DEVELOPMENT OF LOW-COST LORA RSSI BASED DETECTION OF CATTLE RUSTLING: A CASE STUDY IN LAOS

DEVELOPMENT OF LOW-COST LORA RSSI BASED DETECTION OF CATTLE RUSTLING: A CASE STUDY IN LAOS

A Thesis Presented to

The Graduate School of Bangkok University

In Partial Fulfilment

of the Requirements for the Degree

Master of Engineering in Electrical and Computer Engineering

by

Thipchinda Prasayasith

2021

This thesis has been approved by

School of Engineering

Bangkok University

Title : Development of Low-Cost LoRa RSSI Based Detection of Cattle Rustling: A Case Study in Laos

Author : Thipchinda Prasayasith

Thesis Committee :

Thesis Advisor

(Assoc.Prof.Dr. Poompat Saengudomlert)

Thesis Co-advisor

(Dr. Waleed S. Mohammed)

Graduate Program Director

(Assoc.Prof.Dr. Chakkaphong Suthaputchakun)

External Representative

Taparugssanagoon A.

(Assoc.Prof.Dr. Attaphongse Taparugssanagorn)

(Asst.Prof.Dr. Wisarn Patchoo)

Dean of the School of Engineering

12 / 6 / 2023

Thipchinda, P., Master of Engineering in Electrical and Computer Science, December 2022, Graduate School, Bangkok University

Development of Low-Cost LoRa RSSI Based Detection of Cattle Rustling: A Case Study in Laos (54 pp.)

Advisor of Thesis**:** Poompat Saengudomlert, Ph.D.

ABSTRACT

This thesis investigates detection of cattle rustling by using low-cost Internet-of-Things (IoT) equipment. Recently, cattle localization based on LoRa received signal strength indicator (RSSI) values has gained a lot of attention since it requires low-power consumption and supports long-range communications. However, LoRa RSSI based localization is known to be inaccurate due to the frequency-changing nature of LoRa signals. Therefore, instead of trying to estimate an exact location, this paper focuses on detection whether an animal-attached sensor node is inside or outside the fence area. We propose the use of transmitter reference points (RPs) installed in selected locations to help determine the status of each animal. Experimental results indicate that, when received RSSIs from the same transmitter-receiver locations change over days, the use of RSSIs from transmitter RPs instead of RSSIs from a previous day as the train dataset allows the detection algorithm to adapt over time and provide higher detection accuracies in comparison to detection without using transmitter RPs. Finally, comparisons among well-known classification algorithms, including *k*-nearest neighbors (*k*-NN), support vector machine (SVM), decision tree, and random forest, are performed in terms of detection accuracies.

Keywords: Livestock monitoring, cattle tracking, IoT, LoRa, RSSI.

ACKNOWLEDGEMENT

First and foremost, I would like to earnestly acknowledge the sincere efforts and valuable time given by my supervisors, especially Dr. Poompat Saengudomlert for his invaluable patience, always giving feedback sessions, and paying attention to every single detail of all my work. Besides, I would also like to appreciate my co-advisor Dr. Waleed S. Mohammed for his helpful comments, persisting support and unflinching encouragement. This thesis work would not exist without their tremendous motivation and help.

I would like to thank my beloved family who unwaveringly support me and always cheer me up. I appreciate my father for kindly creating the solid base towers with cover protection for my LoRa receivers. Their unfailing faith and love for me has greatly encouraged me to keep working hard. Without that support I could not have succeeded in this project.

Last but not least, I would like to express my special thanks of gratitude to BU-CROCCS (Bangkok University Center of Research in Optoelectronics, Communication and Computational Systems) for location and resources in conducting this research.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

Page

CHAPTER 1

INTRODUCTION

1.1 Background

The Internet of things (IoT) refers to a system of computing devices that interconnect machines, objects or even living things such as people and animals. IoT enables the ability to transmit data over a network without requiring human-to-human or human-to-computer interactions. Smart farm is one of several applications that utilize IoT technology to improve agricultural productions. It is a concept of farming management using information and communication technologies (ICT) to increase the quantity and quality of products. It is well known that smart farms provide numerous benefits to agriculture sectors such as rice planting, vegetable and fruit growing. In fact, smart farms not only cover branches of vegetation planting, it is also related to animal husbandry. In the countryside, many people depend on agriculture and on animal husbandry to make a living and rise above poverty. Animal husbandry is one of the occupations that require a lot of time and efforts to make a profit since farmers have to take care of animals closely all the time. It is imperative that farmers do not lose their animals. Fortunately, modern ICT can help reduce the work load significantly. Integrating IoT technology with animal husbandry, referred to as livestock monitoring, is a topic that has gained a lot of attention.

Livestock monitoring involves observing animal behaviors related to movement, reproduction, nutrition, and the cattle health by applying modern technologies. From previous researches, two main aspects of livestock monitoring are health monitoring and location monitoring. Livestock health monitoring applications allow farmers to detect animal diseases in individual cattle early on in order to quickly evaluate responses to veterinary treatments (Abdullahi, et al., 2019) (Chaudhry, et al., 2020) (Luo, et al., 2020). Livestock location monitoring, also known as cattle tracking, allows farmers to secure and protect the livestock from thefts, natural disasters and organized crimes.

Over the last few years, progresses in low-power wireless network technologies have attracted a lot of new applications. Cattle tracking is one such application. Several studies have tried to solve the problem of localization with varying degrees of accuracy.

For cattle on large farms, it is imperative to utilize wireless technologies to specify their locations. Global Positioning System (GPS) has been deployed to track and monitor animals in order to conserve and protect the animals as well as their natural habitats (Panicker, et al., 2019) (Li, et al., 2018) (Zinas, et al., 2017) (Molina, et al., 2019). However, there are challenges in using GPS modules such as high battery consumption and high hardware costs, making the GPS approach not so attractive. Tracking-based IoT applications require low-power consumption and long-range communications. Recently, there are several low-power wide area network (LPWAN) technologies that use radio frequencies for data transmissions such as Sigfox, LoRa, etc. In particular, with transmission distances up to 15 km in rural areas, LoRa is suitable for monitoring cattle locations in large farm areas. Its low-power consumption is appropriate for IoT systems that work for several years on small batteries. LoRa can be applied to connect sensor devices, gateways, etc., wirelessly to the cloud. For example, the authors in (Ayele, et al., 2018) integrated

LoRa and Bluetooth system to monitor herds of wild animals by deploying ultra-low power IoT devices.

The main goal of this thesis is to create a cattle rustling detection system in which sensor nodes are attached to the cattle to determine their zones of locations (inside or outside the fence) on a farm in Vientiane, Laos. Towards this goal, LoRa received signal strength indicator (RSSI) values are used for classification between inside and outside zones. In general, an RSSI reflects the distance between a transmitter and a receiver but is sensitive to physical characteristics of the communication channel.

Two kinds of LoRa RSSI based localization have been studied previously, namely indoor localization (Ali, et al., 2021) (Anjum, et al., 2019) (Choi, et al., 2018) and outdoor localization (Panicker, 2019) (Li, 2018) (Zinas, 2017), (Choi, 2018) (Anjum, 2020) (Aernouts, 2018) (Lam, 2017) (Fargas, 2017) (Joshita, 2021) (Dieng, 2017) (Dieng, 2019) (Munoz, 2020) (Stojkoska, 2018). In this thesis focuses on outdoor localization since a large outdoor farm is considered. There are several studies regarding animal tracking using LoRa RSSIs combined with localization algorithms. In (Choi, et al., 2018), the authors present LoRa RSSI-based outdoor positioning to evaluate the accuracy of the fingerprinting approach. Moreover, they also compare their proposed LoRa RSSI-based fingerprinting to the GPS method to show the reduction of battery consumption. The use of machine learning techniques such as support vector machine and linear regression has been investigated for the fingerprinting approach in LoRa RSSI-based localization (Anjum, et al., 2020). In (Aenout, et al., 2018), the authors collected LoRa RSSI datasets from 68 LoRa Wide Area Network (LoRaWAN) base stations spread out in real city environments with buildings that can obstruct LoRa signals, and used LoRa RSSI-based fingerprinting with the *k*-nearest neighbor (*k*-NN) algorithm for outdoor localization. They found the average error to be as large as 400 m. The authors of (Lam, et al., 2017) tested LoRa RSSI based localization for outdoor environments, with the purpose of reducing noise in the system caused by obstacles, electronic interference, etc., by increasing from 6 to 14 receiver anchor points. It was demonstrated that the localization error is still high (more than 15 m) with 14 receiver anchor points.

In the above-mentioned papers related to LoRa RSSI-based localization using the fingerprinting approach, experimental results indicate that LoRa RSSIs provide limited accuracy, especially when the number of receivers is small. Therefore, this research aims to classify the animal location (inside or outside the fence area) instead of estimating the exact location in order to avoid high installation costs from a lot of receiver anchor points. In addition, this thesis proposes an approach of classification by using transmitter reference points (RPs) that are mounted inside and outside the fence area on a farm. An RP is an immobile transmitter that operates in the same way as the animal-attached transmitter sensor node. Since RSSIs for localization on different days may vary from the changes in environments on a daily basis, the use of RPs provides reference RSSI data that can adapt over time, making the classification more accurate compared to using reference/train data from the previous day.

