UNIVERSITY OF LIVERPOOL
DEPARTMENT OF ELECTRICAL ENGINEERING & ELECTRONICS

Final report for project 'Deep Learning based RFF Identification for LoRa Using Raspberry Pi'

Author: Xiuyuan Chen (201447371)

Project Supervisor: Dr Junqing Zhang

Project Assessor: Dr Valerio Selis

Declaration of academic integrity

The standard University of Liverpool statement of academic integrity [6] should go here as follows:

I confirm that I have read and understood the University's Academic Integrity Policy.

I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.

I confirm that I have not copied material from another source nor committed plagiarism nor fabricated, falsified or embellished data when completing the attached piece of work. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work.

SIGNATURE…….....……
DATE……

Abstract

Radio frequency fingerprint identification (RFFI) is a novel non-cryptographic device authentication technique that identifies wireless devices through the hardware characteristics at physical layer. The final year project aims of developing a deep learning-based RFFI for Long Range (LoRa) network using Raspberry Pi. Five LoPy4 devices were deployed as the signal transmitters. RTL-SDR was utilized as the signal receiver. Signal transmitting algorithm was implemented in PC by MicroPython to specify the configuration parameters of LoRa modulation. LoRa preambles containing eight symbols were extracted by employing time synchronization (TS) algorithm. It was experimentally found that estimated carrier frequency offset (CFO) varies over time, which could degrade the classifier accuracy. Therefore, CFO compensation was exploited to compensate the frequency drift. 5500 preamble packets were gathered from five LoRy4 devices and represented as spectrograms for training, validation and testing phases. Three neural network (NN) architectures were designed in total, which were two convolutional neural networks (CNN) and one multilayer perceptron (MLP). The generalization abilities of three NN models were compared, and the one with three convolution layers reached the best classification performance. All these algorithms were implemented in Python scripts. A graphical user interface (GUI) application for RFFI system was developed using PyQt5 and run in Raspberry Pi for device inference. The designed RFFI system can achieve a classification accuracy of 95.60%.

Contents

1 Introduction

The Internet of Things, referred to as IoT, whose inception started in 2009, has now connected to more than 50.1 billion smart devices [1]. Unlike the Internet in the traditional sense, the Internet of Things encompasses a network of physical objects. Connected sensors and wearables, such as toothbrushes, watches and thermometers, can share data with each other, analyse information and complete some particular tasks for better user experience. The proliferation of IoT devices is extremely rapid. They are deployed in various fields of life. It is estimated by International Data Corporation (IDC) that the global revenue of IoT has reached approximately $8.9 trillion by the end of 2020 [2]. Notwithstanding the widespread popularity of IoT, its lack of security is a significant drawback that has been receiving sustained concerns [3]. Universally, IoT security problems involve six dimensions, which are network protection, device authentication, data encryption, IoT API and PKI, and software defence. Among them, device authentication is a of critical importance security approach that permits legitimate users to access the network while blocking malicious users. Figure 1 presents damages caused by unauthorized access. Without robust authentication system, intruders can arbitrarily tamper with private data and even incapacitate the computer. Many network authentication methods use software addresses such as Internet Protocol (IP) and Media Access Control (MAC) addresses to prevent unauthorized access [4]. Nevertheless, these software addresses can be factitiously tampered or forged. Malicious users can impersonate the legitimate users to access the confidential data once obtaining the cryptographic keys or security certificates. Furthermore, low-cost IoT devices are difficult to perform advanced encryption algorithms or multiple authentications, as energy consumption and hardware capacity are two aspects that limit their computational performance.

