alerting the public about a flood event occurrence ahead of time. Significantly, accurate flood forecasting is essential to predict the hourly water level to reduce the risk of floods. It is also crucial for water resources systems planning and management. However, it is difficult to accurately forecast the river discharge since the river's flood analysis is a complicated non-linear operation influenced by many temporal and spatial variables. Besides that, the flow of the river is also a non-linear operation. It is affected by many variables such as rainfall, climate, river basin surface mantle, and riverbed terrain. Therefore, several predictive actions need a plethora of data to predict floods accurately based on the measured surrounding conditions. The hardware prototype will consume many time and power to collect the big data.

The current network in Malaysia has a shortcoming which is the absence of regional resilient network topology and architecture for emergency use. Besides that, a database synchronisation feature must also be set in the resilient network. The database synchronisation feature allows devices to access data from each other natively in the resilient regional network.

1.4 Aim and Objectives

There exist several flood forecasting solutions in the current market. However, the forecasting efficiency is limited due to the conventional approaches, which are complex and time-consuming. Besides that, most of the networks are tree topology networks. Hence, the objectives of this project are:

1. To develop a lightweight AI disaster detection.
2. To set up a NerveNet mesh network testbed with LoRa and Wi-Fi.
3. To synchronise AI results among NerveNet nodes.

1.5 Scope and Limitation of the Study

Natural disasters are the significant factors in social losses, economic losses, and human life loss. Various natural disasters pose a potential danger to densely populated areas worldwide. Therefore, a natural disaster risk management system is essential to reduce natural disaster risk. However, it is quite a vast scope and area to include all types of natural disasters in one risk management system. Besides that, a long time, much power and many resources will be consumed to collect all the variables that influence the occurrence of all types of natural disasters and build forecasting models for every kind of natural disaster. Therefore, in this project, the development of the natural disaster forecasting system will only focus on the flood event that always happens during Malaysia's monsoon season.

Besides that, it is known that many variables influence the occurrence of flood events. It is a high cost to distribute various data collection stations for all types of variables in the whole Malaysia area. Hence, there are only two variables used in this project which are the rainfall and river water level. In addition, it is known that flood events are significant natural disasters in various parts of the world. Several pieces of flood prediction research are overseas to support their natural disaster risk management system. In this project, an efficient flood forecasting system will be introduced to focus on these frequent flood-affected areas during the monsoon season, which is Malaysia, to improve Malaysia's own natural disaster risk management system in a better way.

1.6 Contribution of the Study

An AI flood forecasting model is developed to forecast the flood event occurrence ahead of time. Besides that, the NerveNet mesh network testbed using Wi-Fi and LoRa is deployed as a disaster forecasting network in this study. The data is synchronised between every node within the NerveNet mesh network for data resiliency. Since NICT ASEAN IVO funds the development of the NerveNet mesh network, this project's findings and outcomes will become the guideline and reference for the deployment of NerveNet in Malaysia and other countries.

1.7 Outline of the Report

This report has five chapters. The first chapter introduces the project's title, aim and objectives. Then, the second chapter explores the concept and areas related to the project, such as artificial intelligence, mesh network, LoRa, database synchronisation and so on. After that, the third chapter shows the project's work plan and methodology. Next, the fourth chapter shows the result and the discussion on the performance of the AI model, mesh network and database synchronisation. Lastly, the final chapter concludes the objectives and outcomes of the project and gives future recommendations to improve the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter introduces the project's related areas: IoT, distributed computing, Artificial Intelligence, network topology, LoRa and NerveNet. The combination of these areas will give an overview of edge AI in the LoRa-based mesh network for the flood early warning system.

2.2 Internet of Things (IoT)

IoT is a network of a cluster of connected embedded devices with identifiers. Its communication that uses the standard communication protocol can be implemented without any intervention from a human. In the era of modernisation, IoT is one of the leading technology trends in disaster monitoring and detection system to predict the disaster incidents such as earthquakes, tsunamis, floods and so on, so that the loss of life and property damage can be minimised. IoT provides many functions such as intelligent information processing, reliable information transmission and overall information perception. These characteristics of IoT can provide an effective guarantee for disaster forecasting, detection and precaution ahead of time through the IoT-based early warning system so that the impact of a disaster can be reduced (Samikwa, n.d.).

2.3 Distributed Computing

The distributed processing or distributed computing uses multiple processors, computers, or software components but runs as a single system. The components can be connected within LAN or WAN, which makes the entire network structure work as a single computer to offer benefits such as scalability and redundancy. In other words, the system can be expanded easily, while several components can provide the same services to ensure service continuity when one of the machines is unavailable (IBM, 2021).

2.3.1 Edge Computing

Edge computing is a part of distributed computing technology. It acts as an additional layer between the end devices and the server. Edge computing shifts the computational power nearer to the end devices (Gezer, Um and Ruskowski, 2018). The collected data processing is implemented close to the edge instead of at the central server. The scalability problem of IoT architectures is attributed to the exponential growth of devices that connect to the internet to receive data from the cloud and send data to the cloud.

In general, the end devices in the IoT networks have sensors that produce a large amount of data at a very high speed during their operations. In the traditional IoT mechanism, these data are transmitted to the remote central cloud platform through the internet to be processed. However, there is a problem where the big data transmitting process consumes enormous energy, time, cost and bandwidth. Therefore, edge computing is introduced to process and analyse the valuable information from the raw sensor data at the edge level in real-time. In fact, edge computing utilises the computational resources of the network's edge and shifts the data processing nearer to where data are generated (Rausch, Nastic and Dustdar, 2018). Thus, it can improve the QoS of applications and reduce the tasks' latency (Chen et al., 2018). Figure 2.1 illustrates edge computing’s paradigm. The computing processes of the task can be offloaded to the edge using edge computing. Hence, the response time is reduced, and the system's efficiency increases.

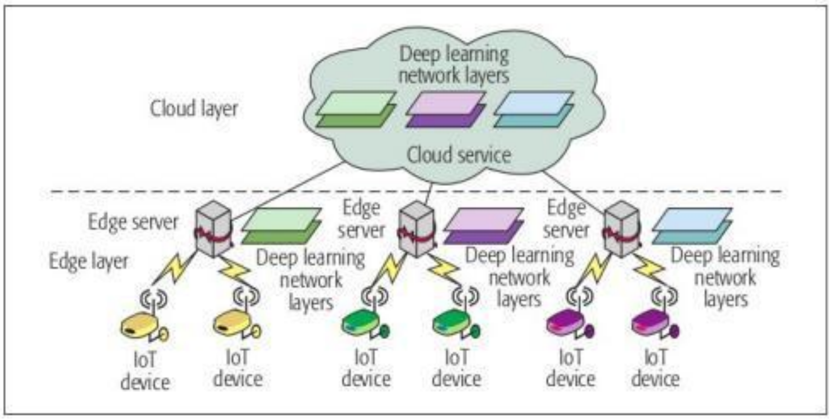


Figure 2.1: The Paradigm of Edge Computing (Samikwa, n.d.).

2.4 Machine Learning

Machine learning is a branch of AI applied in several areas, such as predictive data analytics. In general, before predicting a specific type of output, the machine learning model must undergo a model training process. Before the model building, a dataset in which each data contains some features and a label are required to be prepared according to the use case. After the dataset is divided into a training dataset and a testing dataset, the training set acts as the input to the machine learning model during its training process to identify the weightage of those features. There are several types of machine learning algorithms, such as supervised, unsupervised, and reinforcement learning.

Supervised learning is a data sample training from a dataset with the features and their labels. Supervised learning plays a role in training the machine learning model with the labelled dataset. After that, the machine learning model's predicted outputs are compared with the ground truth to calculate the machine learning model's accuracy.

2.4.1 Random Forest

Random Forest is defined as an ensemble classifier where multiple learning algorithms are utilised to obtain a better accuracy for its prediction (Mushtaq, Augustin and Mellouk, 2012). The Random Forest algorithm uses a few Decision Tree models. Each tree makes its prediction, and the Random Forest model will opt for the most votes class. Figure 2.2 illustrates the Random Forest algorithm.

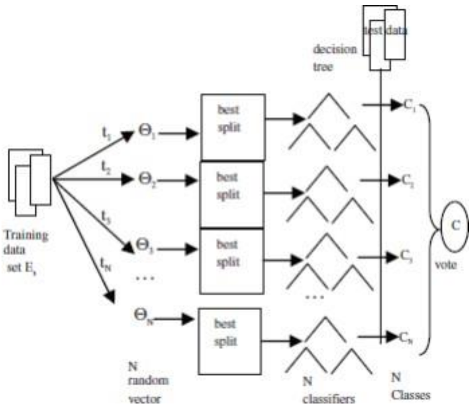


Figure 2.2: Random Forest Algorithm's Architecture (Aung and Hla, 2009).

The Random Forest algorithm has greater accuracy and is more reliable than the decision tree algorithm (Aung and Hla, 2009). However, the Random Forest algorithm does not always perform well in every scenario, especially when it meets the class imbalance instances (Segal, 2004). This problem can be solved by the additional class weighing parameters, but at the same time, it will make the evaluation process complicated, and the Random Forest algorithm becomes not feasible.

2.4.2 Long Short-Term Memory (LSTM)

LSTM was introduced to address the drawbacks of RNN by integrating additional interactions per memory cell (Hochreiter and Schmidhuber, 1997). LSTMs can gain long-term dependencies and remember that information for an extended period. An LSTM model is constructed in a chain structure form. The LSTM has its repeating memory cells in the different structures from the single neural network in an RNN, as shown in Figure 2.3. Each memory cell contains four interacting layers with special communication approaches, as shown in Figure 2.4.

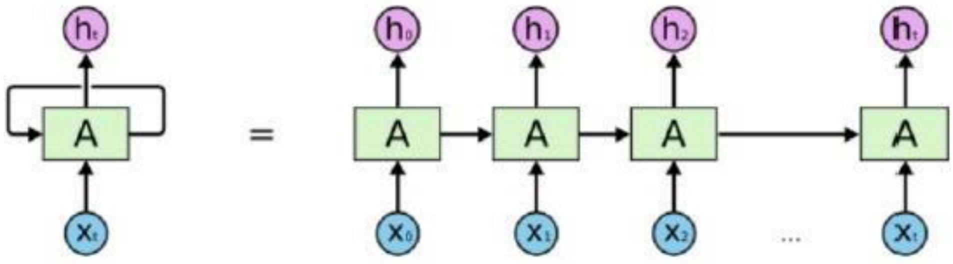


Figure 2.3: Single Neural Network in RNN (Le et al., 2019).

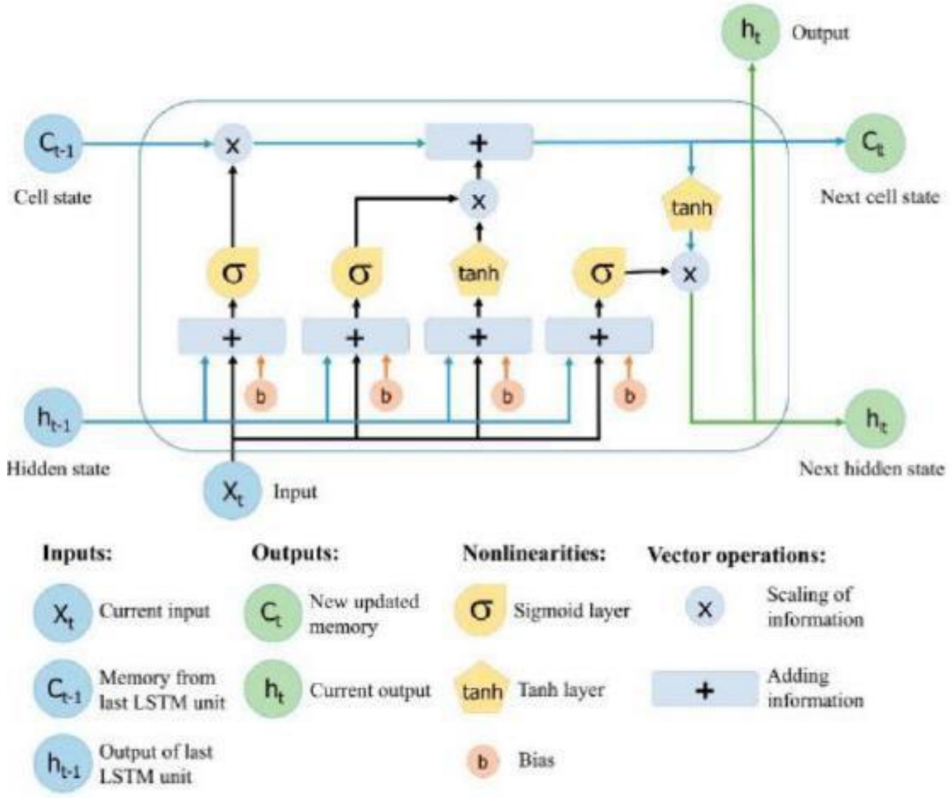


Figure 2.4: The Structure of LSTM (Le et al., 2019).

The structure of an LSTM neural network is some repeating memory cells, as shown in Figure 2.4. Two states are transferred from the previous cell to the succeeding, which are the hidden state and the cell state. The function of the cell state is making the information transferred forward unchanged, but some linear transformations in the data flow may occur. The data can be eliminated or updated through the sigmoid gates in the memory cells. A gate works as various kinds of matrix operations based on various weights.

Several steps are needed to construct an LSTM neural network. First and foremost, the unnecessary information will be identified and eliminated from the memory cell. This process is determined by a sigmoid function based on the present input at a certain time and the product of the previous memory cell at the previous time. That function decides the previous output's parts that are needed to be removed. This gate is known as forget gate and its values range between 0 and 1.

Secondly, the information from the new input is decided whether it needs to be saved in the cell state, and then the cell state is also updated. This step has two layers, which are the tanh layer and the sigmoid layer. A sigmoid layer has the same function as the first step, where it can decide whether the new information should be ignored or updated by giving out a value of 1 or 0. On the other hand, the tanh layer decides the importance level of the new information between -1 to 1 by giving the values that are passed by a weight. After that, these two types of values will be timed to give a piece of new information to the current cell state. Then, the recent memory will be summed with the previous memory to produce new memory.

Last but not least, there is a final step where a sigmoid layer will first decide the cell state's parts that should become the output cell state. The sigmoid gate's product is then timed with a value resulting from the tanh layer of the cell state. The range of the new output value is between -1 and 1.

2.4.3 Gated Recurrent Unit (GRU)

GRU neural network is an easier version of the LSTM neural network. Although they have significant similarities in their structure to solve the vanishing gradient problem in a typical RNN, there are some differences between GRU neural network and LSTM neural network. The structure of the GRU module is illustrated in Figure 2.5 (Le, Ho and Lee, 2020).

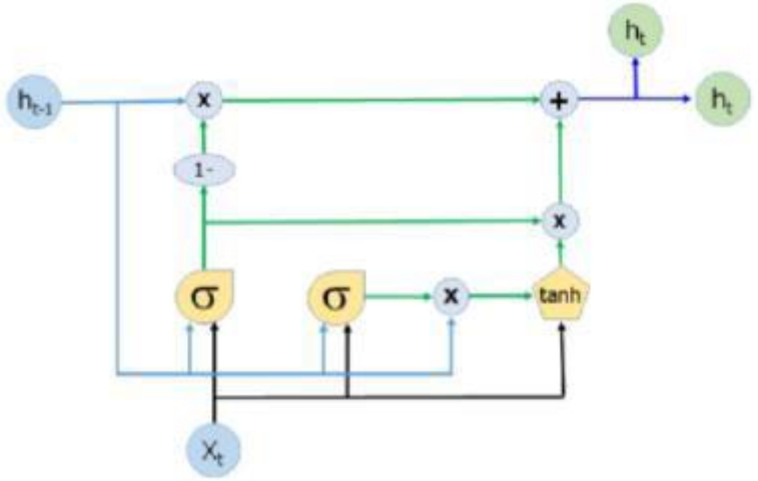


Figure 2.5: A GRU Module's Structure (Le, Ho and Lee, 2020).