1.2 Objective

- Study applications of IoT technologies for animal husbandry.

- Develop a WSN with LoRa technologies and to detect buffaloes leaving the fence area based on LoRa RSSI values.

- Evaluate the performance of the developed system in an authentic farm.

1.3 Scope and Limitation

- A sensor node is used to represent a buffalo. No actual buffalo is used in the experiments.

- LoRa technologies are used for wireless communications.

- The research focuses on detection of buffaloes leaving the fence area, but not on estimating exact locations of buffaloes.

1.4 Thesis Outline

This thesis report is organized as follows. Chapter 2 provides a literature review on relevant technologies and related work. Chapter 3 proposes methodology without transmitter RPs. Chapter 4 shows the result of experiments without transmitter RPs. Chapter 5 provides the proposed approach by using transmitter RPs. Finally, Chapter 6 concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

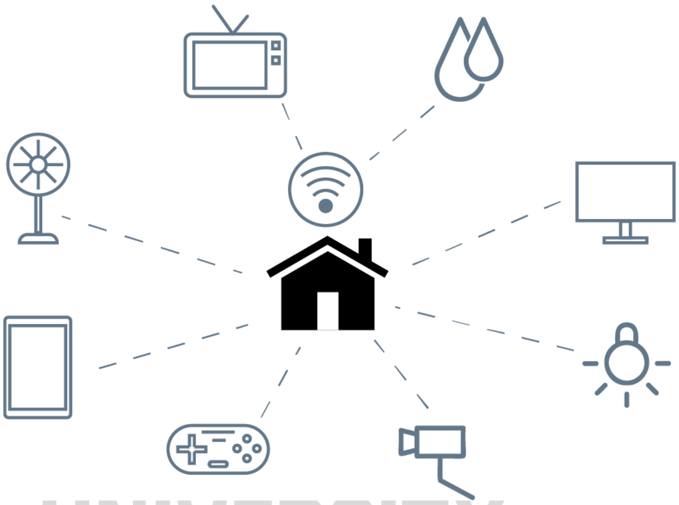
This chapter is organized as follows. Section 2.1 presents background of IoT and its IoT applications. Section 2.2 describes smart farm and livestock monitoring. Section 2.3 present localization technique. Section 2.4 introduces s LoRa/LoRaWAN technology. Section 2.5 presents existing work involving localization based on LoRa RSSIs for indoor and outdoor environments. Finally, Section 2.6 introduce classification techniques used in this thesis.

2.1 IoT

IoT is a network of electronic devices that are connected to the Internet, allowing us to either control devices or receive data from them using our smartphones or computers from everywhere around the world. IoT provides a platform that allows people to connect electronic devices and control them, possibly with big data technology. It helps increase efficiency in work performance, bring economic benefits, and minimize the need for human involvement. In addition, IoT will change our daily life to be more comfortable and also reduce our fatigue. Most IoT projects are using wireless sensor networks (WSNs), which refer to a group of sensor nodes that exchange data in a wireless network. As a WSN is easy to install and flexible in the field for system integrators, it becomes an essential part of IoT solutions.

An IoT system could be a single gadget, for instance a wearable health tracking device, or as complex as a smart city with sensors deployed across the entire city. One of the most outstanding examples of IoT projects is the smart home as shown in figure 2.1. It allows us to control every connected device in our home, from a device used to water the garden outside to all electrical equipment inside the house or even a thermostat in the bathroom. In general, any object that can be connected to the internet will be a candidate for an IoT device.

Figure 2.1: IoT for smart home



IoT is regarded as a significant technology that can improve almost all activities in our lives to become modern. Most of the devices, which have not previously been connected to the Internet, can be 8 networked and operate like smart devices. IoT provides many benefits to several sectors such as agriculture, healthcare, automobile, business or even industrial organizations. Most modern enterprises are already taking advantage of IoT to automate and simplify many of their daily tasks. According to (Mallon, n.d.), the advantages of IoT technology are as follows:

- IoT promotes efficient resource utilization.

- IoT helps reduce human labor in many sectors.

- Applying IoT will lower the cost of production and increase the productivity.
- IoT makes analytics decisions faster and more accurate.
- IoT promotes real-time marketing of products.
- IoT enhances new client experiences.
- IoT guarantees high-quality data and secure processing

2.2 Smart Farm

Smart farm is one of several applications that utilize IoT technology to improve agricultural productions. It is a concept of farming management using information and communication technologies (ICT) to increase the quantity and quality of products. It is well known that smart farms provide numerous benefits to agriculture sectors such as rice planting, vegetable and fruit growing. In figure 2.2, the integrated technology with the agriculture sector allows the farmer to see the real-time status of the farm. In addition, other devices on the farm can also collect data and make intelligent decisions to assist the farmer.

Figure 2.2: Smart farm

Source: SPsmartplants. (2022). *Smart Green House.*

Retrieved from http//: www.spsmartplants.com

The general issues that smart farm targets to solve include aspects such as how much fertilizer to apply, time of application, the specific areas to be applied, and which resources are needed for plant protection. With smart farm, farmers find it easy to measure variables and process data with precision. This enables tasks to be much simpler, providing improvement of yields, cutting costs, moving towards sustainable agriculture, and also increasing the quality of production. Applying the IoT technology to the farm could help crop treatment such as accurate planting, watering, pesticide application and harvesting, which directly affect production rates. Weather predictions and soil moisture sensors allow for water use only when and where needed. Thus, smart farm will save the farmer money and labor compared to traditional farm.

In fact, smart farm not only covers branches of vegetation planting, it is also related to animal husbandry. Animal husbandry is one of the occupations that require a lot of time and efforts to make a profit since farmers have to take care of animals closely all the time. It is imperative that farmers do not lose their animals. Fortunately, modern ICT can help reduce the work load significantly. Integrating IoT technology with animal husbandry, referred to as livestock monitoring, is a topic that has gained a lot of attention.

Livestock monitoring involves observing animal behaviors related to movement, reproduction, nutrition, and the cattle health by applying modern technologies as shown in figure 2.3. From previous researches, two main aspects of livestock monitoring are health monitoring and location monitoring. Livestock health monitoring applications allow farmers to detect animal diseases in individual cattle early on in order to quickly evaluate responses to veterinary treatments (Abdullahi,2019) (Chaudhry, 2020) (Luo, 2020). Livestock location monitoring, also known as cattle tracking, allows farmers to secure and protect the livestock from thefts, natural disasters and organized crimes.

Figure 2.3: Smart necklace

Source: FarmingUK Team. (2015). *High tech 'cow bell' brings revolution to cattle movement.*

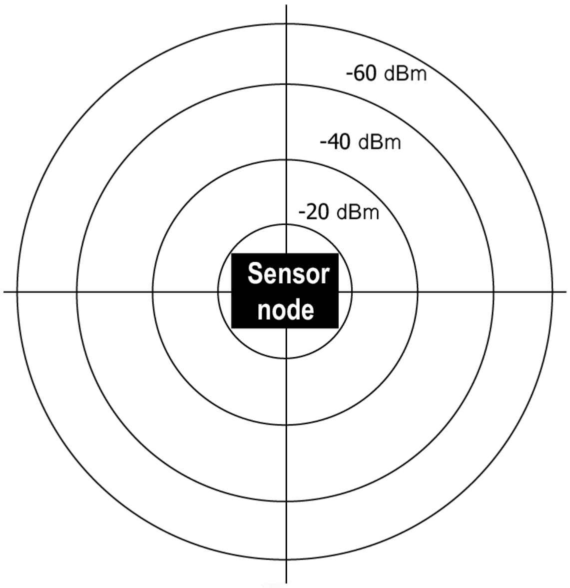
Retrieved from http//: www.farminguk.com

Over the last few years, progresses in low-power wireless network technologies have attracted a lot of new applications. Cattle tracking is one such application. Several studies have tried to solve the problem of localization with varying degrees of accuracy. For cattle on large farms, it is imperative to utilize wireless technologies to specify their locations. Global Positioning System (GPS) has been deployed to track and monitor animals in order to conserve and protect the animals as well as their natural habitats (Panicker, 2019) (Li, 2018) (Zinas, 2017) (Molina, 2019). Nevertheless, using GPS modules is a solution with some limitations such as high battery consumption and high hardware costs, making the GPS approach not so attractive.

2.3 Localization

Localization is the process of finding the position of a sensor node in a wireless sensor network (WSN). The easiest way to find the exact location is attaching a GPS module to the node. However, there are challenges when a large number of nodes exist since GPS modules are expensive and consume a lot of power. In addition, GPS cannot be used indoor. Therefore, several methods have been proposed to solve the issue of localization instead of using GPS such as triangulation, trilateration, time of arrival (TOA), and received signal strength indicator (RSSI) (Kulaib, 2011). This paper applies the RSSI technique, which is commonly used for sensor node localization. In this research, the system location always has data transmissions between transmitters and receivers; to finding a node location is mainly based on the distances between the anchor nodes and node of interest (with unknown location). The RSSI is used to measure the received signal strength to estimated distance between the transmitter and the receiver. As the distance between the transmitter and the receiver increases, the signal strength decreases as shown in figure 2.4. The RSSI values are measured in dBm and typically have negative values ranging between 0 dBm (excellent signal quality) and -100 dBm (extremely poor signal quality).