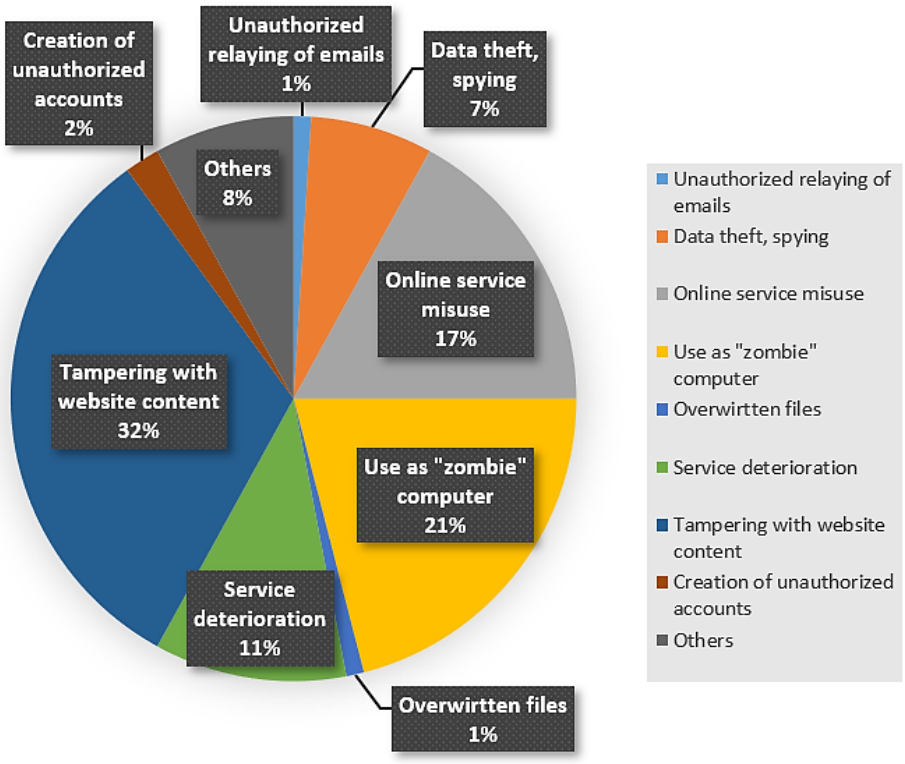


Figure 1. Contents of unauthorized access damage (data referred from [23])

To address the vulnerabilities, device fingerprinting technique has emerged to gather IoT device features and treat them as device-specific fingerprints for identifying devices [5]. A variety of possible features have been tested by predecessors including network traffic characteristics and channel characteristics. These features are unfortunately unreliable because they are either time-consuming or environmental-dependent. A superior device authentication scheme called radio frequency fingerprint identification (RFFI) extracts unique features of wireless devices from their transmitted signals. These radio frequency (RF) characteristics are already introduced to hardware devices during the manufacturing process, and they are inherent to electronic components. Therefore, RF fingerprints are arduous to be falsified and they remain relatively stable in the presence of node mobility and environment changes. In addition, RFFI never enhances network security at the cost of communication latency and computational complexity, which is very essential to low-cost IoT devices.

Overall, the project intends to explore the feasibility of RFFI technique on limited power and/or cost IoT devices. The designed RFFI scheme will be customized for Long Range (LoRa) network, as LoRa enables energy management. The remainder of the report will be organized

as follows. In the Literature Survey section, some relevant researches about RFFI are presented. In the Industrial Relevance section, the report describes the market potential and industrial applicability of RFFI technique. The Theory section introduces some theoretical backgrounds of involved technologies, and the Design section demonstrates layouts and models proposed for simulating a RFFI system. In the Experimental Method section, details of required equipment and algorithm configurations are provided. The experimental results and referred algorithms are thoroughly analysed and evaluated in the Results and Calculations section. In the Discussion section, the experimental contributions, errors, and some of noteworthiness future problems are raised. The culminating Conclusions section concisely summaries the project achievements.

2 Literature Survey

Cybersecurity has always been an issue that has received global attention. Many experts and scholars are paying great efforts on searching and solving security vulnerabilities. In 2016, Zou et al. [4] analysed existing security threats and weaknesses of wireless networks at different protocol layers, and emphasised that software address authentications are vulnerable to diverse malicious attacks. One primary attacking mean proposed calls MAC spoofing. It happens when a malicious network node falsifies a MAC address to conceal its identity for illegal purposes. Furthermore, he claimed that IP address also suffers the same type of attack, and added that responding to these forged IP addresses sometimes significantly waste the network capacity and, more seriously, paralyze the whole network. It is obvious that many security risks exist in the IP and MAC addresses. Their susceptibility to forgery is not suitable to perform device authentications. As Xu et al. [5] suggested, a qualified authentication scheme should satisfy two properties, which are forgery-impossible and location-independent. Although networks can reduce address attacks by exploiting multiple authentication mechanisms at different layers [4], it can be really challenging for device performance as many algorithms are required to be implemented.