The cell in the GRU model is different from the architecture of the LSTM cell, where it has no separate memory cell(Junyoung et al., 2014). Besides that, the GRU has two gates. Firstly, the reset gate determines the amount of data from the previous memory that is needed to be forgotten. Secondly, the update gate determines the amount of data from the previous memory needed to be transferred to the future.

The performance and convergence speed of the GRU model is sometimes greater than the LSTM model. Besides that, the GRU model's training is also easier than the LSTM model since there are lesser gating classes and parameters in the GRU model's training (Junyoung et al., 2014).

2.5 Network Topology

The influencing factor of network resilience includes network topology. There are several basic network topologies, such as point-to-point topology, ring topology, bus topology, star topology, and the list goes on. Besides that, there are some advanced network topologies based on the basic topologies' extension, such as mesh topology. The feasibility of the computer networkings on the project, which are star topology, mesh topology and hybrid topology network, are discussed in the following section.

2.5.1 Star Topology

In the star topology network, there is a central hub that connects all the end devices. The central hub plays the role of transmitting the data packet from one node to another node depending on the specified destination nodes. There is a serious issue that when the central hub fails or is broken, the whole network will also fail. This topology is commonly used in the scenario of a wireless access point (AP) forming a Wireless Local Area Network (WLAN). Figure 2.6 illustrates the structure of the star topology.

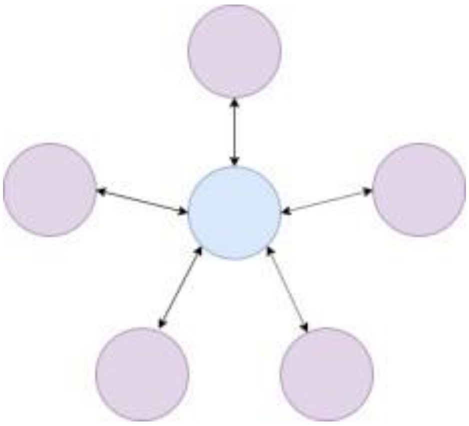


Figure 2.6: The Structure of Star Topology (Lim, 2021).

2.5.2 Mesh Topology

In the mesh topology, all the devices are connected. They can transmit data packets from one device to another device. A full mesh network is a network with which all devices are directly connected. On the other hand, a partial mesh network is a network that can still communicate with each other through relaying data packets, although not all the devices are directly connected. Since the mesh network always has an alternative route to relay the data packets even if some of the devices are failed or broken in their original path, the mesh network highly contributes to the aspects of the network resilience. Figure 2.7 illustrates the structure of the mesh network.

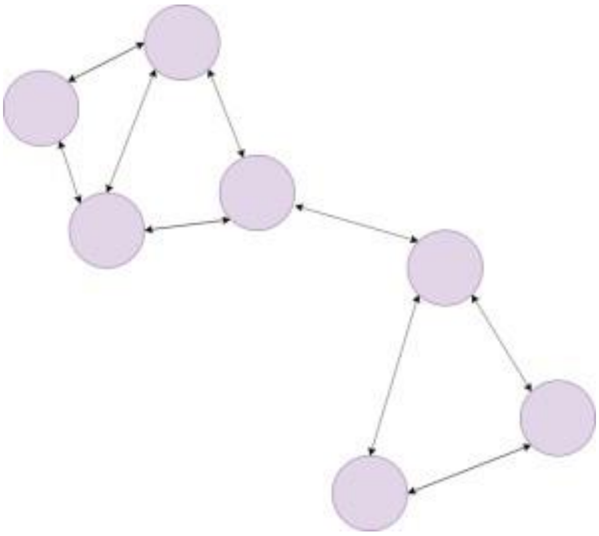


Figure 2.7: The Structure of Mesh Topology (Lim, 2021).

2.5.3 Hybrid Topology Network

There may be some scenarios that only one kind of network topology cannot efficiently solve. Therefore, a hybrid network is introduced in which different network topologies are joined together to build a larger and more efficient network for various kinds of complex scenarios. Figure 2.8 shows the structure of the hybrid topology network of a star network and a mesh network.

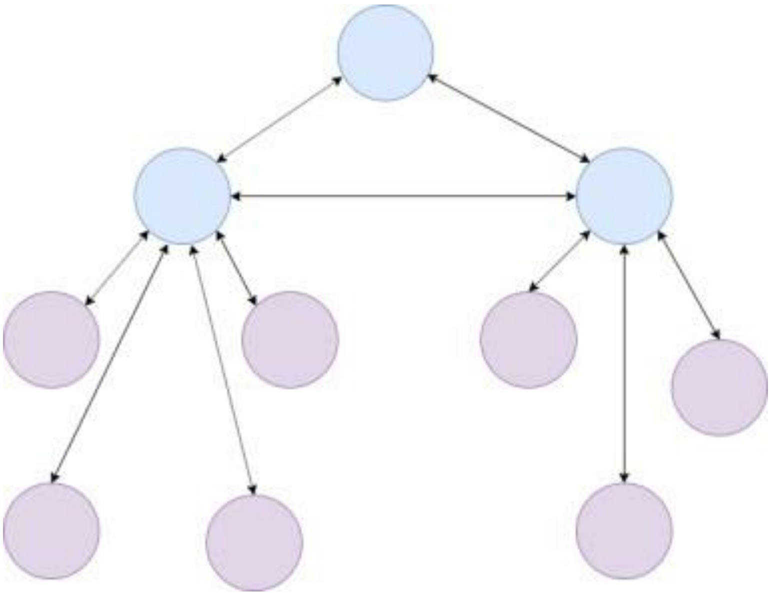


Figure 2.8: The Structure of Hybrid Topology (Lim, 2021).

2.6 Long Range (LoRa)

IoT has exponential growth in several application areas in these few years. However, most of the devices in IoT are connected through mobile communication networks such as Wi-Fi, Cellular data, Bluetooth and so on. These communication networks aim for the human's internet consumption, such as video streaming, file uploading and downloading as well as web browsing, which all need enormous bandwidth.

In the real scenario, it is estimated that 75 % of IoT devices, such as sensors, often transmit small data packets only and do not require a huge amount of bandwidth that consumes high power. In addition, IoT devices in remote areas rarely have sufficient power to support the high-power consumption of mobile communication network connectivity for a very long period. Therefore, Low Power Wide Area Network (LPWAN) is introduced. It is a wireless WAN technology used for the devices' low bit rates interconnection, which consumes low power and lower bandwidth only over a long range. The forefront of LPWAN is LoRa.

LoRa (Long Range) is an LPWAN modulation technique patented by LoRa Alliance, which is derived from Chirp Spread Spectrum (CSS) technology (The Things Network, 2022). LoRa is ideal for long-range transmission with relatively low bit rates. Data can be transmitted through LoRa at a wider range as compared to Wi-Fi and Bluetooth, which makes it suites for low-power remote applications such as sensors and actuators (The Things Network, 2022).

2.6.1 LoRaWAN

In industry, the wireless sensor network will decrease the signal's bandwidth or increase the signal's power when it transmits the data over a long range. Therefore, LoRaWAN is introduced. It is a class of wireless sensor network for the devices to sustain their network connection to send the data and have a prolonged battery life regularly. Besides that, it is also a standard protocol based on LoRa modulation (HIVEMQ, 2022). As shown in Figure 2.9, the LoRa-enabled sensors first broadcast the data packets in omni directional. After that, the LoRaWAN gateways will receive these data packets using their antennas. At the same time, the LoRaWAN gateways will also demodulate the LoRa data packets and send them to the LoRaWAN network server through mobile communication networks such as Wi-Fi, 4G, 5G and so on. In fact, the network server also manages the networks, such as the over-the-air activation (OTAA) process and the elimination of repeated messages. Furthermore, the network server will check the data packets and determine their target LoRaWAN application server. Finally, the application server will receive, decrypt, and process the messages with the application protocols. Moreover, the application server will send back some desired messages to the devices before.

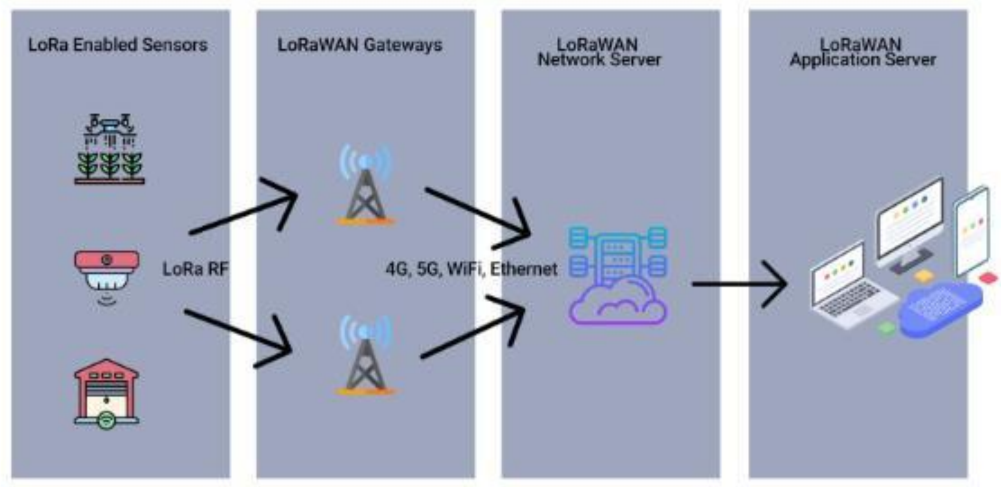


Figure 2.9: The Structure of LoRaWAN (HIVEMQ, 2022).

2.7 NerveNet

NerveNet is a resilient network developed by NICT in Japan. NerveNet is a specially developed network for the regional area to provide reliable network access and a stable, resilient information-sharing platform in emergencies, even if the base station is destroyed in a disaster. The base stations of NerveNet are interconnected by the Ethernet-based wired or wireless transmission systems such as satellite, Wi-Fi, LoRa and so on. They will form a mesh-topological network.

Nowadays, the current trend of the common network infrastructures uses the tree topology. As compared to it, NerveNet has the characteristic that it is more tolerant to the faults such as node failures, disconnections, destruction of the base station and so on. Since the base station in the NerveNet supports basic services such as SIP proxy, DNS, DHCP and so on, the NerveNet can also continuously provide connectivity services to the devices.

NerveNet has the property of database synchronisation. It uses a hearsay daemon to synchronise the database of every node within the NerveNet network. Hearsay daemon synchronises MySQL databases by updating the queries only and will not delete any actions when there is a lack of queries in another node's database. When the NerveNet node is connected to the NerveNet network, it will seek the difference in the table with other nodes.

After that, the database will be updated with the latest data. However, suppose all the NerveNet nodes are shut down. In that case, the data in the database will be deleted. Since all the existing databases are empty, they cannot relieve the data back by using the hearsay daemon synchronisation.

2.8 Summary

There are some basic requirements such as IoT, edge computing, Artificial Intelligence, mesh network, LoRa and NerveNet to design a reliable flood forecasting system. Edge computing and AI support an efficient and intelligent way to forecast the flood ahead of time by generating alert messages. After that, the NerveNet mesh network can help achieve fault tolerance during emergencies. Lastly, the LoRa-based end devices have the advantages of wide range connectivity coverage and low power consumption.

CHAPTER 3

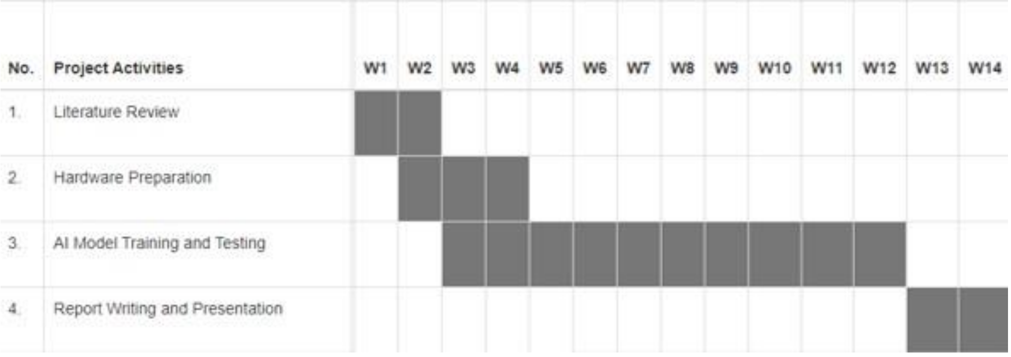
METHODOLOGY AND WORK PLAN

3.1 Introduction

This project aims to develop an edge AI flood forecasting resilient mesh network with NerveNet Wi-Fi and LoRa connection. The flood warning message and the forecasted flood water level graph are synchronised throughout the network. Case studies and literature reviews in Chapter 2 above are necessary to build the planned prototype.

3.2 Work Plan

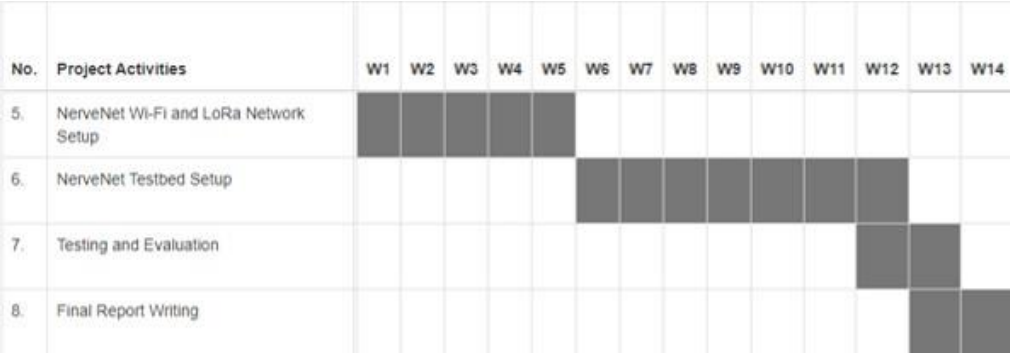
In part one of the Final Year Project, the main task is to study the knowledge and theory to build the whole system. As mentioned previously, the literature review includes the concept of IoT, edge computing, AI, mesh network, LoRa and NerveNet. Then, the hardware used is prepared. Next, the AI models used for the project are determined. The model training and testing are carried out in the cloud-based platform. Lastly, the progress report and presentation are also made. Figure 3.1 illustrates the Gantt Chart for Final Year Project part 1.



No.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
1.	Literature Review	■	■												
2.	Hardware Preparation		■	■	■										
3.	AI Model Training and Testing			■	■	■	■	■	■	■	■	■	■		
4.	Report Writing and Presentation													■	■

Figure 3.1: Gantt Chart for Final Year Project Part 1.

Part two of the Final Year Project's scope is to set up the NerveNet Wi-Fi and LoRa Network. The testbed is set up around the UTAR Sungai Long campus to test the network's performance. Then, the testing and evaluation are carried out by using the testbed. Finally, the final report is made. Figure 3.2 illustrates the Gantt Chart for Final Year Project part 2.



No.	Project Activities	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
5.	NerveNet Wi-Fi and LoRa Network Setup	■	■	■	■	■									
6.	NerveNet Testbed Setup						■	■	■	■	■	■	■		
7.	Testing and Evaluation												■	■	
8.	Final Report Writing													■	■

Figure 3.2: Gantt Chart for Final Year Project part 2.

3.3 Artificial Intelligence (AI) Flood Forecasting

In the real application, the AI model works upon the hydrological data collected from the sensors around the Sungai Long area, such as rainfall, river water level, humidity and so on, in real-time on the edge devices carried by the buses. In this project, five types of AI models, which are Random Forest, XGBoost, SVM, LSTM and GRU, are trained and tested on the prepared dataset to evaluate and record the performance of the system in flood water level forecasting.

The trained models are then compared, and the best AI model is transferred to the edge devices to forecast the real flood water level in the real scenario. The experimental setup is done in such a way that the dataset for the training and testing is scaled within the limits of the actual dataset (Samikwa, n.d.).