Figure 2.4: Examples of the RSSI range, with the RSSI value decreasing with the transmission distance



2.4 LoRa and LoRaWAN

LoRa is a combination of two terms: Long and Range. It is a wireless radio frequency technology with low power consumption and provides secure data transmissions for M2M and IoT applications. LoRa is based on chirp spread spectrum (CSS) modulation, which has low power characteristics like frequency-shift keying (FSK) modulation but can be used for longer ranges with the transmission distance up to 15 km in rural areas. LoRa can be applied to connect devices, sensors, gateways, machines, animals, people, etc., wirelessly to the cloud.

LoRa technologies use license-free sub-gigahertz radio frequency bands in different regions. In the USA, it operates in the 915 MHz band. In Europe, it operates in the 868 MHz band. In Asia, it operates in the 433-to-434 MHz, 865-to-867 MHz, and 920-to-923 MHz bands. In theorical, the use of low bandwidth 400 MHz bands has much lower power limits than 900 MHz in the ISM Bands. 433 MHz uses for keyless entry and other low data rate or cost sensitive application. Due to this thesis based on low-cost equipment, the use of 433 MHz is more appropriate. Table 2.1 summarizes key specifications of LoRa.

Table 2.1: LoRa technology specifications (EverythingRF, 2018)

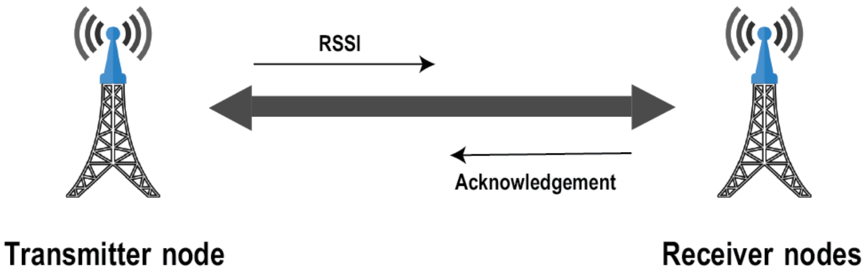
Governing body	**LoRa Alliance**
Frequency	ISM 433/868/915 MHz
Range	Up to 5 km (Urban) and 15 km (Rural)
Data rate	0.3 - 27 kbps
Modulation	CSS modulation based on FM technology
Standard	801.15.4g
Error detection	32-bit CRC

LoRa Single channel gateway

Single channel gateway is a LoRa device that operates as a gateway by transmitting LoRa packets to the network. It can only receive on one channel and one spreading factor at the same time, while multi-channel gateway can receive up to 8 channels and 6 spreading factors. A single channel is not much expensive compared with multi-channel gateway.

This thesis aims to use offline gateway which is peer-to-peer communication in single channel mode by using 5 LoRa receiver gateway with 1 transmitter for RSSI data transmission. The sensor node which is the transmitter can be configured to periodically transmit the RSSI data in dBm. The receiver node can listen to the incoming data and store data to a memory card. The receiver node can also send an acknowledgment message back to the transmitter node when the message is received.

Figure 2.5: Peer-to-peer communication between transmitter and receiver node



2.4.1 LoRa Parameters

The LoRa specification in Table 2.1 is the standard which is set from manufacturers to use in general situations for both indoor and outdoor areas. Adjustment of LoRa parameters to correspond with user scenario is possible. More specifically, it is possible to adjust the bandwidth, the spreading factor, and the coding rate in LoRa transceiver modules to influence the range, the data rate and penetration of LoRa signals. These three parameters will impact the operations of transceivers. Each parameter can be explained as follows (Markqvist, 2020):

Bandwidth

Bandwidth (BW) is the frequency range of the chirp signal used to carry the baseband data. LoRa can be configured to use bandwidths in pre-determined steps from 7800 Hz to 500,000 Hz in the sub-GHz bands, and from 250 kHz to 1.6 MHz in the 2.4 GHz band. The default bandwidth of LoRa RFM96W module is 125000 HZ. Selecting a narrower bandwidth will result in a slower transfer rate, but will increase the transmission range. On the other hand, selecting a wider bandwidth will result in an increased data rate, but will decrease the transmission range.

Spreading Factor

In LoRa, the value of Spreading Factor (SF) is the number of bits per data symbol, and its power of 2 is the number of chips used to represent each symbol. A symbol refers to one or more bits of data, and can be a kind of waveform or a code. The symbol rate is the number of symbols transferred per time unit. It could be equal to or less than the bit rate. The following formula is the relationship between the symbol rate, BW, and SF.

$$Rs = \frac{\mathrm{BW}}{2^{SF}}$$

Rs = Symbol Rate

SF = Spreading Factor

BW = Bandwidth

If a higher SF is selected, each payload data symbol will be spread out over more chips, which means there will be more processing gain at the receiver side. LoRa can be configured for SFs between 7 and 12. The default SF of LoRa RFM96W module is 7. A higher SF also increases the time on air, which increases energy consumption, reduces the data rate, but improves the communication range. For successful packet transmissions, the choice of the SF and the modulation method must be consistent between a transmitter and a receiver.

Coding Rate

In LoRa modulation, forward error correction is applied in every data packet transmission. This operation is done by encoding each group of 4-bit data into a 5-bit, 6-bit, 7-bit, or 8-bit codeword. The corresponding code rates are usually specified as fractional numbers equal to 4/5, 4/6, 4/7, or 4/8. The default code rates of LoRa RFM96W module is 4/5. Using extra bits allows LoRa signals to tolerate short interferences and become more reliable. The Coding Rate (CR) can be adjusted according to channel conditions for data transmissions. When channel conditions become poorer, it is recommended to increase the CR. However, increasing the CR will also increase the duration for data transmissions (KWKWII, 2018).

The following formula can be used to calculate the bit rate (R_b) in terms of the BW, the CR, and the SF.

$$R_b = SF \text{ x } \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} \text{ x } 1000$$

SF = Spreading Factor (7, 8, 9, 10, 11, 12)

CR = Coding Rate (1, 2, 3, 4)

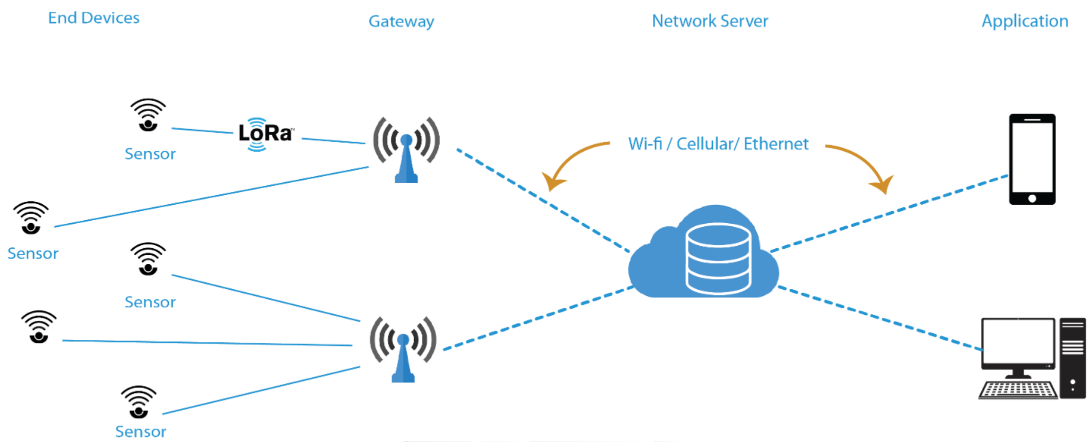
BW = Bandwidth in kHz (7.8, 10.4, 15.6, 20.8, 31.25, 41.7, 62.5, 125, 250, 500)

2.4.2 LoRaWAN

LoRaWAN refers to a low-power wide-area network protocol developed by LoRa Alliance and is freely available. It is a point-to-multipoint networking protocol that uses the LoRa modulation scheme. LoRaWAN offers long-range bi-directional communication between sensors and base stations at distances up to 15 km with very low power consumption, allowing operations for up to ten years without having to replace the batteries.

The LoRaWAN network architecture consists of 4 parts, which are end devices, gateways, servers and applications as shown in figure 2.5. The network is typically based on a star-of-stars topology, where the gateways perform as a bridge that forwards data between a central network server terminal and end-device sensors. The end devices use a wireless single-hop data transfer to one or many gateways, which are connected to network servers via standard IP connections. The applications act as displays to monitor as well as analyze output data from end-devices, for instance sensor nodes, etc.