Xu et al. [5] introduced a promising non-cryptographic based authentication technique named device fingerprinting that can extract unique features from signal transmissions in wireless communication. He has enumerated many features at different layers that are capable to be the device fingerprints. Sivanathan et al. [6] have investigated the feasibility of using network

traffic characteristics to classify IoT devices. Their classification model has reached approximately 99 percent accuracy. However, it took over weeks to gather the network traffic characteristics. If other devices join the network, the system must recollect the data packets, which will waste a lot of time. Meanwhile, Liu et al. [7] performed device authentication with the use of channel state information (CSI) and received signal strength (RSS). They have compared the authentication accuracy of CSI-based method with which of RSS-based method, and discovered that CSI-based method would have a better authentication performance of more than 90 percent accuracy. Nevertheless, these two methods are influenced by channel conditions, so that two distant locations may generate dramatically different results if the same transmitter is used [5]. A practical device fingerprinting identification system should be efficient and stable regardless of environmental changes.

Based on the physical (PHY) layer, Shen et al. [8] proposed a concept that is to extract hardware imperfections of IoT devices to form device fingerprints, which refers to radio frequency fingerprint identification (RFFI) technique. These imperfections are introduced during the manufacturing process of IoT devices, which can lead internal transistors to have different parameters such as channel width, channel doping and oxide thickness [5]. The characteristics of all electronic components combine together to be the unique RF fingerprint for IoT devices. RFF can be detected from tiny frequency fluctuations of transmitted signals. Previously, Gopalakrishnan et al. [9] have investigated the application of deep learning on radio fingerprinting. They demonstrated that complex-valued convolution neural networks (CNNs) can be used to fingerprint wireless devices with high accuracy. The same result was obtained by Das et al. [10], who have used 30 LoRa devices to simulate the RFFI system and got 90% accuracy with the use of CNN. Other classification models, such as multilayer perceptron (MLP) and recurrent neural network (RNN), were tested but got only 50%-85% accuracy [8]. Additionally, CNN requires less learnable parameters than other NNs, which reduces the training time [8]. As a result, the project will use CNN to learn RF fingerprints. 90% classification accuracy is the final objective as many researchers viewed it as a minimum standard of good RFFI system.

Recently, Al-Shawabka et al. [11] simulated their RFFI system in Wi-Fi network and used CNN to learn RFFs. However, they found that the classification accuracy dropped greatly if test packets were collected on different days. With regard to the phenomenon, Andrews et al. [12] carried out experiments and concluded that temperature fluctuations can result in

frequency offsets in received signals, as embedded oscillator is very sensitive to temperature variations. This is the reason why accuracy is low if test data is collected on different days. Carrier frequency offset (CFO) algorithms were implemented to eliminate the influence of temperature [13]. To guarantee high accuracy, RFFI system in the project should perform CFO estimation and compensation.

Finally, signal representation is an important factor that should be considered. Riyas et al. [14] used software defined radio (SDR) to sample the signal waveforms. Their CNN classifier operated on raw In-phase/Quadrature (I/Q) samples of a sequence length of 128. The experimental accuracy could reach 90-99 percent. However, other signal representations were not tested. Shen et al. [8] designed a CNN model and compared the difference between directly inputting I/Q samples and inputting spectrograms. It was discovered that the training time of inputting I/Q samples was 55 minutes more than which of inputting spectrograms. Two models got similar accuracy but CNN for I/Q samples required more learnable parameters. Accordingly, the project tends to use signal spectrograms as inputs, because CNN is more efficient in processing pictures.