3.3.1 Study Site and Dataset

A watershed chosen as the study site of the project is the Abashiri River watershed, located northeast of Hokkaido, Japan, as shown in Figure 3.3. The area of the watershed is around 1380 km^2. It has a 115 km long main river to the North Pacific and a range of elevation from 0 m to 978 m (Kimura et al., 2019). In the project, the AI models are trained and tested using the datasets observed at the downstream stations called 'Hongou'. The used datasets are hourly datasets with the water level and rainfall variables from 1st January 2019 to 31st December 2020.

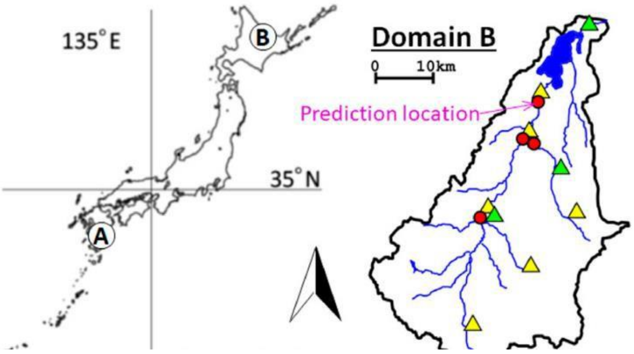


Figure 3.3: Abashiri River Watershed (Kimura et al., 2019).

3.3.2 Data Preparation

During data pre-processing, the rainfall and water level data undergo a train-test split, separated into 70 % of the data as a training dataset and 30 % as a testing dataset. The training data calculates the training process's error and estimates the AI models' parameters. The testing data provides an independent performance evaluation of the AI models after training (Sungai Bedup, Sarawak). As shown in Table 3.1, this project applies the hydrological dataset with one and five months for training and the remaining seven months for testing.

Next, the hydrological dataset has also undergone data standardisation where the values' distribution is rescaled to a mean value of 0 and a standard deviation value of 1. Data scaling is essential to fasten the training process of the AI model because the AI models can converge more rapidly if the dataset features are closer to the normal distribution. Before AI model training, the time series dataset is converted into sequential data with 24-time steps as the sequence length. The model performs equally well when the sequence length is between five to 15 or more. Therefore, in this project, the sequence length value of 24 is used in the model to represent 24 hours in one day.

Table 3.1: Training and Testing Period for the Dataset.

Dataset	Training	Testing
Hongou (January 2019 – December 2020)	January 2019 to May 2020	June 2020 to December 2020

3.3.3 Model Training

In this project, five AI models are used for the short-term prediction of river water levels. For three supervised learning models, Random Forest, XGBoost and Support Vector Machines, the output of these models is a river water level forecast by observing an ordinary multivariate dataset of water level and rainfall. On the other hand, for the deep learning models, LSTM and GRU, the input to these models is 24 hours time steps sequence of multivariate time series dataset of water level and rainfall, and their outputs are a forecast of river water level ahead of 1 hour.

3.3.4 Random Forest

The parameter 'max_depth' represents each tree's depth in the forest. In the project, the max_depth value is determined as 2.

3.3.5 XGBoost

There is a learning task parameter called objective. In the project, the learning objective is a regression with squared error. Besides that, the number of estimator is the number of runs that the XGBoost model tries to learn.

3.3.6 Support Vector Machines (SVM)

The parameter 'kernel' is the function the SVM model uses to solve problems. In the project, the kernel chosen is 'rbf'.

3.3.7 Long Short-Term Memory (LSTM)

There are several parameters in the LSTM model-building process. Firstly, the optimisation algorithm is the stochastic gradient descent procedure's extension to update the weights iterative of the network according to the training dataset. Besides that, it is also widely applied in implementing deep learning in natural language processing and computer vision. Secondly, an epoch is defined as the whole dataset transferring forward and backwards across the model's neural network once. Thirdly, the batch size is the number of samples propagating throughout the entire neural network (Le et al., 2019). Table 3.2 demonstrates the parameter values of the LSTM model.

Table 3.2: The Parameters' Values of LSTM Model.

Training Parameters	Details
Sequence Length	24
Optimisation Algorithm	Root Mean Squared Propagation (RMSProp)
Epochs' Amount	50
Batch Size	64

3.3.8 Gated Recurrent Unit (GRU)

In the project, there are the same parameter type and values as the LSTM model to construct the build and train the GRU models.

3.4 Application of the Trained AI Model

Figure 3.4 shows the overall process for applying the trained AI models in edge devices. In the development process, the models are trained in a cloud-based platform. Then, they are saved from being exported out in .h5 format. These trained models are then converted to the .pb format.

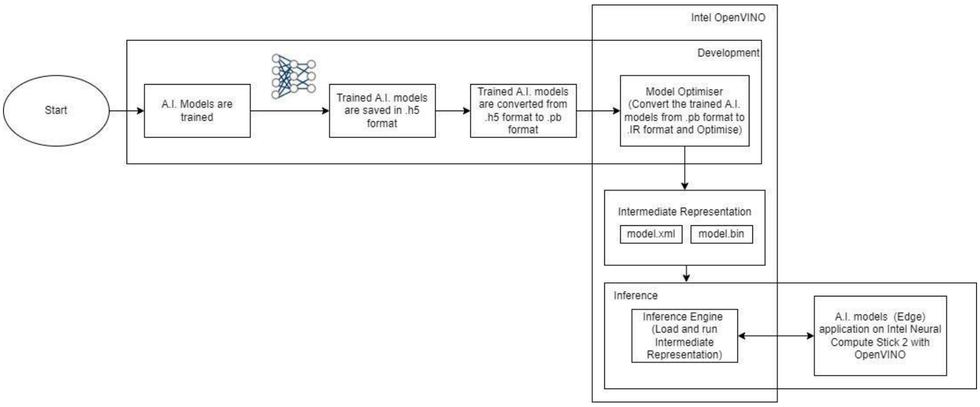

Intel OpenVINO
Development
Start
A.I. Models are trained
Trained A.I. models are saved in .h5 format
Trained A.I. models are converted from .h5 format to .pb format
Model Optimiser (Convert the trained A.I. models from .pb format to .IR format and Optimise)
Intermediate Representation
model.xml
model.bin
Inference
Inference Engine (Load and run Intermediate Representation)
A.I. models (Edge) application on Intel Neural Compute Stick 2 with OpenVINO

Figure 3.4: General Process for Trained AI Models' Application in Edge Device.

The OpenVINO toolkit is used to enable the faster running of the application of the AI models. There are two main components in the OpenVINO toolkit, which are the model optimiser and inference engine. Firstly, when the trained model in .pb format is fed into the model optimiser, it converts them to the .IR format. At the same time, it optimises the performance, space, and hardware-agnostic with conservative topology transformations. The output of the model optimiser is the trained_AI_models.xml and trained_AI_models.bin (Dubey Abhishek, 2020). Secondly, the AI inferencing process is performed at the inference engine, client edge devices which are Intel Neural Compute Stick 2 in the project, rather than straightly running the AI over the ordinary general-purpose hardware. Before feeding to the inference engine, the data is scaled using the scaler.gz exported from the training process. The scaled data is then reframed. The historical time series data representing the last 24 hours is extracted from the scaled dataset by retrieving the top 24 values of the rainfall and water level data. After that, the sequence data and the trained model in .IR format are fed into the inference engine to generate the water levels ahead of 1 hour in text form and the result graph in image form, as shown in Figure 3.5.

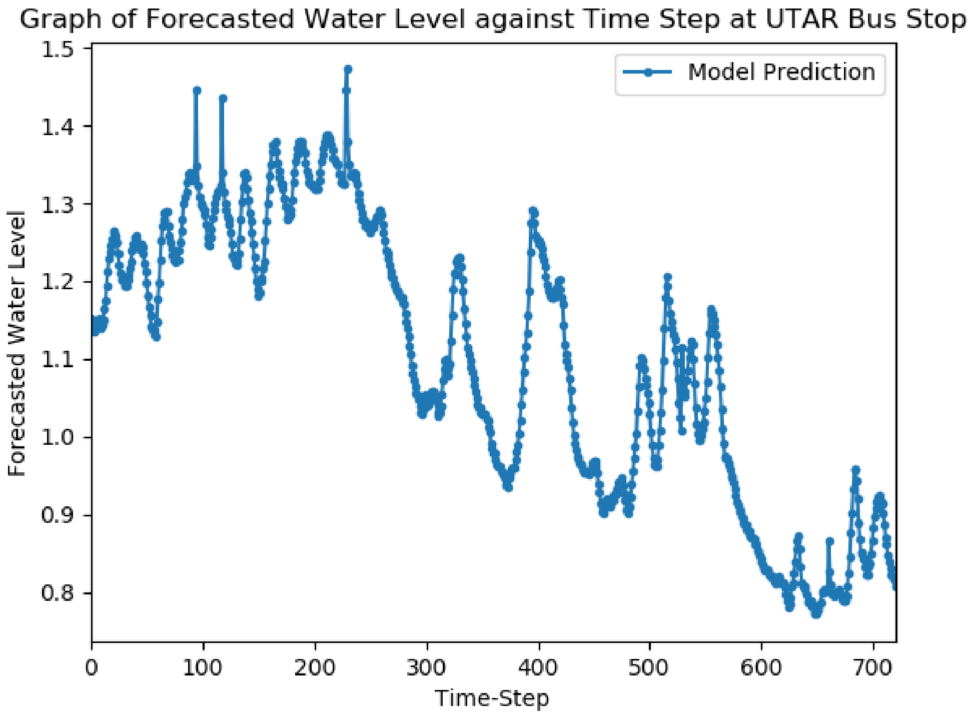


Figure 3.5: The Example Graph of Forecasted Water Level Result against Time Step.

3.5 GPS Tracking

Before the GPS tracking process, the bus locations at which the AI model is triggered to forecast the water level are determined. The GPS coordinates of those locations are determined by using Google Maps. In this project, there are three bus routes for three different edge nodes respectively which are Bandar Sungai Long & Palm Walk route 1, Bandar Sungai Long & Palm Walk route 2 and MRT Bukit Dukung Station Route 6. Each route has five different location points to cover the whole area of Sungai Long to forecast the whole area's water level. Figure 3.6 illustrates the determined location points in the Bandar Sungai Long & Palm Walk route 1, and the Bandar Sungai Long & Palm Walk route 2. Furthermore, the location points in the MRT Bukit Dukung Station Route 6 are shown in Figure 3.7. In addition, Figure 3.8 illustrates the determined location points in the Bandar Mahkota Cheras 1 Route 4.

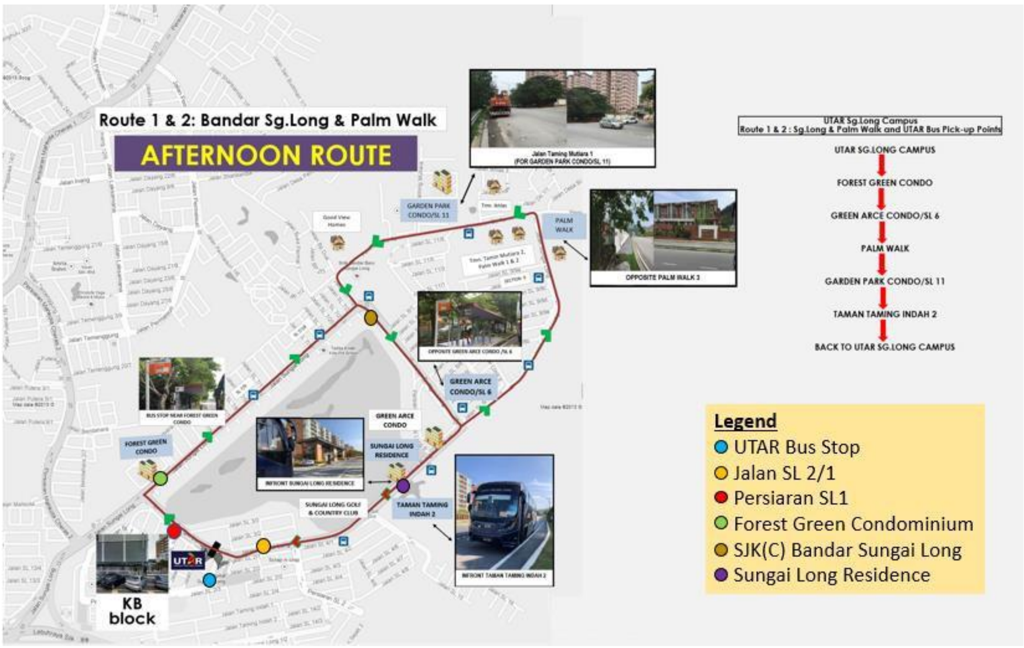

Route 1 & 2: Bandar Sg.Long & Palm Walk
AFTERNOON ROUTE
Jalan Taming Mutiara 1 (FOR GARDEN PARK CONDO/SL 11)
GARDEN PARK CONDO/SL 11
PALM WALK
OPPOSITE PALM WALK 3
OPPOSITE GREEN ARCE CONDO /SL 6
GREEN ARCE CONDO/SL 6
GREEN ARCE CONDO
SUNGAI LONG RESIDENCE
INFRONT SUNGAI LONG RESIDENCE
BUS STOP NEAR FOREST GREEN CONDO
FOREST GREEN CONDO
SUNGAI LONG GOLF & COUNTRY CLUB
TAMAN TAMING INDAH 2
INFRONT TAMAN TAMING INDAH 2
KB block
UTAR
UTAR Sg.Long Campus
Route 1 & 2 : Sg.Long & Palm Walk and UTAR Bus Pick-up Points
UTAR SG.LONG CAMPUS
FOREST GREEN CONDO
GREEN ARCE CONDO/SL 6
PALM WALK
GARDEN PARK CONDO/SL 11
TAMAN TAMING INDAH 2
BACK TO UTAR SG.LONG CAMPUS
Legend
● UTAR Bus Stop
● Jalan SL 2/1
● Persiaran SL1
● Forest Green Condominium
● SJK(C) Bandar Sungai Long
● Sungai Long Residence

Figure 3.6: Location Points in Bandar Sungai Long & Palm Walk Route 1 and Route 2.

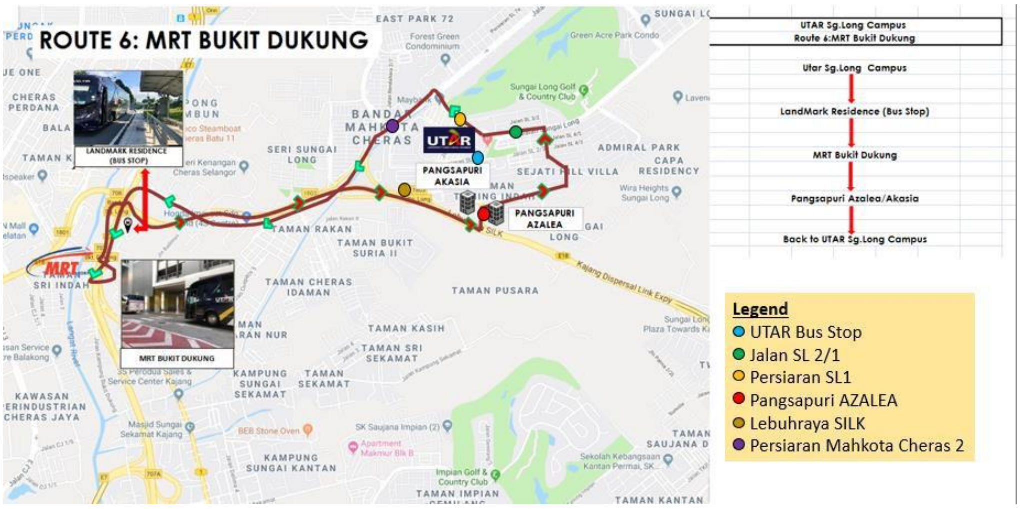


Figure 3.7: Location Points in MRT Bukit Dukung Station Route 6.