Fig 2.6: LoRaWAN architecture



- An end-device such as a sensor node is an object with an embedded low-power communication device.
- A gateway is the antenna unit that receives broadcasts from end devices and sends data back to end devices.
- A network server is a server that routes messages from end devices to the relevant application, and back.
- An application is a piece of software running on a server (The Things Network, n.d.).

2.5 LoRa RSSI Based Localization

Recently, the LoRa technology has received lots of attention. Several studies use the LoRa technology to implement indoor and outdoor positioning system. Due to the variation of RSSI values over time, it is a challenge to maintain accurate. In general, indoor environments are more complex than outdoor because there are several factors that can weaken signals, including obstacles, noise from electronic devices, and human movement. Therefore, the authors of (Ali, 2019) proposed LoRa technology integrated with deep learning methods as a solution for the use of fingerprinting in indoor localization. Based on good penetration capability of LoRa signals, their use with the deep learning approach helps solve the issue of changing environmental conditions. They investigated how accurate the model produced by the training process in estimating the location in a dynamic environment.

In outdoor localization, with the LoRa transmission distance up to 15 km in rural areas, several researchers take this advantage to apply LoRa to IoT application such as livestock monitoring. In (Dieng, 2019), the authors proposed an RSSI-based distance estimation scheme for localization of cattle communicating with LoRa. The proposed solution is to decrease the cost of cattle localization with GPS and allow accurate localization of cattle without GPS by using a LoRa RSSI technique. Although RSSI is a common technique that many IoT application apply, there is a challenge in accuracy. The use of RSSI has advantages and disadvantages as shown in Table 2.2.

Table 2.2: Advantage and disadvantage of LoRa

Advantage	Disadvantage
- It is cost-efficient.	- It is extremely sensitive to spectrum interference including noise and multipath fading.
- It is easy to implement.	- It can require fingerprinting.
- It is compatible with the majority of the technologies.	- Lower accuracy
- Low hardware requirements	

Several research in terms of LoRa RSSI based outdoor localization have been studied earlier, including (Panicker, 2019) (Li, 2018) (Zinas, 2017), (Choi, 2018) (Anjum, 2020) (Aernouts, 2018) (Lam, 2017) (Fargas, 2017) (Joshita, 2021) (Dieng, 2017) (Dieng, 2019) (Munoz, 2020) (Stojkoska, 2018).

2.6 Classification Algorithms

Classification is a technique where we categorize data into a given number of classes. The main goal of classification is to determine the category or class of new observations on the basis of training data. There are a lot of classification technique. Applying each technique depends on the category of data collected. Choosing the technique wisely will lead to an efficient result. The common techniques that have been used widely in machine learning and will be adopted in this thesis are as follows:

- *k* -nearest neighbors
- Support vector machine
- Decision tree classification
- Random forest classification

2.6.1 *k*-Nearest Neighbors

k-nearest neighbors (*k*-NN) is a data classification method that uses a set of datapoints and their classes to predict the class value for a new datapoint by considering its *k* nearest datapoints. It stores all the available data and classifies a new data point based on the similarity. When a new datapoint appears, it can be easily classified into one category by comparing it to the training data.

Pros:

- *k*-NN is simple and is the easiest to perform.
- There is no require to build a model, tuning many parameters, or make additional assumptions like some of the other classification algorithms.
- It is flexible and can be used for classification, regression, and search.

Cons: The algorithm will get significantly slower as the size of data increases.

2.6.2 Support Vector Machine

Support vector machine (SVM) is one of the most popular machine learning algorithms, which is used for classification as well as regression problems. The goal of the SVM algorithm is to create the best line or plane that can divide a multi-dimensional space into classes so that we can easily put the new datapoint in the correct category. SVM can be of two types (Javatpoint.com)

- Linear SVM: Linear SVM is used for linearly separable data. If a data set can be classified into two classes by using a single straight line or plane, then such a data set is termed as linearly separable, and the classifier is called a Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used when data are not linearly separable. If a data set cannot be classified by using a line or plane, then such a data set is not linearly separable, and the classifier is called a non-linear SVM classifier.

Pros:

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is more effective in high-dimensional spaces.

Cons: The algorithm is mathematically complex and computationally expensive.

2.6.3 Decision Tree

Decision tree is a non-parametric machine learning method that used a tree-like model for solving both classification and regression problems The goal is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from training data.

Pros:

- Simple to understand and visualize.
- Can handle both numerical and categorical data.
- When compared to other algorithms, decision trees require less effort for data preparation and pre-processing.

Cons: A few changes in the data can affect a large change in the structure of the decision tree causing instability.

2.6.4 Random Forest

Random forest algorithm is a classifier that consists of many decision trees. Instead of relying on one decision tree, it takes the prediction from each tree. Based on the majority vote of predictions, the algorithm predicts the final result. The large number of trees reduces the overfitting of data sets and increases precision. This algorithm can handle a data set containing continuous variables, which is the weakness of the decision tree algorithm.

Pros:

- It can handle large data sets efficiently.
- Can perform both classification and regression tasks.
- It provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

Cons: Not appropriate for real-time prediction, difficult to implement, and highly complex when compared to decision trees.

CHAPTER 3

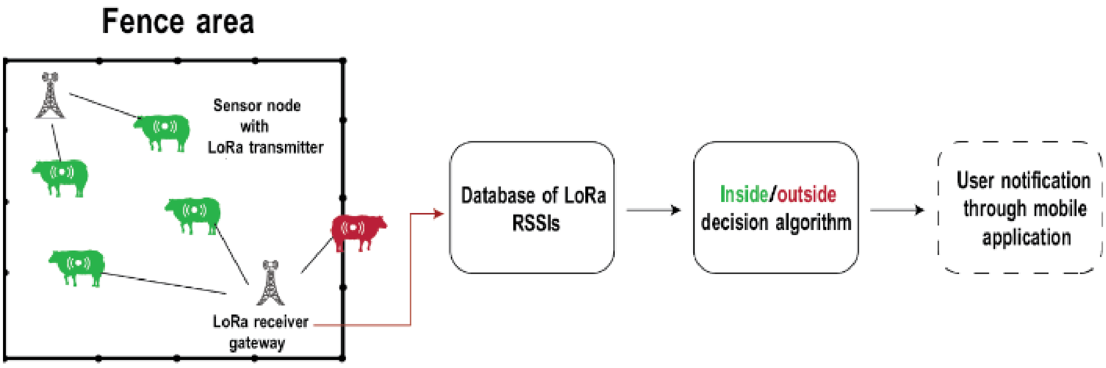
METHODOLOGY WITHOUT TRANSMITTER REFERENCE POINTS

This chapter presents the methodology used in this thesis to create a cattle rustling detection system in which sensor nodes with LoRa modules are attached to the cattle to detect whether each animal is inside or outside the fence area.

3.1 System Implementation

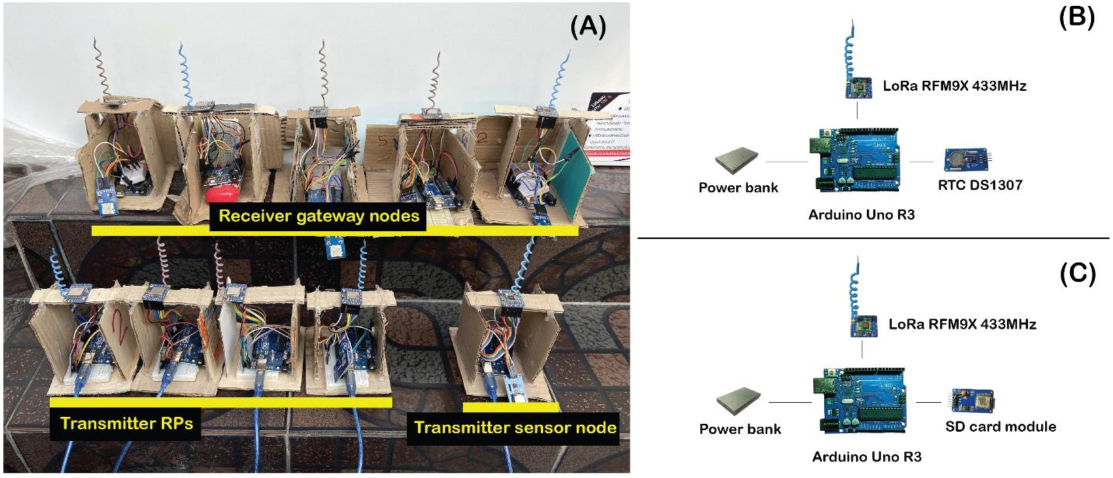
The developed system can be divided into four main parts: animal-attached sensor nodes, LoRa gateways, an RSSI database, and a detection algorithm. The sensor nodes are attached to the cattle to transmit node numbers as well as timestamps to the gateways, which collect RSSI values and their corresponding transmitter nodes in a database. LoRa gateways are installed only in the fence area. The goal is to create a detection system that operates to monitor animals and will notify the farmer when an animal exits the fence area as shown in figure 3.1. However, this thesis covers up to the detection algorithm, with mobile notification left as future work. In addition, LoRa RSSIs collected at LoRa gateways are stored in SD cards and manually collected to form train and test datasets in this work, with automatic transmissions of these RSSIs to a database left as future work.