3 Industrial Relevance

The pervasiveness of IoT devices has been spreading to various domains during the past decades, including home automation, individual health care and smart mobility [16]. They hence progressively become an indispensable part of life. According to the latest research released by IoT Analytics [17], global IoT device connections have reached approximately 11.7 billion in 2020 and are predicted to reach 30.9 billion in 2025. Furthermore, as statistics illustrated in Figure 2, it is predicted that there would be approximately 75.44 billion IoT connected devices worldwide, which is 2.5 times the number of IoT devices in 2020. The manufacturing market size of IoT is expected to increase from $33.2 billion in 2020 to $53.8 billion by 2025, at a Compound Annual Growth Rate (CAGR) of 10.1% [19]. It is apparent that IoT market is growing at an alarming rate to fill the huge gap in demand.

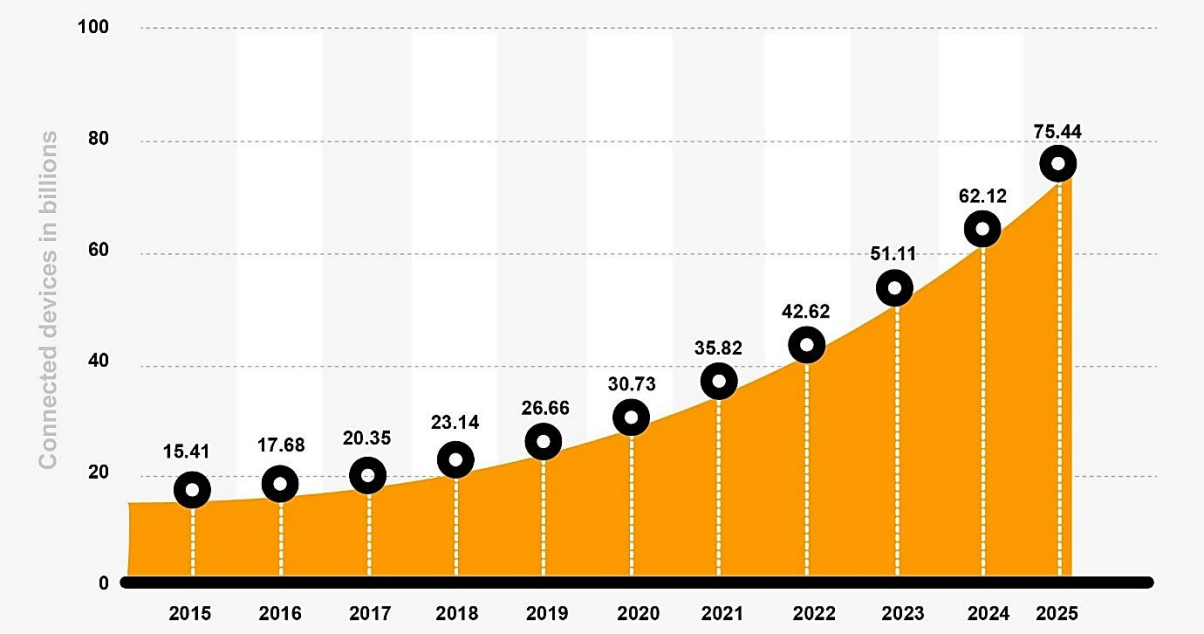


Figure 2. Predictive worldwide growth trends in IoT connected devices from 2015 to 2025 (taken from [18])

Although the Internet of Things has brought huge wealth to people, IoT security is an inevitable problem. McAfee released a report in 2020 showing that the world economy has lost more than $1 trillion because of cybercrime, which is nearly one percent of global GDP [20]. Beaming also found that more than 1.5 million UK organisations became victims of cybercrime in 2019, which was equivalent to 25% of whole UK businesses [24]. Many cyber-attacks especially unauthorized access can cause disastrous impacts to consumer world. Loss of intellectual properties will cause consumers to distrust IoT. Figure 3 shows the categories of cyber incidents and financial losses to UK businesses. Although the total cost of IoT hacking to UK businesses is relatively low, the average cost per incident is contrastingly high, reaching approximately 30 thousand pounds. Other cyber incidents can as well indirectly endanger IoT. Increased security threats and data privacy are two openly-questioned IoT issues that have not received well resolve. IoT should conduct safety measures to prevent cybersecurity threats. Authenticity, confidentiality, integrity and availability are four prime network security requirements [4]. In contrast to wired communication, the authenticity and availability of wireless communication is more vulnerable to spoofing and jamming attacks as wireless signals can be easily intercepted and interfered [4]. Even though IoT contains both wired and wireless communication, people tend to use wireless IoT devices due to their convenience and portability.