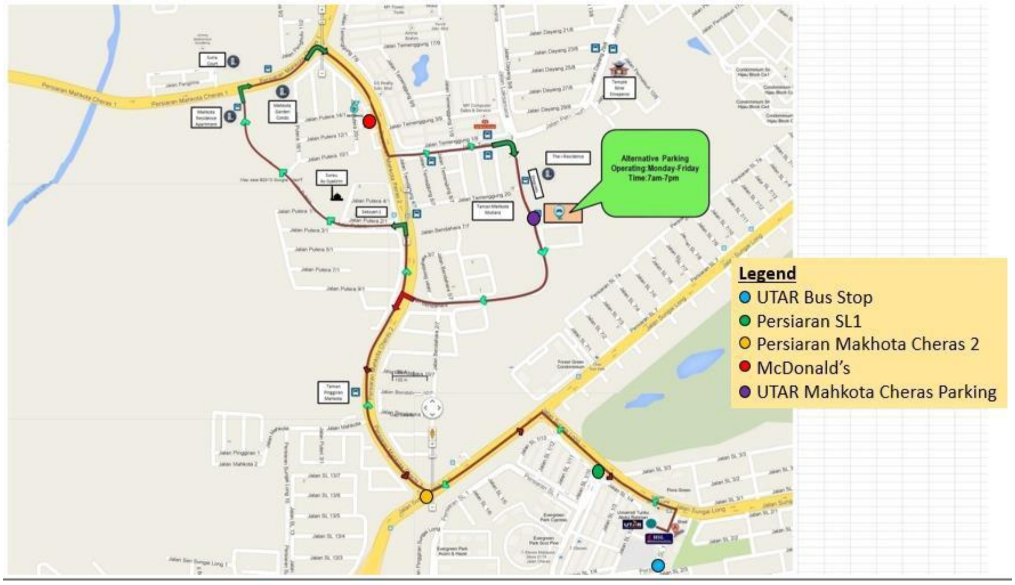

Alternative Parking Operating:Monday-Friday Time:7am-7pm
Legend
● UTAR Bus Stop
● Persiaran SL1
● Persiaran Makhota Cheras 2
● McDonald's
● UTAR Mahkota Cheras Parking

Figure 3.8: Location Points in Bandar Mahkota Cheras 1 Route 4.

The overall process flow for the GPS tracking process is shown in Figure 3.9. The GPS tracking process of each edge device is first initialised by extracting the GPS coordinates of the location points in the bus route in which the edge device will pass through it. Then, the real-time location coordinates of the edge device are read in the NMEA format via the GPS receiver at intervals. The edge device will continuously read its location's coordinates until it reaches the surrounding of one of the location points. When the edge device detects that it is within 111 m range of the specific location point, the historical time series data from the local dataset of that detected location point is retrieved by the edge device. This historical dataset is then checked to determine whether its amount is equal to or more than the number of steps which is 24-time steps in our AI models. Otherwise, the process will return to the step of continuously detecting the location coordinates. If the condition is fulfilled, the water level of that detected location is forecasted by applying the trained AI models in the edge device. This process repeats until the bus carrying the edge device has finished running its route.

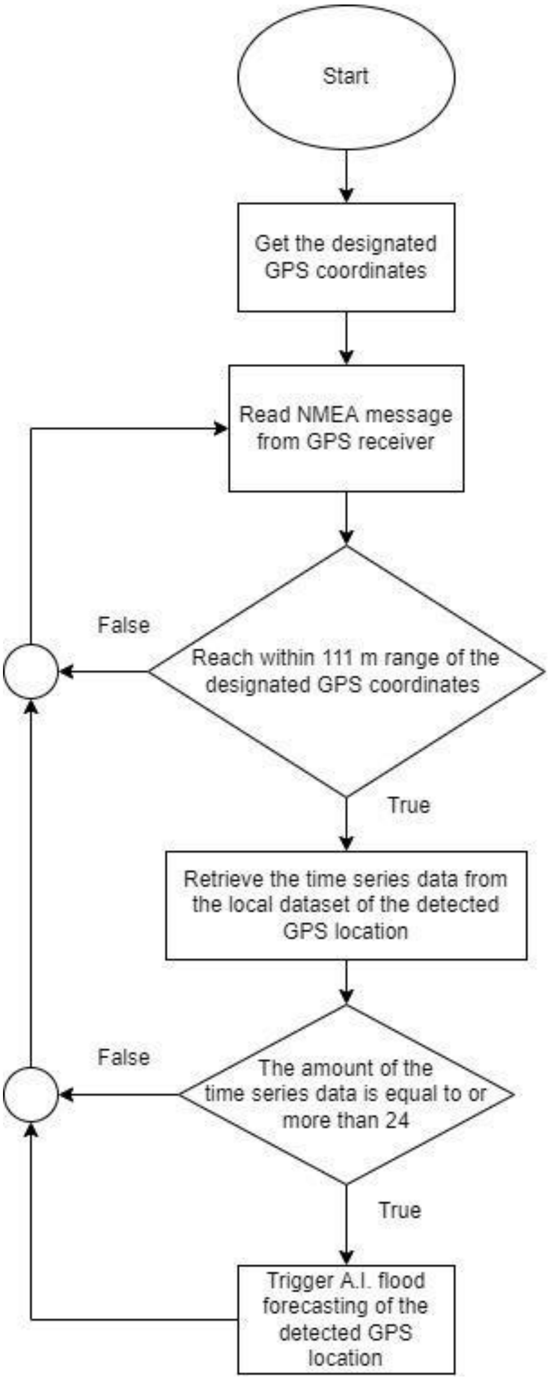

Start
Get the designated GPS coordinates
Read NMEA message from GPS receiver
False
Reach within 111 m range of the designated GPS coordinates
True
Retrieve the time series data from the local dataset of the detected GPS location
False
The amount of the time series data is equal to or more than 24
True
Trigger A.I. flood forecasting of the detected GPS location

Figure 3.9: General Process for GPS Tracking.

3.6 Job Scheduling

There are two types of AI flood forecasting model outputs: the forecasted water level ahead of time in text form and the result graph in image form. At the initial stage, the forecasted warning text message is transmitted from the edge devices to the subscriber side through the LoRa based on the order of the text generated. However, it is observed that the edge devices will repeatedly publish the same warning text message throughout the whole bus route if it does not get the acknowledgement from the subscriber side that it has already received the message packets. This disobeys the objectives and purpose of the project where the warning text message that is more important than the message before may not be able to be transmitted to the gateway or server in time, resulting in colossal damage loss by flood due to late information.

3.6.1 Create task

The proposed solution to this issue is priority-based job scheduling. There are two separate workflows of the job scheduling mechanism. Figure 3.10 illustrates the first workflow, which is creating the task. Firstly, after the AI flood forecasting model generates the forecasted water level ahead of time, the result will fall into a condition. If the forecasted water level is within the normal level, the workflow will return to the AI flood forecasting model to continue generating the next forecasted water level. On the other hand, if the forecasted water level is outside the normal level, a sending job with the priority. The job consists of several variables such as unique record id, publisher node id, subscriber node id, forecasted water level, the determined threshold level, the remaining time of flood occurrence, the forecasted flood location and the priority of the job. The determined threshold level is the severity of the forecasted river water level. Table 3.3 illustrates three levels of the severity of the forecasted water level.

Table 3.3: The Severity of The Forecasted Water Level(Department of Irrigation and Drainage, 2022).

Threshold Level	Description	Indicator
Alert	The river level is significantly higher than the normal river water level.	1
Warning	The river water level is approaching to the level that flood may occur and the public needs to prepare for the evacuation action.	2
Danger	The river water level can bring a considerable flood and the public needs to start to evacuate.	3

Besides that, the job's priority is defined as the urgency level of the message packets to be sent to the subscriber side. The priority of the job can be computed by the multiplication of the severity indicator and the remaining time of flood occurrence. After the job is created, it is stored in the job_list.csv. Finally, the workflow will return to the AI flood forecasting model and continue repeating the works.

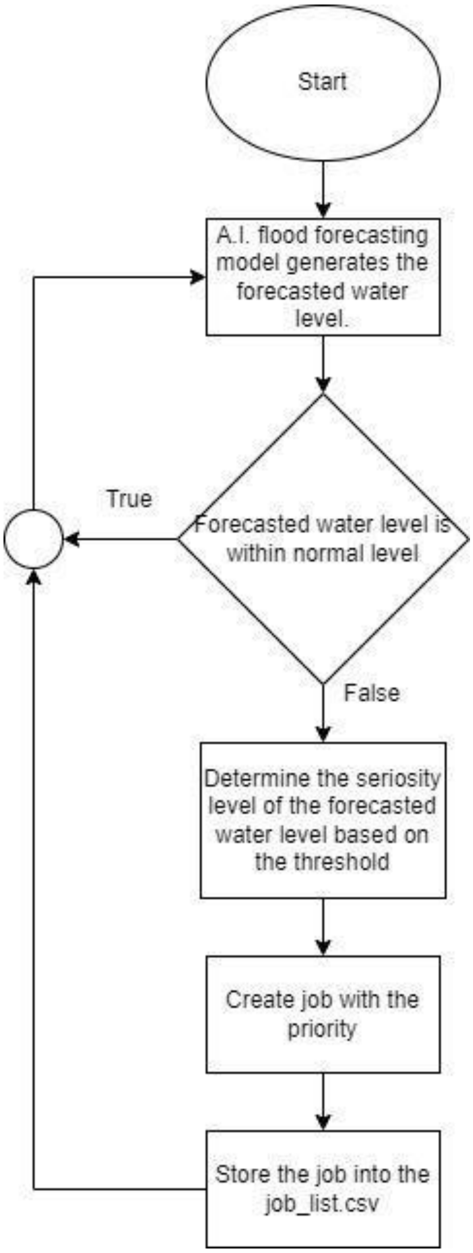

Start
A.I. flood forecasting model generates the forecasted water level.
Forecasted water level is within normal level
True
False
Determine the seriosity level of the forecasted water level based on the threshold
Create job with the priority
Store the job into the job_list.csv

Figure 3.10: The Workflow for Creating the Task.

3.6.2 Job Scheduling

Figure 3.11 illustrates the mechanism's second workflow, job scheduling. Firstly, the acknowledgement variable, which is used to determine whether the subscriber node receives the message packet or not, is always initialised to False. Secondly, the job_list.csv is read to get the job list. If the job list is empty, the edge device will display that the job is finished and is waiting for the new job. After that, the workflow will return to the step of initialising and resetting the acknowledgement variable to False. On the other hand, if there are jobs in the job list, the edge device will search for the job with the highest priority in the job list. Then, that particular job is transmitted from the edge device to the subscriber node through LoRa within 5 minutes. After that, the response or received data from the transmission on the edge device side is read. By using the regular expression (regex), the edge device will search for the "Completed" keyword in the received data, and it will set the acknowledgement variable to True if there is a "Completed" keyword in the received data strings.

If the acknowledgement variable remains as False, the workflow will return to the step of initialising and resetting the acknowledgement variable to False. On the contrary, if the acknowledgement variable is True, which indicates the subscriber node receives the message packets, the particular job will be removed from the job_list.csv. Then, the edge device will display that that particular job is sent successfully. Finally, the workflow will go back to the step of resetting the acknowledgement variable to False and continue repeating the works.

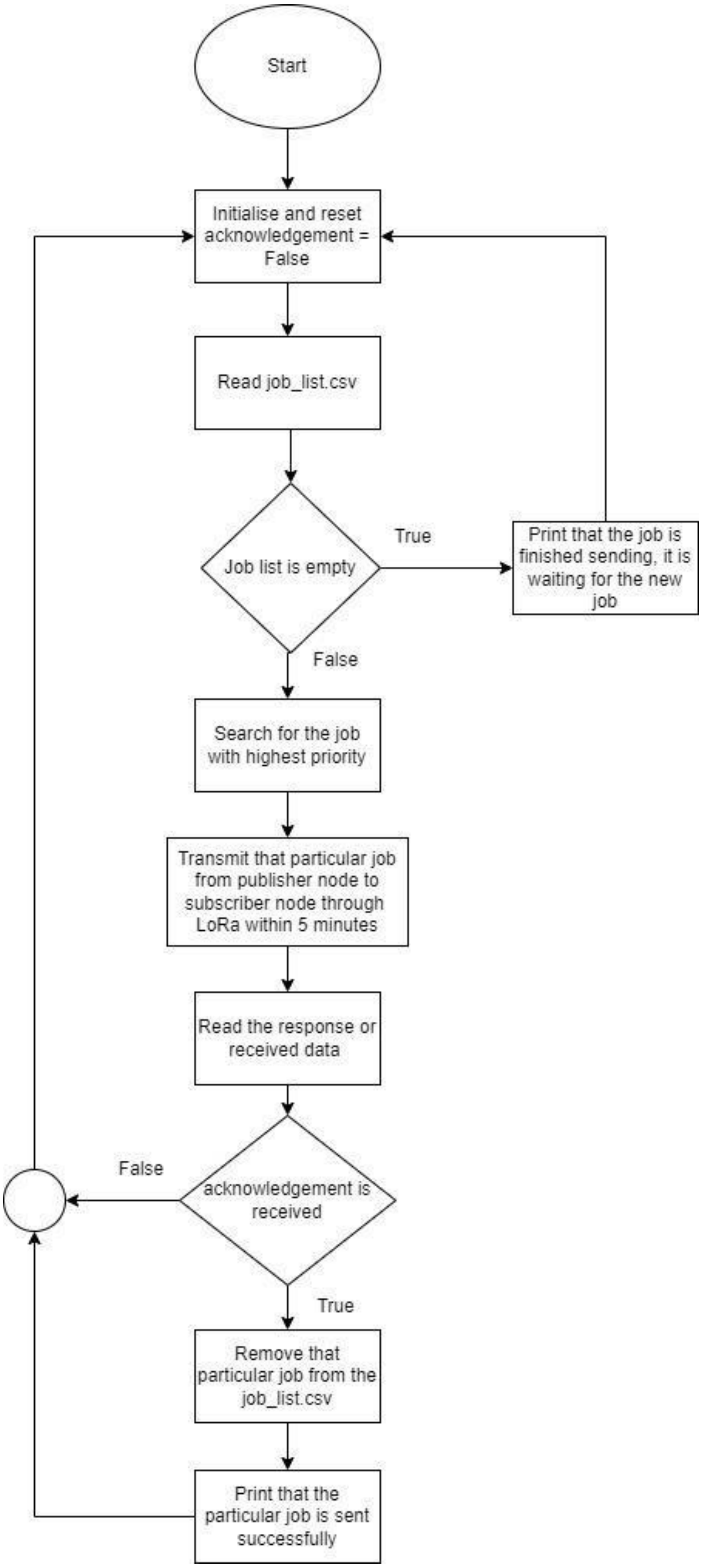

Start
Initialise and reset acknowledgement = False
Read job_list.csv
Job list is empty
True
Print that the job is finished sending, it is waiting for the new job
False
Search for the job with highest priority
Transmit that particular job from publisher node to subscriber node through LoRa within 5 minutes
Read the response or received data
False
acknowledgement is received
True
Remove that particular job from the job_list.csv
Print that the particular job is sent successfully

Figure 3.11: The Workflow of Job Scheduling.

3.7 NerveNet Wireless Mesh Network

A mesh network has a redundant connection for every device within the network. If the primary peer of the device is broken or failed, the devices will search for an alternative pathway to transmit the data packets to their destination.