Figure 3.1: Components of cattle rustling detection system based on LoRa RSSIs.



Low-cost sensor nodes, gateway receivers and transmitter RPs have been created. Each sensor node is a wearable device attached to a buffalo, and consists of an Arduino Uno microcontroller, a 433-MHz RFM96W LoRa module, a real time clock DS1307 module, and a power bank. A sensor node and a transmitter RP have the same hardware as shown in figure 3.2. The use of RPs will be investigated in the chapter 5. A receiver gateway consists of a LoRa module, a power bank, and an SD card module to store RSSIs, transmitter node numbers, and timestamps as datasets. All the modules are connected using jumper wires and breadboards. In addition, the LEDs have been added as indicators that are turned on when data messages are sent or received.

Figure 3.2: (A) Test equipment with four transmitter sensor nodes and five receiver gateway nodes based on LoRa hardware, (B) Components of a transmitter node, (C) Components of a receiver node.



3.2 Test Area

The fence area for experiments has an area of approximately 25,500 m^2. The area is approximately a rectangle surrounded by a fence as shown in figure 3.3, and is on a large outdoor farm with few trees and a swamp located in the suburb of Vientiane, Laos. There is a herd of buffaloes discharged daily to graze on the farm from morning until evening on each day.

Figure 3.3: Fence area for experiments, with latitude and longitude coordinates of the four corners.

According to the thesis objectives mentioned in chapter 1, the goal is to find whether the location of each buffalo-attached transmitter sensor node is inside or outside the fence while buffaloes are discharged to graze. For practical use, the developed sensor node can be attached to a buffalo as a smart necklace to detect its current location.

For experiments, this work focuses on the right side of the fence area as shown in figure 3.3. The test area contains six zones. Three green zones are inside and three red zones are outside the fence. The sensor node for a smart necklace is a moving transmitter that communicates with a set of fixed receivers. There are five receivers mounted in different areas inside the fence as shown in figure 3.3. Both of receivers and transmitter are set to 1.5 m height from the ground. Each receiver, as shown in figure 3.4, functions as a gateway to store RSSI values, transmitter node numbers and timestamps, to be used for localization by the *k*-NN algorithm, which is a data classification method that uses a set of datapoints and their classes to predict the class value for a new datapoint by considering its *k* nearest datapoints. The receiver locations are chosen to be inside the fence only for ease of maintenance and future setup of wireless connectivity.

Figure 3.4: Receiver gateway node mounted on a solid base with cover protection.

CHAPTER 4

RESULTS WITHOUT TRANSMITTER REFERENCE POINTS

4.1 Experimental Results without Transmitter RPs

In each experiment, the transmitter sensor node was viewed as a buffalo grazing on the farm with no obstacle and referred to simply as the buffalo. The buffalo will send data messages, which contain transmitter numbers and timestamps to each gateway receiver inside the fence area, and the transmitted data together with RSSIs are saved in the SD card of each receiver. Data are collected by taking a random walk from zone 1, zone 2, and so on up to zone 6. Each zone is approximately a square area whose size is 25 m × 25 m. Data are gathered in each zone for 15 minutes. Then, the same process is repeated in the next zone until all six zones are covered. The first experiment was performed in the afternoon of 11 November 2021, with example raw data in Table 4.1. A total of 545 datapoints is obtained from the buffalo in one and a half hour, where 280 datapoints are from the inside and 265 datapoints are from the outside. Each of these datapoints contains RSSIs from all five receivers. Incomplete datapoints with some RSSIs missing are not taken into consideration. The entire dataset is randomly split into a train dataset and a test dataset, which contain 70% and 30% of the datapoints, respectively. Afterwards, we applied Python programming with the scikit-learn library (Geron, 2019) to analyze the data using the *k*-NN algorithm to decide, for each test datapoint, whether the buffalo is inside or outside the fence area at that time.

The results are presented in terms of the precision, recall, and accuracy defined as follows. A positive detection result refers to a buffalo being outside while a negative result refers to a buffalo being inside. Let TP and FP denote the percentages of true positives and false positives, respectively. Let TN and FN denote the percentages of true negatives and false negatives, respectively. Then,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{2}$$

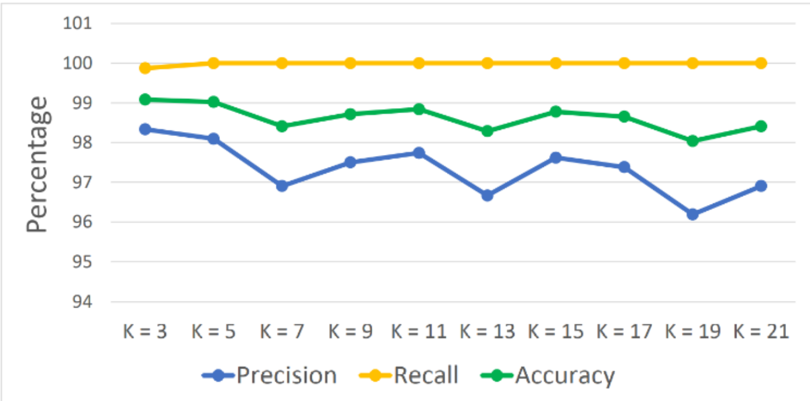
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\%. \tag{3}$$

Using k = 11 for the k-NN algorithm and taking the averages from 10 runs, the values of precision, recall, accuracy are 97.87%, 100%, and 98.9%, respectively. Since the performance of the k-NN algorithm depends on the value of k, we vary k from 3 to 21 to find its appropriate value. From the numerical results in figure 4.1, the value of k does not affect the accuracy significantly. Therefore, the middle value of k = 11 is selected for all upcoming experiments.

Table 4.1: Example raw data received on 11/11/2021.

Transmitter node number	Timestamp	RSSI (dBm)	Receiver node number
RP1	11/11/2021, 10:42:7	-41	2
RP2	11/11/2021, 10:43:9	-58	3
Buffalo	11/11/2021, 10:42:24	-45	4
RP3	11/11/2021, 10:42:15	-40	5
RP4	11/11/2021, 10:43:58	-69	1

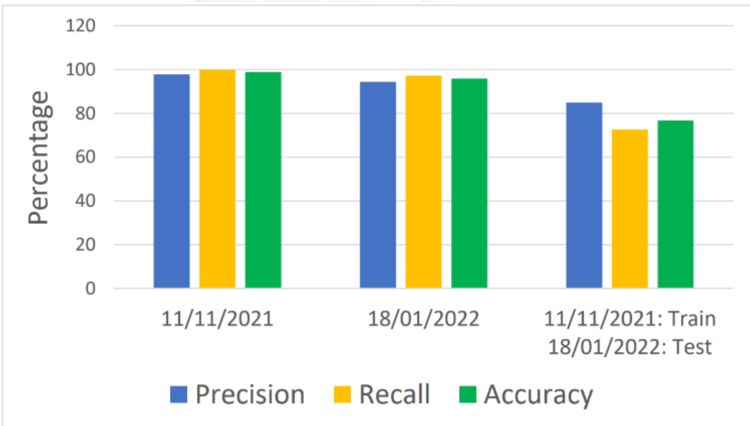
Figure 4.1: Precisions, recalls, and accuracies for the 11/11/2021 dataset for different values of *k* for the *k*-NN algorithm.



Another experiment was performed on 18 January 2022 using the same method at the same location. This time, the obtained dataset contains a total of 476 datapoints, including 231 datapoints from inside and 245 datapoints from outside. Based on the new dataset, the accuracy is 95.86%.

Based on the datasets on two different days, the detection accuracy is quite satisfactory (more than 95%). However, it may not be practical to get a train dataset every day before using the system. Having the datasets from two days, the former is used for training and the latter is used for testing. In particular, from the 11/11/2021 train dataset and the 18/1/2022 test dataset, the detection accuracy decreases to 76.69%, as shown in figure 4.2.

Figure 4.2: Precisions, recalls, and accuracies for the 11/11/2021 and 18/01/2022 datasets with (1) train and test datasets on the same day with 70% of datapoints for training (2) train and test datasets on different days.



From the results in figure 4.2, it can be observed that the system cannot to detect accurately when train and test datasets are obtained on different days. This is because RSSIs from the same transmitter and receiver locations can change over time. Hence, in the next chapter, the use of transmitter RPs is proposed and investigated as a solution to perform data classification without a train dataset from a previous day. The proposed approach is appropriate in scenarios whose environments change over time, causing significant changes in RSSIs that are used for detection.

CHAPTER 5

PERFORMANCE IMPROVEMENT USING TRANSMITTER REFERENCE POINTS

5.1 Methodology with Transmitter RPs

Transmitter RPs are fixed transmitters mounted inside and outside the fence area. They work in the same way as the buffalo transmitter. Each transmitter RP repeatedly sends its node number and timestamps to the gateway receivers. We presume that, when the buffalo comes close to a transmitter RP, the RSSIs from both transmitters would be close. Thus, it is possible to indicate whether the buffalo is inside or outside by comparing its RSSIs to those from all the RPs. Both sides of the fence (inside and outside) contain two RPs, which are mounted in the centers of zones 1, 2, 5, and 6, as shown in figure 5.1.