3.7.1 Hardware Selection

The hardware chosen to deploy the network is listed.

i. 8 units of Archer T4U AC1300 Wireless Dual Band USB Adapter
ii. 4 units of Alfa Awus036ACH 1200mbps Wi-Fi Adapter
iii. 4 units of LoRa RFLink RM-92A + RFLink RM-92XUSB + 920MHz Whip Antenna ANT-92XA + Antenna Cable
iv. 2 units of USB GPS GlobalSat BU-353S4 G-STAR IV
v. 2 units of USB GPS DFRobot GPS Module
vi. 3 units Raspberry Pi 4 8 GB
vii. 5 units of Intel NUC BXNUC10i7FNH3 64 GB RAM 2 TB
viii. 1 unit of TP-Link TL-WN725N 150Mbps Wireless N Nano USB Adapter

3.7.1 Architecture of Hybrid Wi-Fi and LoRa Mesh Network

The overall architecture for the hybrid Wi-Fi and LoRa mesh network testbed is shown in Figure 3.12. As discussed earlier, after the bus reaches the designated locations, the edge devices which are node 201, node 202 and node 203, will run the AI flood forecasting inference on the Intel Neural Compute Stick 2, which outputs the flood alert text message and result graph image. Next, the flood alert text messages are then transmitted to the node 204 device located at the 8th floor's pantry through the LoRa route, and then the alert text message will be synchronised to every node in their own 'disaster_application' database. On the other hand, the result graph image is saved in node 201, node 202, and node 203 themselves until they are near the UTAR bus stop. The result graph images are then transmitted to node 210, located at the ground floor's pantry, through the Wi-Fi route, and then the result graph image will also be synchronised to every node in their own 'shbt_boxshare' database.

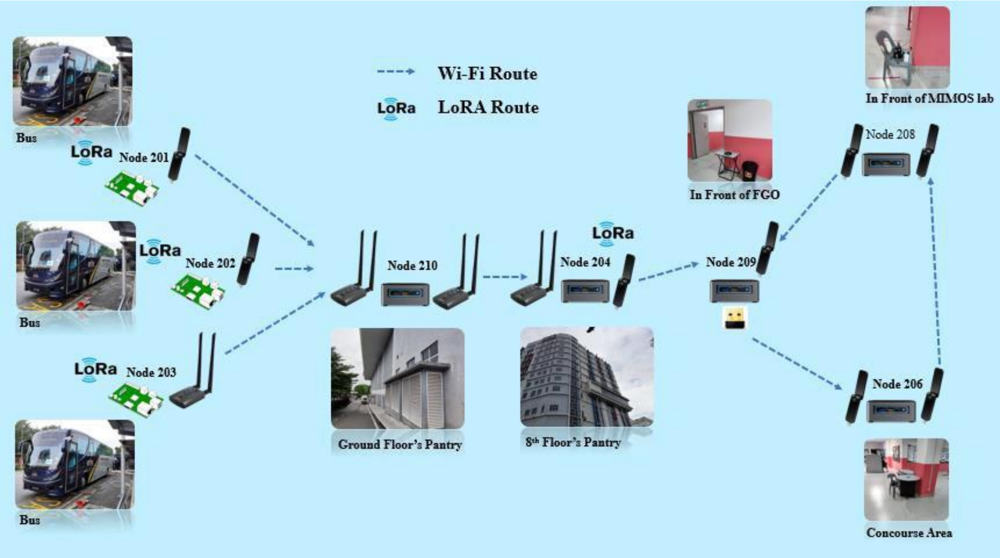


Figure 3.12: The Architecture of Hybrid Wi-Fi and LoRa Mesh Network.

During the construction of the architecture, it is observed that the Wi-Fi network between node 204 and node 210 is disconnected if the Archer T4U AC1300 Wireless Dual Band USB Adapters are used to provide the connectivity for node 204 and node 210. This is because the connectivity range provided by this type of adapter is insufficient to cover the long distance between node 204 and node 210. They are located at the ground pantry and 8th floor's pantry, respectively, where it is estimated that the distance between them is around 80 m if it is assumed that one floor has 10 m high. Therefore, the Alfa AWUS036ACH 1200 Mbps Wi-Fi Adapters are introduced. According to its datasheet, the AWUS036ACH has 802.11ac standards with hardware-based Wi-Fi optimisation that gives out an extra-large coverage and a strong strength to penetrate the walls so that the Wi-Fi dead spots are eliminated (ALFA NETWORK DISTRIBUTOR, 2010b). In addition, the external antennas of the AWUS036ACH are changed to Alfa APA M04 Accurate 7dBi Wi-Fi directional Antenna instead of its original five dBi omnidirectional antennas. As shown in Figure 3.13, the directional antenna concentrates its gain in a particular direction and has stronger strength of the signal in the direction it points to (ALFA NETWORK DISTRIBUTOR, 2010a). Thus, the Wi-Fi network between node 204 and node 210 can be easily connected, although the distance between them is long.

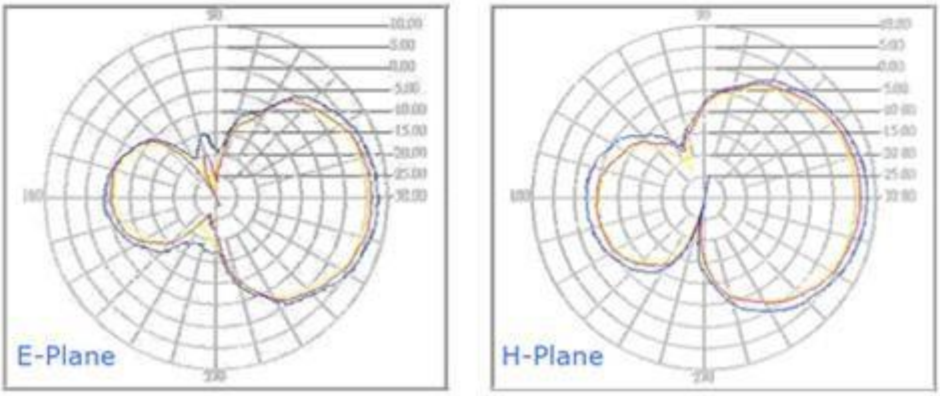


Figure 3.13: The Coverage of Directional Antenna (ALFA NETWORK DISTRIBUTOR, 2010a).

Moreover, the same problem goes to the Wi-Fi network between node 210 and the edge devices, which are node 201, node 202 and node 203. The Alfa AWUS036ACH 1200 Mbps Wi-Fi Adapters are also used at node 210 and node 203 to solve this issue. There is a difference which is the external antennas of the AWUS036ACH are changed to Alfa Wi-Fi Antenna ARS-N19 9 dBi Dipole Antennas. It has a higher gain than the original antennas and the directional antennas, as mentioned above, to give a stronger signal strength. Besides that, since the buses' rest place covers a large coverage area which is from the bus stop to the back door of the KB block, the omnidirectional antenna is preferable because it has its gain distributed to every direction, as shown in Figure 3.14. Thus, the Wi-Fi adapter can detect more signals overall from the buses that are distributed near and behind the KB block.

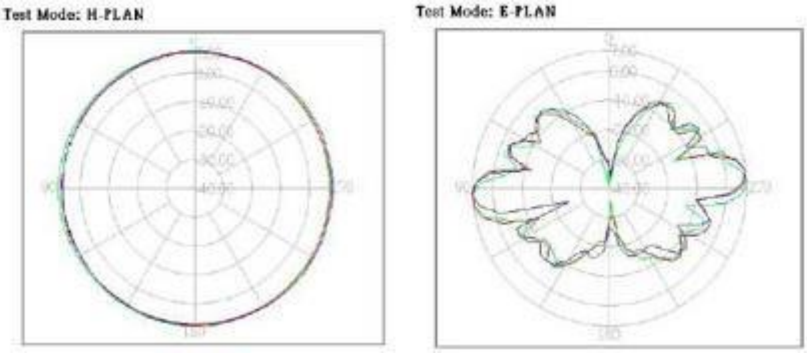


Figure 3.14: The Coverage of Omni-directional Antenna (ALFA NETWORK DISTRIBUTOR, 2010c).

3.8 Database Synchronisation

Wi-Fi mesh network framework provides database synchronisation to share common data within the mesh network. In NerveNet, a MySQL database is included. The lookup feature is built by using a service daemon called PTMGR (Path Tree Management Generation). It needs to be installed in the essential node within the mesh network. PTMGR continuously seeks for peers' network status to identify if any node is down or a new node has joined the network. If PTMGR steadily maintains the connection between nodes, the nodes could directly connect or access each other and perform NerveNet SQL database synchronisation. The nodes will compare the data rows within each other to update with the latest data.

3.9 Performance Evaluation

In the project, there are several types of performance evaluation, such as AI model performance evaluation, NerveNet x86 Wi-Fi Mesh performance evaluation, NerveNet LoRa messaging performance, image synchronisation test and text synchronisation test.

3.9.1 Mean Absolute Error (MAE)

The MAE is the mean of the differences between the original value with the forecasted value. On an excellent flood forecast, the MAE should be smaller. Moreover, it can be computed by the equation (3.1) (Adhikari and Agrawal, n.d.).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{3.1}$$

3.9.2 Mean Absolute Percentage Error (MAPE)

The MAPE is the percentage of the mean of the total error. On an excellent flood forecast, the MAPE should be smaller. In addition, it can be computed by the equation (3.2) (Adhikari and Agrawal, n.d.).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{e_i}{y_i}\right| x100 \tag{3.2}$$

3.9.3 Root Mean Squared Error (RMSE)

RMSE is the square root of the average squared deviation of the forecasted flood water level value. On an excellent flood forecast, the RMSE should be smaller. In addition, it can be computed by the equation (3.3) (Adhikari and Agrawal, n.d.).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {e_i}^2} \tag{3.3}$$

3.9.4 R Squared (*R^2*)

R squared is the coefficient of determination and goodness of fit. With an excellent flood forecast, the R^2 should be larger. In addition, it can be computed by the equation (3.4) (Adhikari and Agrawal, n.d.).

$$R^2 = 1 - \frac{sum\ squared\ regression\ (SSR)}{total\ sum\ of\ squares\ (SST)} \tag{3.4}$$

3.9.5 Latency

The latency of the NerveNet Wi-Fi network is benchmarked using the ping command. Ping uses the Internet Control Message protocol. The ping command will output the minimum, maximum, and average latency in milliseconds in the terminal after repeating the ping 100 times. In the performance evaluation, the average latency is taken to evaluate the NerveNet Wi-Fi network.

3.9.6 TCP/UDP Throughput

The TCP and UDO throughput of the NerveNet Wi-Fi network is benchmarked by using Iperf3. For TCP throughput, 100 packets are sent to the receiver node. After that, throughput is obtained in milliseconds by dividing the total data size received at the receiver side by the time taken. For UDP throughput, the sender bandwidth is set at 50 MBps. By default, the duration to send dummy data is 10 seconds. Therefore, 500 MB of data will be sent in total. The UDP packet receiver node will not acknowledge the sender node if the packet is lost in the transmission. The UDP throughput is obtained by dividing the received data size by 10 seconds.

3.9.7 Jitter

In real-life applications, several influencing factors, such as queuing, congestions, slow network connection, and so on that, causes a certain delay between continuous packets, also called jitter. The jitter with a high value will reduce the performance of the NerveNet Wi-Fi mesh network and the user experience. The jitter is also obtained when measuring UDP throughput using the Iperf3 tool.

3.9.8 LoRa Messaging Performance

The NerveNet LoRa data transmission performance is evaluated by sending the LoRa message with 30 Bytes and 90 Bytes payload size ten times and 40 times at once, respectively. After that, the packet delivery ratio (PDR) of LoRa packets and the number of packets lost in each case are recorded. The packet delivery ratio of LoRa packets can be calculated with the equation (3.5).

$$PDR = \frac{Number\ of\ Packets\ Received}{Number\ of\ Packets\ Sent} \tag{3.5}$$

3.9.9 NerveNet Database Text Synchronisation

NerveNet database supports flood warning text message synchronisation to other nodes. The average time synchronising the flood warning text message in real scenarios is recorded. The average of the flood warning text message in the project is around 30 Bytes. Therefore, the database synchronisation throughput can be calculated by dividing the 30 Bytes by the average time taken to synchronise the flood warning text message.

3.9.10 NerveNet Database Image Synchronisation

The NerveNet database supports image synchronisation to other nodes. The average time synchronising the result graph image in real scenarios and different-sized images is recorded. Besides that, the database synchronisation throughput can be calculated by dividing the average number of image size with the average time taken to synchronise the image.

3.10 Summary

Firstly, five different AI models are trained and compared to get the best model for flood water level forecasting. After that, the best AI model is converted into .IR format by the OpenVINO model optimiser so that it can be applied in edge devices. When the edge device pasted on the bus reaches some designated location points, it will trigger the AI model to generate the forecasted water level and result image graph. The forecasted water level is then processed to generate the flood alert messages and is shared among node 204 through LoRa immediately. On the other hand, the result graph is stored in the edge device and shared to node 204 through Wi-Fi when the edge device is close to the UTAR KB block. All the essential data is synchronised in the databases among each node by using the NerveNet mesh network. The performance of the AI model, NerveNet Wi-Fi and LoRa network and the database synchronisation are discussed in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The performances of five types of Artificial Intelligence models, which are Random Forest, XGBoost, SVM, LSTM and GRU, are evaluated with the testing dataset. In addition, the NerveNet Triangular Wi-Fi mesh network performances within x86 (node 206, node 208, node 209) are evaluated. Furthermore, the NerveNet LoRa mesh MQTT messaging performance within x86 and armhf where node 201, node 202 and node 203 act as the publisher side while node 204 acts as the subscriber side. Figure 3.12 shows Wi-Fi links and LoRa links for the whole architecture.

Moreover, the database synchronisation test in terms of text and image is carried out within the whole testbed, as mentioned previously.

4.2 Artificial Intelligence Model Benchmark

There are many methods that can be used for time series forecasting, and there is no clear winner. Model selection should always depend on the complexity of the data and the objectives that want to be achieved, such as the desired period ahead of the forecasting. Some models may be stronger against outliers. However, they may also perform worse than the more sensible ones, and they still are the best option based on the use case. Therefore, it is crucial to explore different kinds of methods for forecasting and their performance in various metrics to solve the problem of determining the suitable and optimal AI model for our case.

Figure 4.1 demonstrates the water level forecasting performance of five AI model types, which are Random Forest, XGBoost, SVM, LSTM and GRU on the proposed testing datasets. Theoretically, the deep learning methods outperform the conventional machine learning methods when the big data comes into its input (Mahapatra, 2018). This is similar to the project's case, where the LSTM and GRU models have a lower value of MAE, MAPE and RMSE than the Random Forest, XGBoost and SVM models. This indicates that the deep learning models have a lower deviation of the forecasted results from the ground truth and a lower error percentage. A higher R2 value indicates a more excellent time series forecasting performance from the deep learning models. As shown in Figure 4.2, although the conventional machine learning' performance is preferable when the data size is small, the deep learning method has an incremental performance when the data size is getting larger (Mahapatra, 2018). In the project, a huge amount of 2-year time series hydrological data comes into the inputs of the AI models. Therefore, the LSTM and GRU in the project have much higher performance than the conventional machine learning methods.

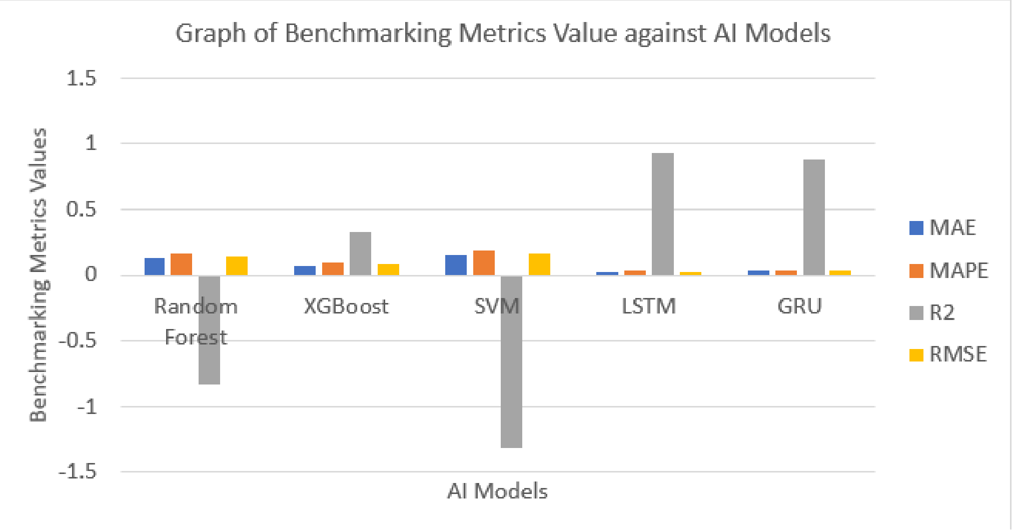


Figure 4.1: Benchmarking Metrics Values of Five Types of AI Models.

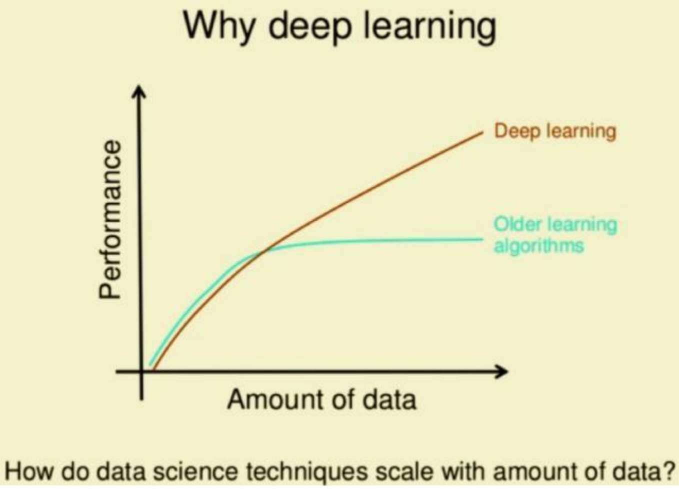


Figure 4.2: The Relationship Between the Amount of Data and the Performance of the Deep Learning and Conventional Machine Learning (Mahapatra, 2018).

Furthermore, this is also because the core of conventional machine learning is simple, and its training needs a plethora of human intervention and domain expertise if there is an error. On the other hand, the deep learning method learns about the high-level features in the data by themselves via the neuron network in an incremental manner. Deep learning itself maps the task as a hierarchy of concepts where a series of easier concepts defines each complex concept. It does not require human intervention and domain expertise (Shchutskaya, 2018). Therefore, deep learning always outperforms when applied to complex problems such as time series forecasting in the project.

As shown in Figure 4.1, the LSTM model has more excellent performance than the GRU model since it has lower MAE, MAPE, RMSE and higher R2. This finding is consistent with the study by (Yang, Yu and Zhou, 2020), where the LSTM model performs better than the GRU model in the case of short text processing and large-size datasets. In the project, there is a huge amount of rainfall and water level dataset where both types of variables are short integers. They act as the inputs to the LSTM and GRU models. Therefore, it can be seen that the LSTM is more appropriate than the GRU models in these scenarios.

All in all, the LSTM has the best performance in the AI water level forecasting since it has the lowest MAE, MAPE and RMSE while the highest R2 among all the proposed AI models. Therefore, LSTM is chosen as the project's AI water level forecasting model. In addition, all the benchmarking metrics on the LSTM model is also within the acceptable range. Therefore, the objective of developing a lightweight AI disaster detection model is also successfully achieved.

4.3 NerveNet x86 Wi-Fi Mesh Benchmark

NerveNet x86 Wi-Fi Mesh Network is benchmarked with ICMP ping and iperf3. The benchmarking metrics of the network are latency, TCP throughput, UDP throughput, and jitter. The network performance evaluation encompasses mesh links only. The network performance evaluation in mesh link is implemented by powering on node 206, node 208 and node 209 within the Wi-Fi domain.

Latency is the primary benchmarking metric used to discuss the performance of the Wi-Fi network since latency is the influencing factor that affects the throughputs (Glonaldots, 2015). As shown in Figure 4.3, there are no significant differences in the latencies of the two route directions between node 206 and node 208. Besides that, the two opposite route directions between node 208 and node 209 are also almost similar. This can be explained by the Wi-Fi client interface and Wi-Fi AP interface for the route between node 206 with node 208 are using the Archer T4U AC1300 Wireless Dual Band USB Adapter. Therefore, the routes between node 206 and node 208 have the same gain and transmission rate to send data to its destination and back to the source in almost equal time. The same reason goes for explaining the latencies of the routes between node 208 and node 209. However, the route from 209 to node 206 is higher than that from 206 to 209. This is because the Wi-Fi client interface in node 209 uses TP-Link TL-WN725N 150Mbps Wireless N Nano USB Adapter, while the Wi-Fi AP interface in node 206 uses Archer T4U AC1300 Wireless Dual Band USB Adapter, which has a higher gain and transmission rate to send data to its destination and back to the source in a shorter time.

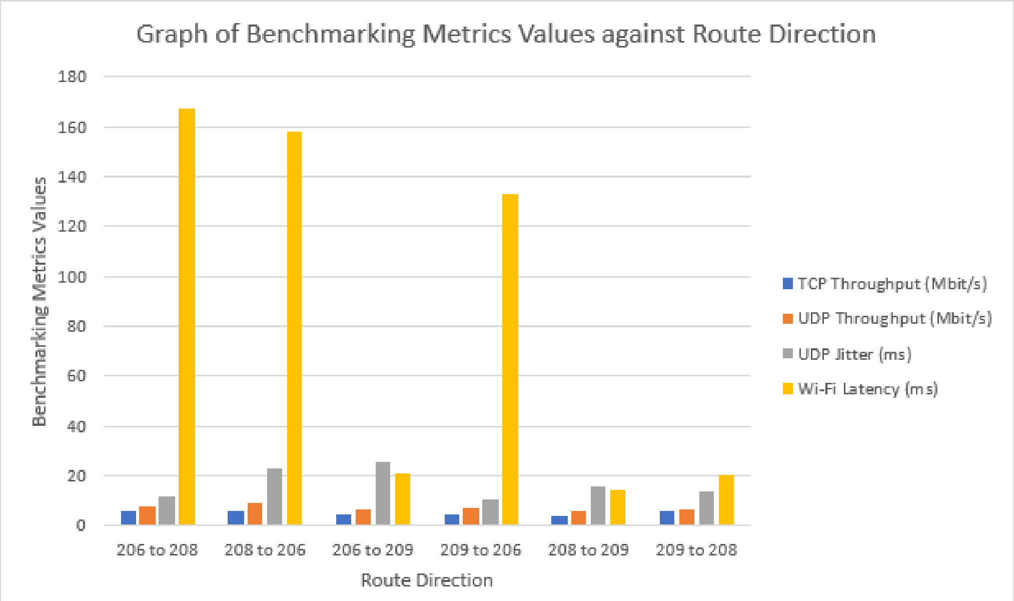


Figure 4.3: Benchmarking Metrics Values of NerveNet x86 Wi-Fi Mesh Network.

The shorter the distance between sending and receiving end, the greater the network latency (Glonaldots, 2015). Therefore, the latencies of the route between node 208 and node 209 are the lowest since it has the shortest distance between the sending and receiving end, as shown in Figure 4.4. Although the route between node 206 and node 208 has a shorter distance as compared to the route between node 206 and node 209, the route between node 206 and node 208 still takes a longer time to send data to its destination and back to the source as the data needs to penetrate the thick concrete wall of the staircase. Therefore, the latencies of the route between node 206 and node 208 are the highest. In addition, the route between node 206 and node 209 has latency with intermediate values.

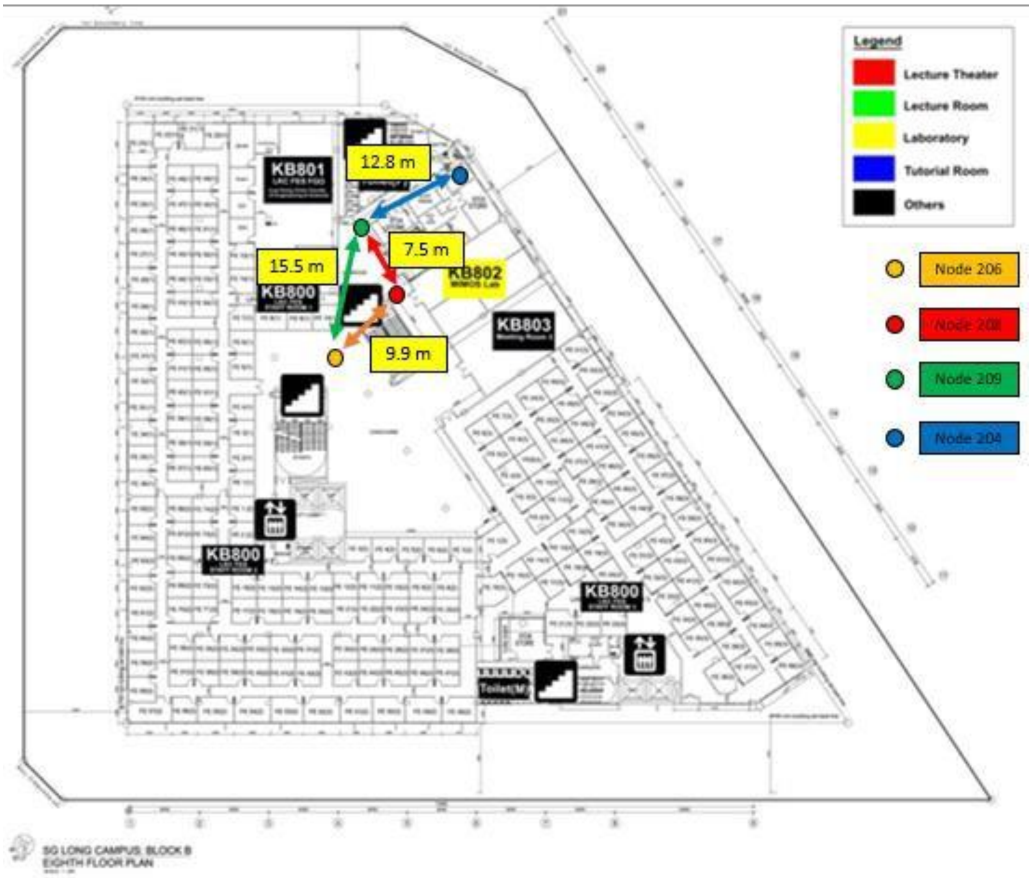


Figure 4.4: The Distance Between the Nodes.

As shown in Figure 4.3, the UDP jitter of all the route directions is similar. The highest UDP jitter is 25.579 milliseconds, and the lowest UDP jitter is 10.610 milliseconds. This fulfils the QoS requirement of jitter for applications such as video conferencing, which is less than 30 milliseconds

(Khalifeh, Gholamhosseinian and Hajibagher, 2011). Hence, the NerveNet x86 Wi-Fi Mesh Network has the acceptable range of UDP jitter to handle the applications that need low jitter, such as providing VoIP services.

4.4 NerveNet LoRa Messaging Performance

The primary benchmarking metric of the NerveNet LoRa network is the number of lost LoRa packets since the testing of the NerveNet LoRa MQTT applies the Quality of Service (QoS) level zero. The QoS level zero guarantees best-effort message delivery but not a message delivery guarantee since the sender only transmits the message at most once (HIVEMQ, 2015). Therefore, the LoRa message packets may be lost during the transmission process. It is crucial to evaluate the number of lost LoRa packets to conclude the efficiency of the LoRa in the flood alert message transmission in the NerveNet partial mesh network testbed.

MQTT payload sizes of 30 Bytes and 90 Bytes are used to test the NerveNet LoRa MQTT messaging performance. Besides that, the influencing factors include the number of LoRa packets sent at once since it can affect the ratio of lost packets. Hence the number of LoRa messages published at once is varied at 10 and 40 messages. After the LoRa MQTT subscriber has not received any message for 20 minutes, the remaining LoRa packets are considered lost. After that, the number of lost message packets under different scenarios is recorded accordingly, and the packet delivery ratio is calculated using the equation (4.1).

$$PDR = \frac{Number\ of\ Packets\ Received}{Number\ of\ Packets\ Sent} \tag{4.1}$$

The test is carried out using node 204 as a LoRa MQTT subscriber, while node 201, node 202 and node 203 act as the LoRa MQTT publisher for the bus route of Bandar Sungai Long & Palm Walk, Bandar Makhota Cheras 1 and MRT Bukit Dukung respectively. Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8, illustrate the packet delivery ratio under different locations that have the LoRa packets received to study the efficiency of the NerveNet LoRa messaging performance based on the distance between the subscriber node and publisher node.

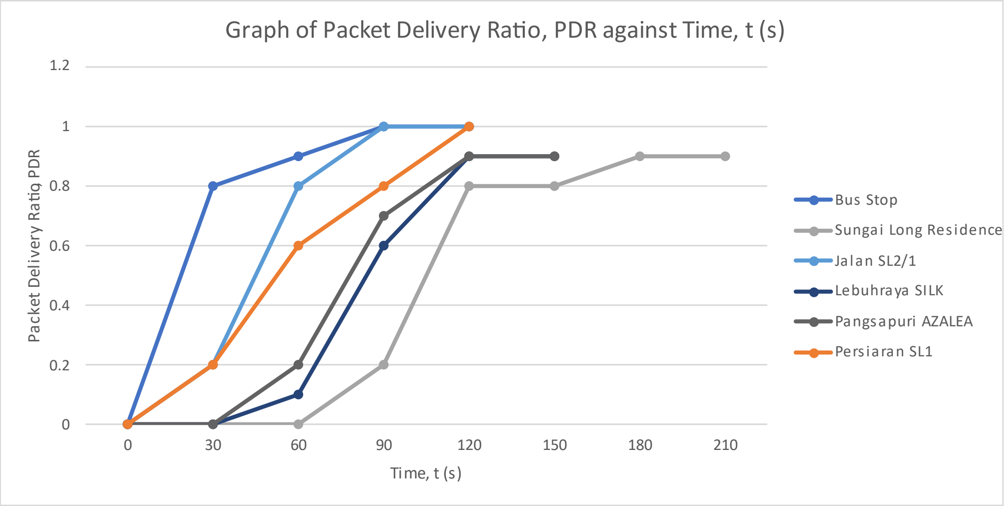


Figure 4.5: The Packet Delivery Ratio over Time for Scenario of 10 Messages with MQTT Payload Size of 30 Bytes Published at Once.

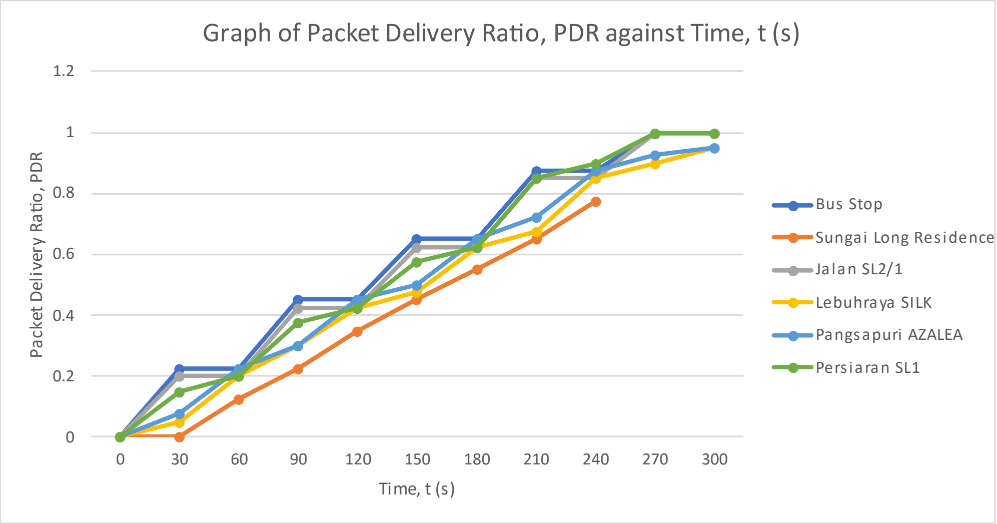


Figure 4.6: The Packet Delivery Ratio over Time for Scenario of 40 Messages with MQTT Payload Size of 30 Bytes Published at Once.