Figure 5.1: Deployment of transmitter RPs both inside and outside the fence area.

After four transmitter RPs has been made, experiments on two different days that are two weeks apart were performed. In particular, experiments were performed on 15 and 29 March 2022. On each day, the buffalo moved with a random walk from zone 1, zone 2, and so on up to zone 6. During the first three minutes in each zone that has a transmitter RP, the buffalo transmitter has been put close to the RP to calibrate both transmitters to have similar RSSIs before taking a random walk around the zone. The four transmitter RPs as well as the buffalo transmitter sent data messages to all receivers repeatedly. Waiting times between successive message transmissions are randomly selected from 3 to 5 second. The sizes of datasets received from both days are shown in Table 5.1. As before, incomplete datapoints with some RSSIs missing are not taken into consideration.

Table 5.1: Number of datapoints from different transmitter nodes.

	Buffalo		RPs	
	Inside	Outside	Inside	Outside
15/03/2022 dataset	144	146	1119	995
29/03/2022 dataset	100	124	741	741

In Table 5.1, the number of datapoints from the transmitter RPs is much higher than that from the buffalo transmitter because there are four RPs but only one buffalo transmitter.

As specific example datapoints, figure 5.2 shows the first 200 datapoints from transmitter RP 1 on 15 March 2022. Each datapoint contains five RSSIs collected by all five receivers denoted by RX1 to RX5. Observe that the values are quite stable in the range of -70 to -40 dBm. The RSSIs are consistent with the fact that transmitter RP 1 is closer to RX1 and RX2 than to RX4 and RX5 while RX3 is in the middle.

Figure 5.2: The first 200 datapoints from transmitter RP 1 to all five receivers (denoted by RX1 to RX5) on 15 March 2022.

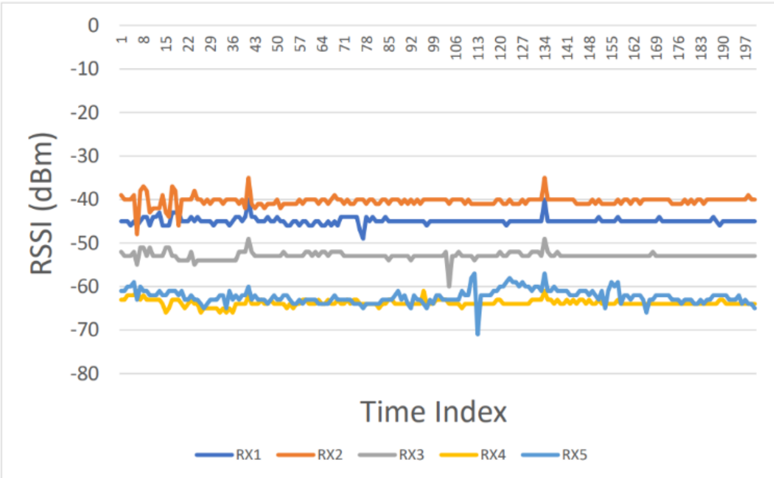
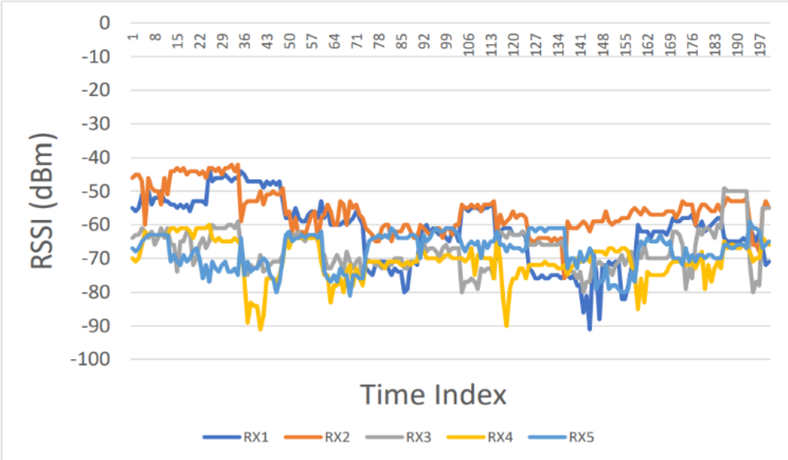


Figure 5.3 shows the first 200 datapoints from the buffalo transmitter on 15 March 2022. Compared to the datapoints from transmitter RP 1 in figure 5.2, the RSSIs vary significantly in the range of -90 to -40 dBm because the buffalo transmitter moved according to a random walk in each zone and moved across different zones.

Figure 5.3: The first 200 datapoints from the buffalo transmitter to all five receivers (denoted by RX1 to RX5) on 15 March 2022.



5.2 Results with Transmitter RPs

Using the datasets from both days as mentioned in Table 5.1, Python programming was applied to perform inside/outside detection and compute precisions, recalls, and accuracies using Eq. (1) to Eq. (3). First, without using datapoints from the transmitter RPs, the datapoints from the buffalo transmitter on 15 March 2022 were used as a train dataset while the datapoints from the buffalo transmitter on 29 March 2022 were used as a test dataset. Then, on each day, the datapoints from the transmitter RPs were used as a train dataset while the datapoints from the buffalo transmitter were used as a test dataset.

Figure 5.4: Precisions, recalls, and accuracies for the 15/03/2022 and 29/03/2022 datasets with (1) train and test datasets on different days (2) train and test datasets on the same day with datapoints from RPs for training.

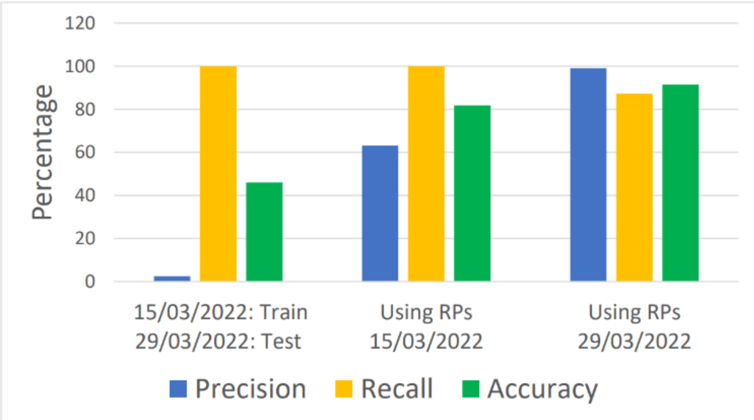
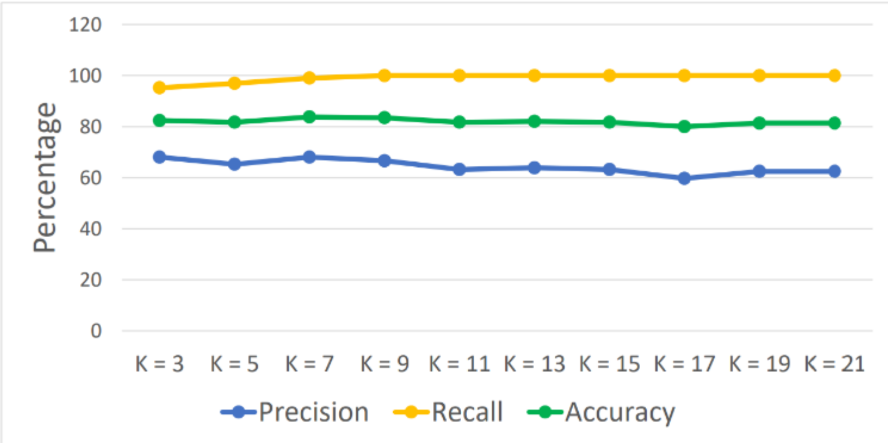


Figure 5.4 shows the accuracies for different cases of train and test datasets. The results indicate that using a train dataset from a previous day leads to a low accuracy of 45.99% even though the recall is close to 100%. The low accuracy is consistent with chapter 4, where the test dataset was taken on another day after the train dataset.

When the datapoints from the transmitter RPs were used as the train dataset, the accuracy increases to 81.73% for the datasets on 15 March 2022, and 87.21% for the datasets on 29 March 2022. These increases in detection accuracies demonstrate advantages of using the transmitter RPs when RSSIs from the same transmitter and receiver locations change over time.

The results shown in figure 5.4 were obtained from applying the *k*-NN algorithm with k = 11. Figure 5.5 shows the detection accuracies using the transmitter RPs on 15 March 2022 as we vary k from 3 to 21. Over all, the accuracies do not change significantly with *k.* A similar trend is observed for the results from 29 March 2022. Therefore, the middle value of k = 11 is selected for all upcoming experiments.

Figure 5.5: Precisions, recalls, and accuracies for the 15/03/2022 dataset for different values of k for the k-NN algorithm.