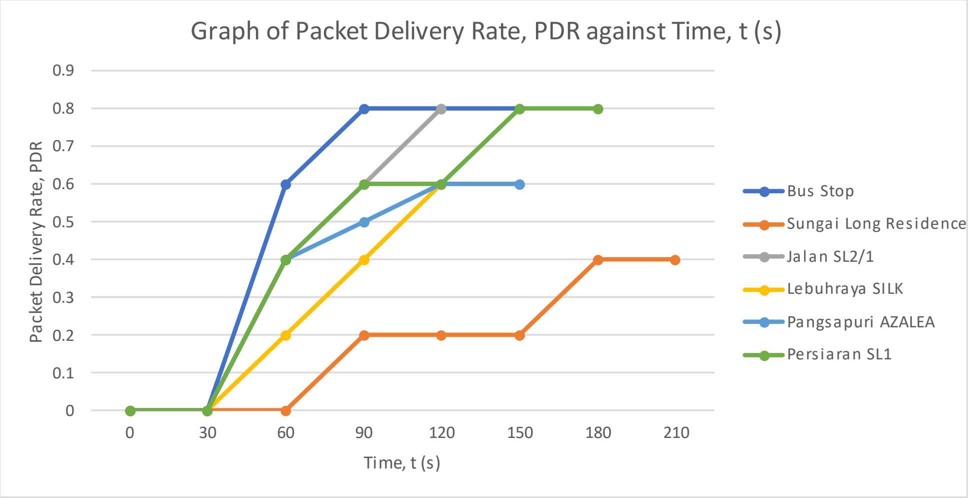


Figure 4.7: The Packet Delivery Ratio over Time for Scenario of 10 Messages with MQTT Payload Size of 90 Bytes Published at Once.

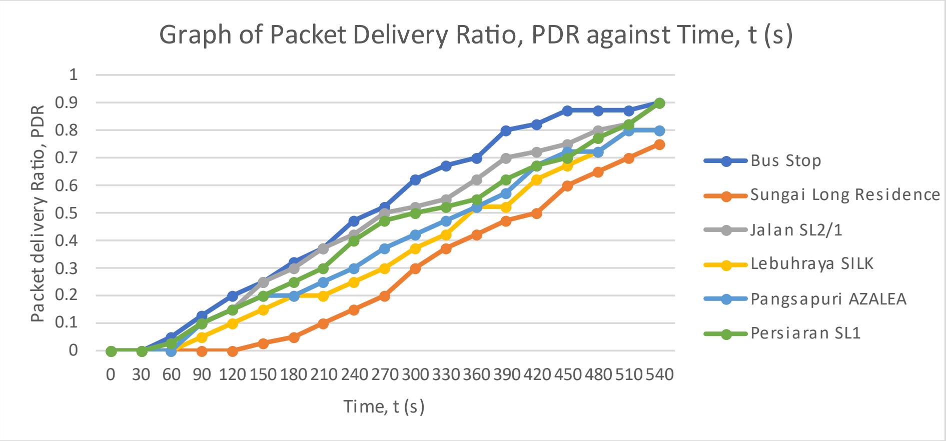


Figure 4.8: The Packet Delivery Ratio over Time for Scenario of 40 Messages with MQTT Payload Size of 90 Bytes Published at Once.

Theoretically, when the distance between the nodes and gateway increases, the packet delivery rate through LoRa decreases. Besides that, the obstacles could also influence the packet delivery rate (Choi, Lee and Lee, 2020). Those observations are similar to our cases. As shown in Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8, it can be seen that the further the distance between the publisher side and subscribe side, as shown in Figure 4.9, the lower the LoRa packet delivery rate. Furthermore, although the distance between the publisher node around the Pangsapuri AZALEA and Lebuhraya

SILK with the subscriber node is almost similar, the publisher node around Pangsapuri AZALEA still has a slightly higher LoRa packet delivery rate. It can be seen that the difference is attributed to the obstacles, such as trees around the Lebuhraya SILK area. The big change in humidity of the trees zone causes the decline of the packet delivery rate relying on the time and weather.

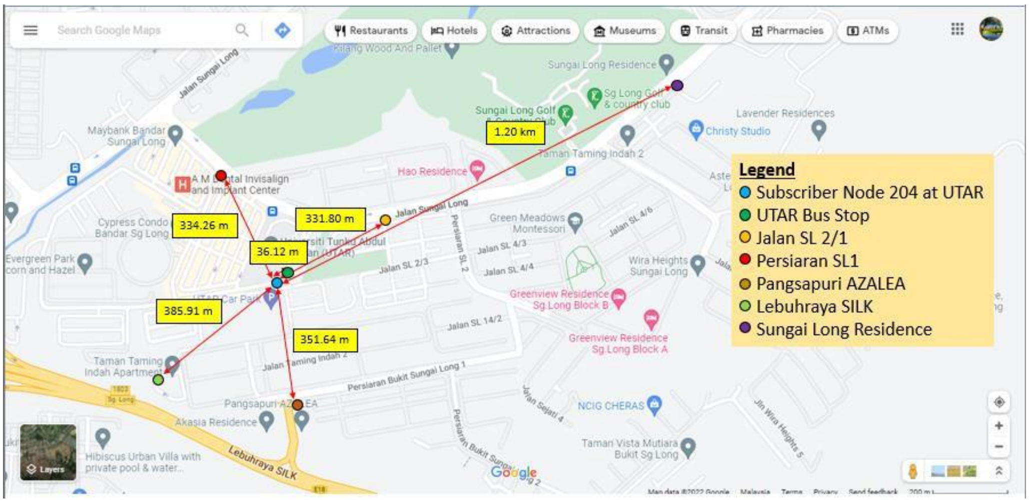


Figure 4.9: The Distance Between Subscriber Node and Publisher Node at Different Locations.

Besides the distance between the publisher node and subscriber node, the manipulating variable is LoRa MQTT payload size because the bandwidth is fixed. A larger payload size means a higher bit rate, increasing the risk of LoRa signals being interfered with or corrupted. As shown in Figure 4.10, the number of NerveNet MQTT LoRa messages lost increases when the payload size increases. Hence, it is concluded that the larger the LoRa MQTT payload size, the slower the LoRa packet transmission, and the higher the risk of the LoRa packet being lost. According to the European Telecommunications Standards Institute (ESTI), the number of lost packets of the LoRa is placed in the good category which is between 3 % and 14 % (Suharjono et al., 2018).

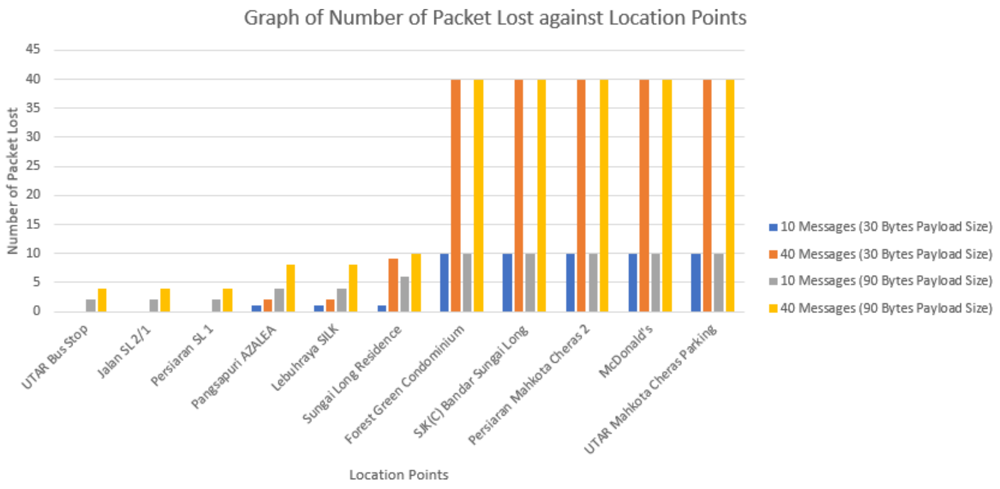


Figure 4.10: Number of LoRa MQTT Packet Lost at Different Location Points.

However, some of the location points, which are Forest Green Condominium, SJK(C) Bandar Sungai Long, Persiaran Mahkota Cheras 2, McDonald's and UTAR Mahkota Cheras Parking, have 100 % of LoRa packets lost and the packet delivery ratio of zero value. This is due to many obstacles around UTAR Sungai Long Campus, such as the thick concrete wall of the shop lots and condominiums and the electromagnetic interference from the residential areas that corrupt or interfere with the LoRa message packets during transmission. This decreases the transmission range of the LoRa packets. However, there is one exceptional location point which is Sungai Long Residence. The publisher node at that location can successfully send the LoRa packet message to the receiver node at the UTAR KB block, although there is a long distance between them which is 1.20 km. This is because the location point is a small hill with a clear space between that particular location point and the UTAR KB block, as shown in Figure 4.11. Therefore, it can conclude that if there are obstacles or interference between the subscriber node and publisher node, the NerveNet transmission range will be reduced. On the other hand, the NerveNet LoRa network can achieve long-distance transmission properties if there is a clear space between the subscriber node and publisher node.

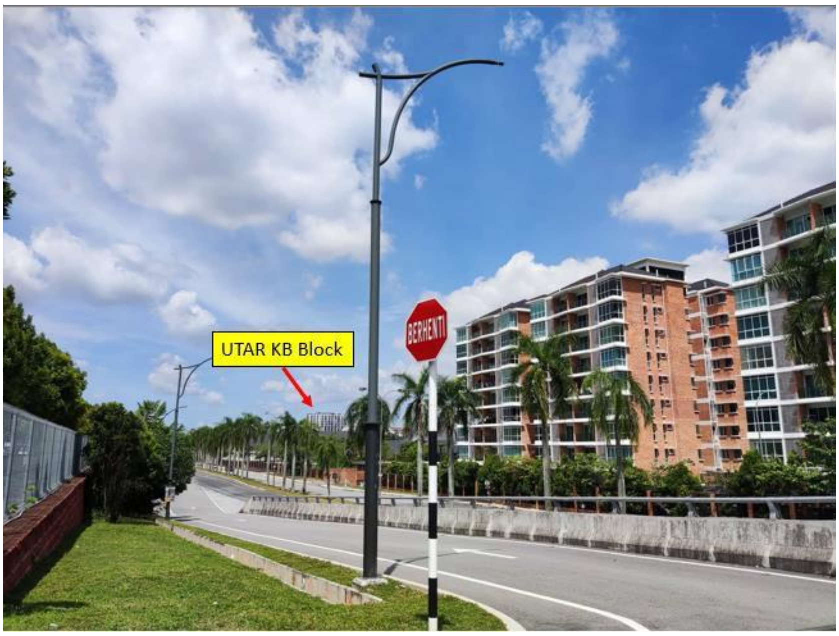


Figure 4.11: Clear Space between Sungai Long Residence with UTAR KB Block.

4.5 Text Database Synchronisation Test

The AI flood warning text message in the real scenario is used to study the efficiency of the text database synchronisation if the project is applied in real life. The average text size is around 30 Bytes depending on the number of characters in the location string and the water level string.

The texts start to be synchronised from node 204 since node 204 is the text receiver node for those three edge devices, which are node 201, node 202 and node 203. Each text is synchronised throughout the network several times. Then, the average time taken for each text to be synchronised in the database in each node is calculated to get more accurate results.

As shown in Figure 4.12, when the distance between the initial receiver node, which is node 204, with the afterwards receiver node increases, the time taken for each text to be synchronised in the database in each node increases. The maximum time taken for a text to be synchronised throughout every node is at most 30 seconds based on the network condition. Since the system only needs to transmit small-size flood warning text messages, it is sufficient for the system to synchronise the flood warning text message at a very fast speed so that the control centre can get the notifications from the gateway node and take action in the shortest time.

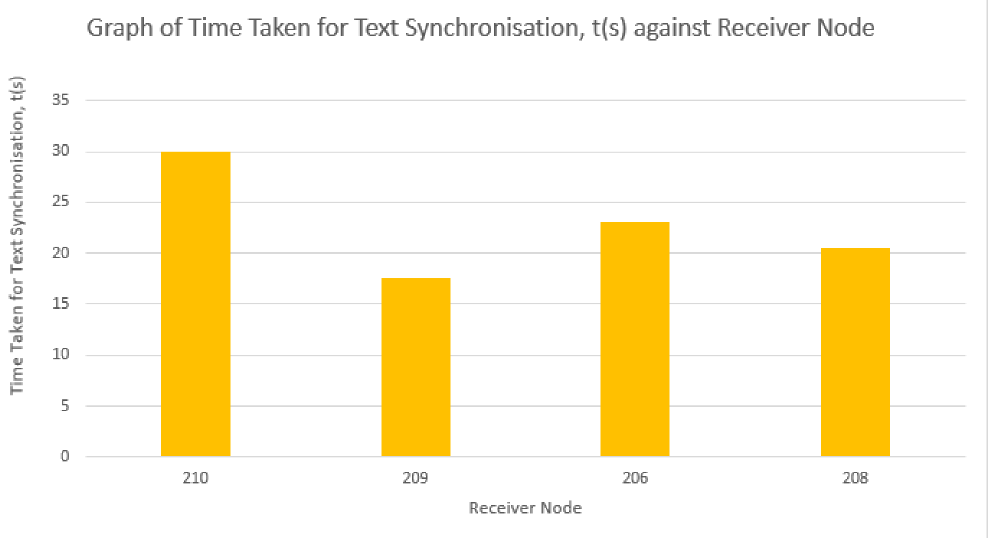


Figure 4.5.1: The Time Taken for Text Synchronised at Every Node.

4.6 Image Database Synchronisation Test

Three different sizes of images are used to measure the performance of the image database synchronisation test. Besides that, the AI output result graph image in the real scenario is also used to study the efficiency of the image database synchronisation if the project is applied in real life. The size of the images is tabulated in Table 4.1.

The images are started to be synchronised from node 210 since node 210 is the image receiver node for those three edge devices, which are node 201, node 202 and node 203. Each image is synchronised throughout the network several times. Then, the average time taken for each image to be synchronised in the database in each node is calculated to get more accurate results.

Table 4.1: Image File Size Used for Image Database Synchronisation Performance Benchmarking.

Image File Name	File Size
real_image.png	31.9 KB
image_small_size.jpg	47.4 KB
image_middle_size.jpg	745 KB
image_large_size.jpg	1.14 MB

As shown in Figure 4.13, when the image file size increases, the time taken for each image to be synchronised in the database in each node increases. The maximum time taken for the image with a size of equal to and less than 1.14 MB to be synchronised throughout every node is at most 30 minutes 55 seconds based on the network condition. Moreover, based on the test results, it can be observed that the result graph image requires at most 8 minutes 23 seconds for it to be fully synchronised in the real scenario. Therefore, since the system only needs to transmit a small size image after finished running a route, it is sufficient for the system to synchronise the images during the rest time of the buses at the UTAR bus stop because the minimum rest time of the bus is at least 15 minutes according to the schedule of the bus as shown in Figure 4.14.