5.3 Performance Comparisons among Different Classification Algorithms

Up to now, only the k-NN algorithm only has been used to classify between inside and outside locations. To further investigate the detection performance when the transmitter RPs are used, several well-known classification algorithms were considered, including support vector machine (SVM), decision tree and random forest. As for k-NN, these algorithms are available in the scikit-learn library of Python programming.

The operating principle of each algorithm is explained as follows. In SVM, the 5-dimensional space containing RSSI datapoints from five receivers is divided into two regions to separate datapoints from inside and outside RPs as much as possible. These two regions are then used to classify datapoints from a buffalo. Three different kernel functions are considered: linear, polynomial, and radial basis function (RBF)

(Geron, 2019). For the decision tree algorithm, RSSI datapoints from inside and outside RPs are used to develop a tree of decision rules that can later be used to classify datapoints from a buffalo. In the random forest algorithm, RSSI datapoints from inside and outside RPs are used to construct multiple decision trees that can later be used to classify datapoints from a buffalo through a majority vote based on the results from these trees.

Figure 5.6: Accuracies from well-known classification algorithms when transmitter RPs are used.

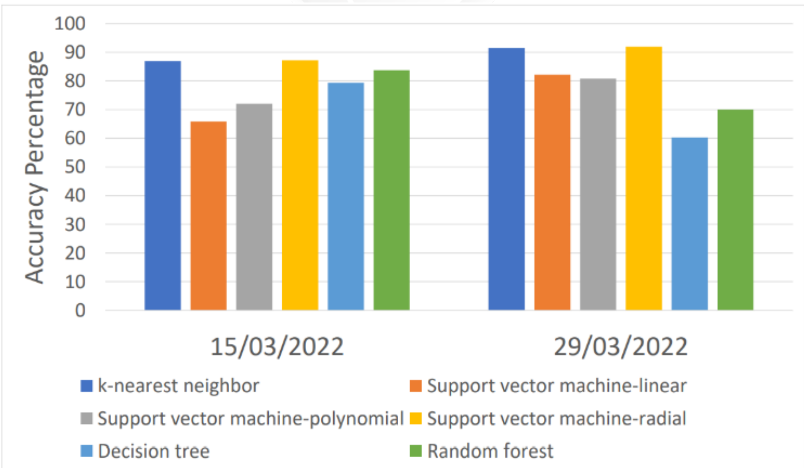


Fig 5.6 shows the accuracy of each algorithm for each day. From the accuracies in Fig. 12, *k*-NN and SVM (with RBF) outperform the other algorithms based on the datasets from both 15 and 29 March 2022, with approximately the same performances between *k*-NN and SVM (with RBF). In particular, the datasets on 15 March 2022 give the accuracy of 86.89% from *k*-NN and 91.51% from SVM (with RBF) while the datasets on 29 March 2022 give the accuracy of 87.24% from *k*-NN

and 91.96% from SVM (with RBF). Therefore, either *k*-NN or SVM (with RBF) is an appropriate choice of a classification algorithm for our detection system.

Between the two algorithms, *k*-NN is attractive due to its simple implementation. While SVM (with RBF) requires more computation on train datasets, it can also be attractive since performing the detection after training does not require high computational complexity.

CHAPTER 6

CONCLUSION

This thesis considers the problem of detecting whether a buffalo is inside or outside the fence area using LoRa RSSIs. Initial experiments indicated that high detection accuracies can be obtained from using 70% of datapoints as the train dataset and 30% of datapoints as the test dataset on each day. In particular, the *k*-NN algorithm with $k = 11$ yields the accuracy of 98.9% on 11/11/2021 and 95.86% on 18/01/2022. Since it may not be practical to get a train dataset every day, using an RSSI dataset on a previous day (11/11/2021) as a train dataset and a test dataset on another day (18/01/2022) was tried and yielded a lower detection accuracy of 76.69%.

Then, was proposed and investigated the use of transmitter RPs to detect whether a buffalo is inside or outside the fence area. In the proposed method, RSSIs from transmitter RPs are used as a train dataset while RSSIs from the buffalo transmitter are used as a test dataset. Experimental results on 15/03/2022 and 29/03/2022 demonstrate that the proposed method yields the detection accuracy of 81.73% on 15/03/2022 and 87.21% on 29/03/2022, which are significantly higher than the accuracy of 45.99% obtained from using train and test datasets from different days (15/03/2022 and 29/03/2022). This is because the use of RSSIs from transmitter RPs instead of RSSIs from a previous day as the train dataset allows the detection algorithm to adapt over time and provide higher detection accuracies in comparison to detection without using transmitter RPs.

Finally, comparisons among well-known classification algorithms, including *k*-NN, SVM, decision tree, and random forest, indicated that *k*-NN and SVM (with

RBF) outperform the other algorithms and are therefore attractive classification algorithms for the developed detection system.

BIBLIOGRAPHY

Abdullahi, U.S., Orisekeh, K., & Nyabam, M. P. (2019). Exploiting IoT and LoRaWAN technologies for effective livestock monitoring in Nigeria. *Arid Zone Journal of Engineering, Technology & Environment*, *1*5, 146-159.

Aernouts, M., Berkven, R. Vlaenderen, K. V. & Weyn, M. (2018). *Sigfox and LoRaWAN datasets for fingerprint localization in large Urban and Rural Areas. MDPI Journal*, *3*(2), 12.

Anjum, M., Khan, M. A., Hassan, S. A., Mahmood, A. & Gidlund, M. (2019). Analysis of RSSI fingerprinting in LoRa networks. *Proceeding in the International Wireless Communications & Mobile Computing Conference (IWCMC)*, Tangier. Morocco.

Anjum, M., Khan, M. A., Hassan, S. A. & Mahmood, A. (December 2020). RSSI fingerprinting-based localization using machine learning in LoRa Networks. *IEEE Internet of Things Magazine*, *3*(4), 53-59.

Ali, I.T., Muis, A. & Sari, R. F. (2021). A deep learning model implementation based on RSSI fingerprinting for Lora-based indoor localization. *EUREKA: Physics and Engineering*, 1, 2471-2474.

Ayele, E. D., Das, K., Meratnia, N. & Havinga, P. J. M. (2018). Leveraging BLE and LoRa in IoT network for wildlife monitoring system (WMS*). Proceeding in the IEEE 4th World Forum on Internet of Things (WF-IoT)*. Singapore.

Chaudhry, A.A., Mumtaz, R., Zaidi, S. M. H., Tahir, M. A. & School, S. H. M. (2020). Internet of Things (IoT) and Machine Learning (ML) enabled livestock monitoring. *Proceeding of the IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, Charlotte, NC. USA.

Choi, W., Chang, S. Y., Jung, Y. & Song, J. (2018). Low-power LoRa signal-based outdoor positioning using fingerprint algorithm. *International Journal of Geo-Information*, *7*(11), 440.

Dieng, O., Diop, B. & Thiare, O. (2017). A study on IoT solutions for preventing cattle rustling in African context. *Proceeding in the ICC conference*. United Kingdom.

Dieng, O., Pham, C. & Thiare, O. (2019). Outdoor localization and distance estimation based on dynamic RSSI measurements in LoRa networks: Application to cattle rustling prevention. *Proceeding in the International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona. Spain.

EverythingRF (2018). *What is LoRa*. Retrieved July 30, 2022, from https://www.everythingrf.com

Fargas, B.C. & Petersen, M. N. (2017). GPS-free geolocation using LoRa in low-power WANs. *Proceeding in the Global Internet of Things Summit (GIoTS)*, Geneva. Switzerland.

FarmingUK Team (2015). *High tech 'cow bell' brings revolution to cattle movement*. Retrieved July 30, 2022, from https://www.farminguk.com

Geron, A. (2019). *Hands-on machine learning with scikit-learn keras & tensor flow*. O’Reilly Media, Inc.

Javatpoint (n.d.). *Support vector machine algorithm*. Retrieved July 30, 2022, from https://www.javatpoint.com

Joshitha, C., Kanakaraja, P., Bhavani, M. D., Raman, Y. N. V. & Sravani, T. (2021). LoRaWAN based cattle monitoring smart system. *Proceeding in the 7th International Conference on Electrical Energy Systems (ICEES)*, Chennai. India.

Kulaib, A. R., Shubair, R. M., Qutayri, M. A. A. & Jason, W. P. Ng. (2011). An overview of localization techniques for Wireless Sensor Networks. *Proceeding in the International Conference on Innovations in Information Technology,* Abu Dhabi. United Arab Emirates.

KWKWII (2018). *Spreading Factor, Bandwidth, Coding Rate and Bit Rate in LoRa (English)*. Retrieved September 5, 2022, from https://www.josefmtd.com

Lam, K.H., Cheung, C. C. & Lee, W. C. (2017). LoRa-based localization systems for noisy outdoor environment. *Proceeding in the IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Rome. Italy.

Li, Q., Liu, Z. & Xiao, J. (2018). Collection collar for vital signs of cows on the grassland based on LoRa. *Proceeding in the IEEE 15th International Conference on e-Business Engineering (ICEBE)*, Xi'an. China.