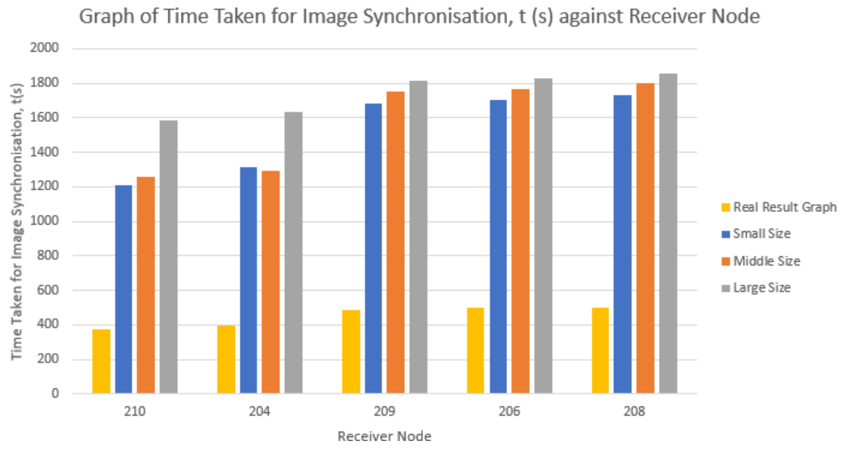
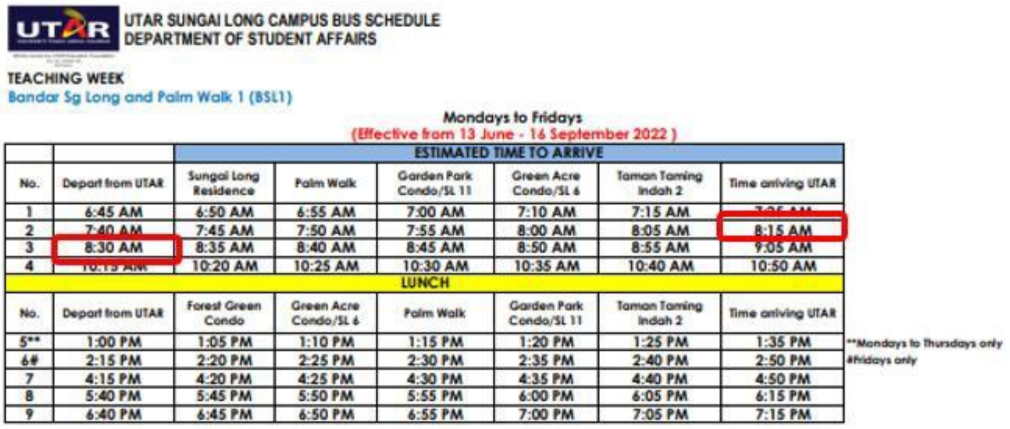


Figure 4.13: The Time Taken for Image Synchronised at Every Node.



UTAR SUNGAI LONG CAMPUS BUS SCHEDULE
DEPARTMENT OF STUDENT AFFAIRS

TEACHING WEEK
Bandar Sg Long and Palm Walk 1 (BSL1)

Mondays to Fridays
(Effective from 13 June - 16 September 2022)

		ESTIMATED TIME TO ARRIVE					
No.	Depart from UTAR	Sungai Long Residence	Palm Walk	Garden Park Condo/SL 11	Green Acre Condo/SL 6	Taman Taming Indah 2	Time arriving UTAR
1	6:45 AM	6:50 AM	6:55 AM	7:00 AM	7:10 AM	7:15 AM	7:25 AM
2	7:40 AM	7:45 AM	7:50 AM	7:55 AM	8:00 AM	8:05 AM	8:15 AM
3	8:30 AM	8:35 AM	8:40 AM	8:45 AM	8:50 AM	8:55 AM	9:05 AM
4	10:15 AM	10:20 AM	10:25 AM	10:30 AM	10:35 AM	10:40 AM	10:50 AM
LUNCH							
No.	Depart from UTAR	Forest Green Condo	Green Acre Condo/SL 6	Palm Walk	Garden Park Condo/SL 11	Taman Taming Indah 2	Time arriving UTAR
5**	1:00 PM	1:05 PM	1:10 PM	1:15 PM	1:20 PM	1:25 PM	1:35 PM
6#	2:15 PM	2:20 PM	2:25 PM	2:30 PM	2:35 PM	2:40 PM	2:50 PM
7	4:15 PM	4:20 PM	4:25 PM	4:30 PM	4:35 PM	4:40 PM	4:50 PM
8	5:40 PM	5:45 PM	5:50 PM	5:55 PM	6:00 PM	6:05 PM	6:15 PM
9	6:40 PM	6:45 PM	6:50 PM	6:55 PM	7:00 PM	7:05 PM	7:15 PM

**Mondays to Thursdays only
#Fridays only

Figure 4.14: The Bus Schedule.

4.7 Summary

From the study, the AI model is considered acceptable for flood water level forecasting. In addition, the NerveNet Wi-Fi mesh network is reliable. The TCP/UDP throughput, jitter, and latency within a triangular topology can fulfil most of the requirements of a flood forecasting system. If properly planned and configured, the NerveNet x86 Wi-Fi mesh devices can handle simple Internet services during natural disaster events. For NerveNet LoRa mesh, the LoRa message could be received in a few seconds if the MQTT payload size is small and there are no obstacles and interference between the subscriber node and publisher node. It is suitable to be implemented in a flood forecasting system since the flood warning message is in short plain text only. Furthermore, since the system only needs to transmit the small size flood warning text messages, it is sufficient for the system to synchronise the flood warning text message at a very fast speed so that the control centre can get the notifications from the gateway node and take action at the shortest time. Moreover, since the system only needs to transmit a small image after running a route, it is enough for the system to synchronise the images during the rest time of the buses at the UTAR bus stop.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, lightweight AI disaster detection has been successfully developed. The AI model used is LSTM, which has proven beneficial for the flood forecasting system. In addition, a NerveNet mesh network testbed with LoRa and Wi-Fi has been successfully designed and deployed. The design of the network meets the requirements set by the International Telecommunication Union. Besides that, the network's performance is also within an acceptable range of a resilient flood forecasting network. Lastly, the synchronisation of AI results among the node has also been successfully achieved. Since the system only needs to transmit the small size flood warning text messages and a small size image, it is enough for the system to synchronise the flood warning text message at a very fast speed so that the control centre can get the notifications from the gateway node and take action as soon as possible.

5.2 Recommendations for future work

Firstly, the AI model training and the testing dataset are obtained from Japan's organisation. Hence, the AI results may not apply to the local area since the weather, season, humidity, and geographical condition of Malaysia are different from Japan. The local dataset can be requested from the local government to build an AI model that can fit the situation in Malaysia's local area so that a better understanding of the feasibility of the AI model in disaster detection in Malaysia.

Secondly, the testbed is not fully deployed with powerful hardware due to budget constraints. There are only some of the nodes that use the powerful Wi-Fi adapter. As a solution, the project team can seek for a sponsor to support the budget that can improve the hardware throughout the whole network architecture. Hence, a better and more extensive network performance test can be conducted to have a better understanding of the feasibility of the network and system.

Besides, the system performance analysis may not be good and accurate enough because only three edge node devices operate on the dedicated bus routes and locations. Some blind spots that the floods frequently occur are not included in the testbed. As a solution, the number of the NerveNet nodes can be increased to have the AI flood forecasting operation in more locations. A larger testbed can be deployed to cover the whole Sungai Long area. Additional flood forecasting applications and features can be developed and tested with this testbed.

In addition, all the node devices in the project use the command-line operating systems, which are Linux and Raspberry Pi OS. The users need to memorise or refer to the documentation to operate the AI flood forecasting system. Hence, a graphical user interface (GUI) can be incorporated into the system because of its more modern appearance and ease of use. The GUI can be developed using some python packages such as the Tkinter package, PyQt5 package, wxPython package and so on. As a result, the system can become more user-friendly and efficient.

Last but not least, there is a problem encountered in the actual application of the system on the bus where the power supply cannot sustain the daily power consumption from the edge nodes in the buses. Each edge node in the bus is supplied by the power bank, which must be removed from the installation and charged with the power socket daily. This increases the labour fee to recruit people for this nuisance operation. Each edge node can be equipped with an off-grid solar power system as a solution. The solar panels installed on the buses' roofs convert the solar energy into DC power with the photovoltaic (PV) effect. After that, a solar inverter is applied to convert the DC power to AC power, which is then stored in the battery storage components called power banks. The AC power is fed into the edge node for its usage. Hence the AI flood forecasting system can maintain operation over a longer period.

REFERENCES

Adhikari, R. and Agrawal, R.K., n.d. An Introductory Study on Time Series Modeling and Forecasting.

ALFA NETWORK DISTRIBUTOR, 2010a. *Alfa APA M05 Accurate 7dBi Wifi Directional Antenna - Antenna Outdoor / Indoor*. [online] Available at: <https://www.alfa.net.my/products/Alfa-APA-M05-Accurate-7dBi-Wifi-Directional-Antenna/17> [Accessed 12 September 2022].

ALFA NETWORK DISTRIBUTOR, 2010b. *Alfa AWUS036ACH WiFi USB 3.0 AC Wifi Adapter Dual Band - Indoor Wireless*. [online] Available at: <https://www.alfa.net.my/products/Alfa-AWUS036ACH-WiFi-USB-3.0-AC-Wifi-Adapter-Dual-Band/66> [Accessed 12 September 2022].

ALFA NETWORK DISTRIBUTOR, 2010c. *Alfa Wifi Antenna ARS-N19 2.4GHz 9dBi Dipole Antenna - Indoor Wireless*. [online] Available at: <https://www.alfa.net.my/products/Alfa-Wifi-Antenna-ARS-N19-2.4GHz-9dBi-Dipole-Antenna/14> [Accessed 12 September 2022].

IBM, 2021. *What is distributed computing*. [online] Available at: <https://www.ibm.com/docs/en/txseries/8.1.0?topic=overview-what-is-distributed-computing> [Accessed 12 September 2022].

Aung, W.T. and Hla, K.H.M.S., 2009. Random forest classifier for multi-category classification of web pages. *2009 IEEE Asia-Pacific Services Computing Conference, APSCC 2009,* pp.372–376. https://doi.org/10.1109/APSCC.2009.5394100.

Chen, M., Li, W., Hao, Y., Qian, Y. and Humar, I., 2018. Edge cognitive computing based smart healthcare system. *Future Generation Computer Systems*, 86, pp.403–411. https://doi.org/10.1016/J.FUTURE.2018.03.054.

Choi, R., Lee, S.G. and Lee, S., 2020. Reliability Improvement of LoRa with ARQ and Relay Node. *Symmetry 2020, Vol. 12, Page 552*, [online] 12(4), p.552. https://doi.org/10.3390/SYM12040552.

Department of Irrigation and Drainage, 2022. *River Water Level Data – The Official Web of Public Infobanjir*. [online] Available at: <https://publicinfobanjir.water.gov.my/aras-air/data-paras-air/?state=WLH&lang=en> [Accessed 12 September 2022].

Dubey Abhishek, 2020. *A quick intro to Intel's OpenVINO toolkit for faster deep learning inference*. [online] Available at: <https://towardsdatascience.com/a-quick-intro-to-intels-openvino-toolkit-for-faster-deep-learning-inference-d695c022c1ce> [Accessed 12 September 2022].

Faruq, A., Arsa, H.P., Hussein, S.F.M., Razali, C.M.C., Marto, A. and Abdullah, S.S., 2020. Deep Learning-Based Forecast and Warning of Floods in Klang River, Malaysia. *Ingenierie des Systemes d'Information*, 25(3), pp.365–370. https://doi.org/10.18280/ISI.250311.

Gezer, V., Um, J. and Ruskowski, M., 2018. *An Introduction to Edge Computing and A Real-Time Capable Server Architecture*. [online] International Journalon Advances in Intelligent Systems. Available at: <https://www.researchgate.net/publication/326441179_An_Introduction_to_Edge_Computing_and_A_Real-Time_Capable_Server_Architecture> [Accessed 12 September 2022].

Glonaldots, 2015. *Latency vs Low Bandwidth - Impact on Web Performance*. [online] Available at: <https://www.globaldots.com/resources/blog/high-latency-vs-low-bandwidth-impact-on-web-performance/> [Accessed 12 September 2022].

HIVEMQ, 2015. *Quality of Service (QoS) 0,1, & 2 MQTT Essentials: Part 6*. [online] Available at: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/> [Accessed 12 September 2022].

HIVEMQ, 2022. *LoRaWAN and MQTT Integration for IoT Application Design*. [online] Available at: <https://www.hivemq.com/blog/lorawan-and-mqtt-integrations-for-iot-applications-design/> [Accessed 12 September 2022].

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation,* [online] 9(8), pp.1735–1780. https://doi.org/10.1162/NECO.1997.9.8.1735.

Junyoung, C., Gulcehre, C., KyungHyun, C. and Bengio, Y., 2014. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. [online] Available at: <https://www.researchgate.net/publication/269416998_Empirical_Evaluation_of_Gated_Recurrent_Neural_Networks_on_Sequence_Modeling> [Accessed 12 September 2022].

Khalifeh, A., Gholamhosseinian, A. and Hajibagher, N.Z., 2011. QOS For Multimedia Applications with Emphasize on Video Conferencing. [online] Available at: <http://www.diva-portal.org/smash/get/diva2:504299/FULLTEXT01.pdf> [Accessed 12 September 2022].

Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I. and Baba, D., 2019. Convolutional Neural Network Coupled with a Transfer-Learning Approach for Time-Series Flood Predictions. *Water 2020, Vol. 12, Page 96,* [online] 12(1), p.96. https://doi.org/10.3390/W12010096.

Kişi, Ö., 2011. A combined generalized regression neural network wavelet model for monthly streamflow prediction. *KSCE Journal of Civil Engineering 2011 15:8*, [online] 15(8), pp.1469–1479. https://doi.org/10.1007/S12205-011-1004-4.

Le, X.H., Ho, H.V. and Lee, G., 2020. Application of gated recurrent unit (Gru) network for forecasting river water levels affected by tides. *APAC 2019 - Proceedings of the 10th International Conference on Asian and Pacific Coasts*, [online] pp.673–680. https://doi.org/10.1007/978-981-15-0291-0_92/COVER.

Le, X.H., Ho, H.V., Lee, G. and Jung, S., 2019. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water 2019, Vol. 11, Page 1387*, [online] 11(7), p.1387. https://doi.org/10.3390/W11071387.

Lim, W.S., 2021. Disaster Resilient Mesh Network With Data Synchronization Using Nervenet.

Mahapatra, S., 2018. *Why Deep Learning over Traditional Machine Learning?* [online] Available at: <https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063> [Accessed 12 September 2022].

Mushtaq, M.S., Augustin, B. and Mellouk, A., 2012. Empirical study based on machine learning approach to assess the QoS/QoE correlation. *2012 17th European Conference on Network and Optical Communications, NOC 2012, 7th Conference on Optical Cabling and Infrastructure, OC and I 2012*. https://doi.org/10.1109/NOC.2012.6249939.

Rausch, T., Nastic, S. and Dustdar, S., 2018. EMMA: Distributed QoS-aware MQTT middleware for edge computing applications. *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*, pp.191–197. https://doi.org/10.1109/IC2E.2018.00043.

Samikwa, E., n.d. Flood Prediction System Using IoT and Artificial Neural Networks with Edge Computing. *DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING*.

Segal, M.R., 2004. *UCSF Recent Work Title Machine Learning Benchmarks and Random Forest Regression Publication Date Machine Learning Benchmarks and Random Forest Regression*.

Shchutskaya, V., 2018. *Latest Trends on Computer Vision Market – InData Labs Blog*. [online] Available at: <https://indatalabs.com/blog/trends-computer-vision-software-market> [Accessed 12 September 2022].

Suharjono, A., Apriantoro, R., Mukhlisin, M., Anif, M., Hidayat, S.S., Setiawan, T.A. and Kadiran, S.A., 2018. Network Design and Performance Evaluation of MQTT Based HetNet Using LoRa Network and IEEE 802.11 for Internet of Things. [online] Available at: <https://www.researchgate.net/publication/333915594_Network_Design_and_Performance_Evaluation_of_MQTT_Based_HetNet_Using_LoRa_Network_and_IEEE_80211_for_Internet_of_Things> [Accessed 12 September 2022].

The Things Network, 2022. *What are LoRa and LoRaWAN?* [online] Available at: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/> [Accessed 12 September 2022].

Yang, S., Yu, X. and Zhou, Y., 2020. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, [online] pp.98–101. https://doi.org/10.1109/IWECAI50956.2020.00027.

Yaseen, Z.M., El-shafie, A., Jaafar, O., Afan, H.A. and Sayl, K.N., 2015. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology*, 530, pp.829–844. https://doi.org/10.1016/J.JHYDROL.2015.10.038.