LoRaWAN (n.d.). *LoRaWAN architecture*. Retrieved July 30, 2022, from https://www.thethingsnetwork.org

Luo, J., Ito, A., Sasak,i A. & Hasegawa, M. (2020). Sensor network for monitoring livestock behaviour. *Proceeding in the IEEE SENSORS*, Rotterdam. Netherlands.

Mallon, S. (n.d.*). IoT Is The Most Important Development Of The 21st Century*. Retrieved July 30, 2022, from https://www.smartdatacollective.com

Markqvist (2020). *LoRa bitrate calculator and understanding LoRa parameters*. Retrieved August 25, 2022, from https://www. unsigned.io

Molina, F.M., Garcia, J. N., Aguirre, K. P., Maqueda, I. G., Ginel, J. E. G., Varo, a. G. & Marin, D. C. P. (2019). A low-cost IoT-based system to monitor the location of a whole herd. *MDPI Journal, 19*(10), 2298.

Munoz, C., Huircan, J., Huenupan, F. & Cachana, P. (2020). PTZ camera tuning for real time monitoring of cows in grazing fields. *Proceeding in the IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, San Jose. Costa Rica.

Panicker, J.G., Azman, M. & Kashyap, R. (2019). A LoRa wireless mesh network for wide-area animal tracking. *Proceeding in the IEEE International Conference* *on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore. India.

Stojkoska, B.R., Bogatinoska, D. C. & Malekian, R. (2018). Real-time of things architecture for wireless livestock tracking. *Telfer Journal, 10*(20), 74-79.

Spsmartplants (n.d.). *Smart farm*. Retrieved July 30, 2022, from https://www.spsmartplants.com

Zinas, N., Kokkonis, G. & Kazanidis, I. (2017). Proposed open source architecture for Long Range monitoring. The case study of cattle tracking at Pogoniani. *Proceeding in the 21st Pan-Hellenic conference*, Larissa. Greece.

APPENDICES

APPENDIX A

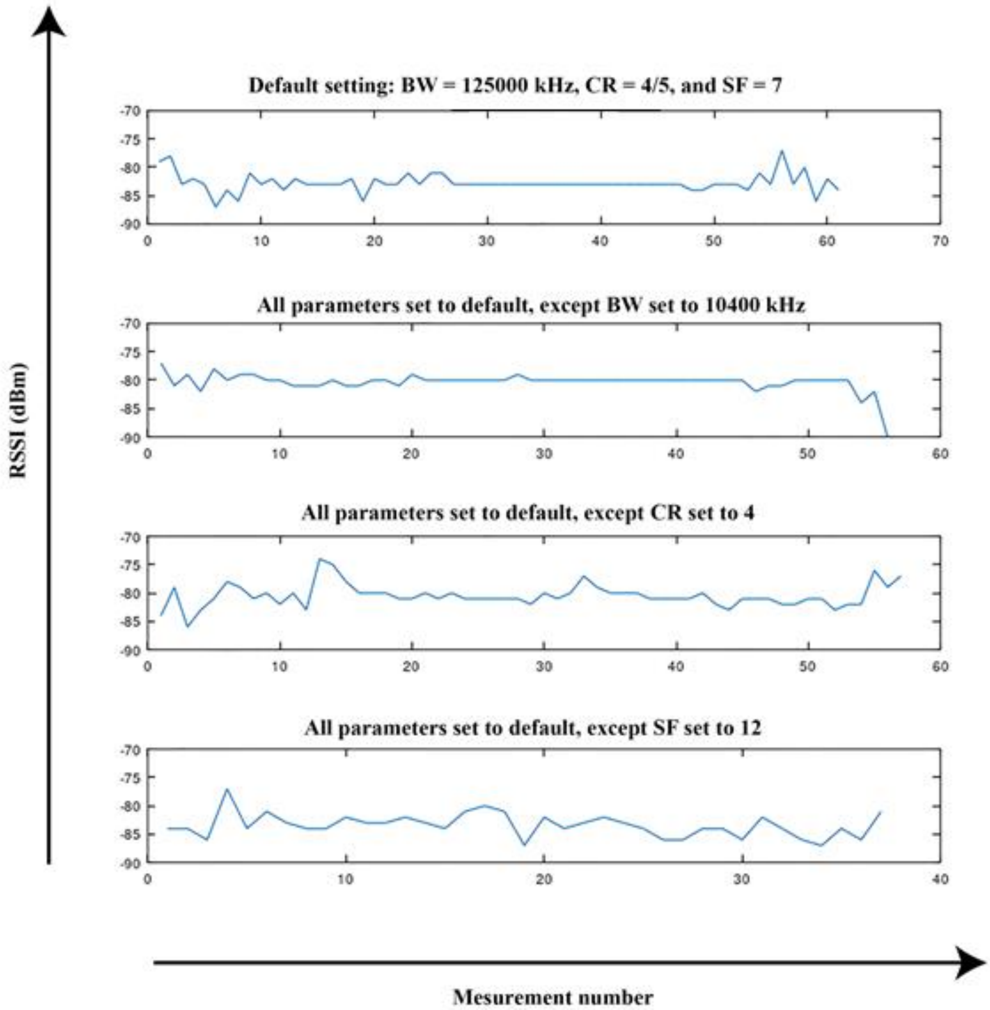


Figure A.1: Variation of LoRa RSSIs when transmission parameters are adjusted

All experiments in this research have been based on using the default parameters in LoRa modules. To see the effects of adjusting LoRa parameters, experiments with different LoRa parameters have been conducted. These experiments were done in the hallway of BU-CROCCS, which located on the sixth floor of the Engineering Building Bangkok University. LoRa parameters including bandwidth (BW), coding rate (CR), and spreading factor (SF) are adjusted to test the stability of the LoRa RSSI values. The result of each test is compared to the default setting, which sets BW = 125000 kHz, CR = 4/5, and SF = 7. Each parameter is adjusted one by one and then tested for 5 minutes.

Each test was based on line-of-sight transmissions in indoor situations with a distance of 50 m. Figure A.1 shows experimental results obtained. First, the BW is reduced from the default value of 125000 kHz to 10400 kHz, keeping the default values for the other parameters. Next, only the CR is changed from the default value of 4 (minimum) to 8 (maximum). Finally, only the SF is changed from the default value of 7 (minimum) to 12 (maximum). Note that each parameter adjustment reduces the bit rate. The goal is to find out whether a reduction in the bit rate could increase the stability of RSSI values.

Over all, the results show that using a narrower bandwidth can make the LoRa RSSI values more stable, as can be observed in Fig A.1. Therefore, as a future research investigation, LoRa RSSI-based localization using small bandwidths is recommended.

APPENDIX B

The default of each classification algorithms in the scikit-learn library

1. Decision tree

Decision tree classifier takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training samples, and an array Y of integer values, shape (n_samples,), holding the class labels for the training samples:

```
>>>
>>> from sklearn import tree
>>> X = [[0, 0], [1, 1]]
>>> Y = [0, 1]
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(X, Y)
```

After being fitted, the model can then be used to predict the class of samples:

```
>>>
>>> clf.predict([[2., 2.]])
array([1])
```

2. Random forest

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap = True (default), otherwise the whole dataset is used to build each tree.

Examples:

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> from sklearn.datasets import make_classification
>>> X, y = make_classification(n_samples=1000, n_features=4,
...                            n_informative=2, n_redundant=0,
...                            random_state=0, shuffle=False)
>>> clf = RandomForestClassifier(max_depth=2, random_state=0)
>>> clf.fit(X, y)
RandomForestClassifier(...)
>>> print(clf.predict([[0, 0, 0, 0]]))
[1]
```

3. Support vector machine

A Support Vector Machine (SVM) is a very powerful Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection. The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classifies the data points.

Example of the SVM command in Python:

Support vector machine linear

```
from sklearn.svm import LinearSVC
model = LinearSVC(loss='hinge', dual=True)
model.fit(X_train, y_train)
print_score(model, X_train, y_train, X_test, y_test, train=True)
print_score(model, X_train, y_train, X_test, y_test, train=False)
```

Support vector machine Polynomial

```
from sklearn.svm import SVC
# The hyperparameter coef0 controls how much the model is influenced
by high degree ploynomials
model = SVC(kernel='poly', degree=2, gamma='auto', coef0=1, C=5)
model.fit(X_train, y_train)
print_score(model, X_train, y_train, X_test, y_test, train=True)
print_score(model, X_train, y_train, X_test, y_test, train=False)
```

Support vector machine radial

```
model = SVC(kernel='rbf', gamma=0.5, C=0.1)
model.fit(X_train, y_train)
print_score(model, X_train, y_train, X_test, y_test, train=True)
print_score(model, X_train, y_train, X_test, y_test, train=False)
```

BIODATA

Name – Last name: Thipchinda Prasayasith

Address: Phonsavanh nuea, Sisattanark district, Vientiane, Laos

Email: nino.thipchinda@gmail.com

Contact number: +8562058892890

Education Background

: Master of Engineering in Electrical and Computer Engineering, Bangkok University, Thailand 2020

: Bachelor of Electronics Engineering, National University of Laos, 2019