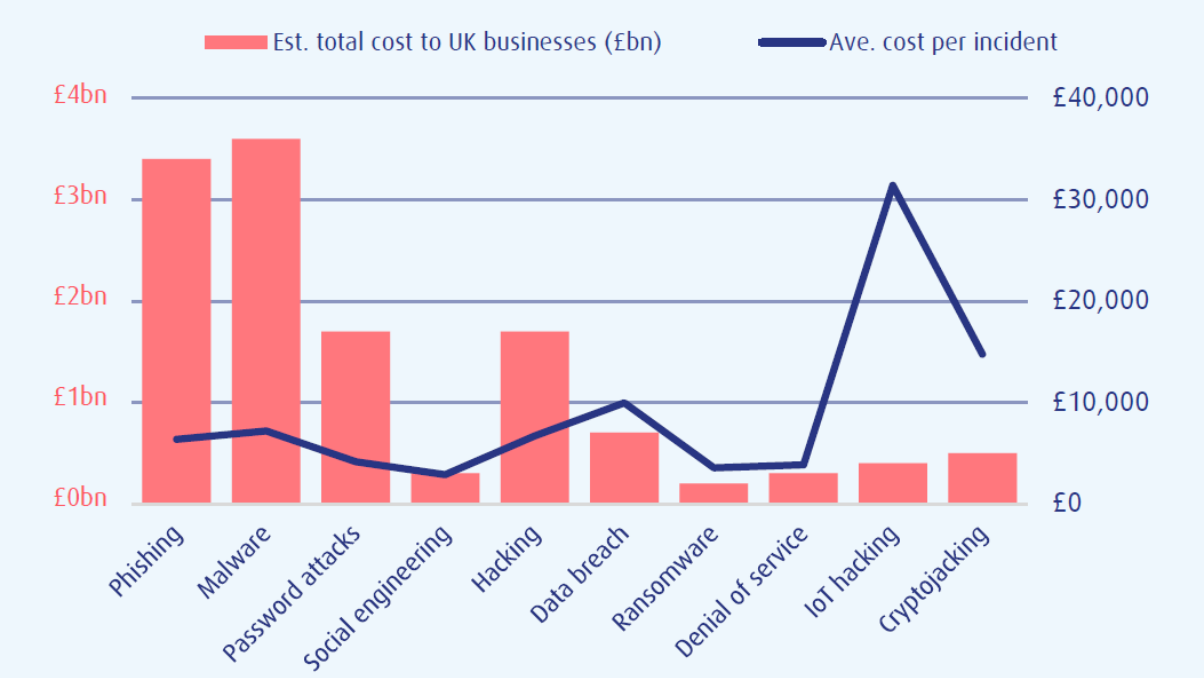


Figure 3. The estimated cost of different cybercrimes surveyed by Beaming (taken from [24])

Typical IoT networks are based on OSI layered protocol architecture. The Internet protocol stack consists of 5 layers, which are physical layer, MAC layer, network layer, transport layer and application layer. Malicious attacks can occur at all layers. For example, at the MAC layer, malicious nodes can forge or steal assigned MAC addresses of legitimate nodes to deceive the user authentication system [8]. With the fake identities, attackers can conduct various illicit cybercriminal activities, including thieving private data, falsifying financial information, etc. The similar attacks also emerge at IP address of the network layer. Main types of wireless attacks at network and MAC layers are listed in Table 1. They seriously threaten personal privacy and property safety. According to the 2020 Norton cyber safety report, two-thirds of 500 million Internet consumers reported that they were more worried about their privacy than ever (67%) and showed great uneasiness about their identity being stolen (66%), and 92% of them expressed concerns on data privacy [21]. People are attaching more and more importance to individual privacy. It is extremely imperative to improve the authentication accuracy under real scenarios. High-computing-power IoT devices can perform encryption algorithms or multiple authentications for high security. However, it is more challenging for IoT devices with computational power and memory constraints to perform complicated authentication approaches. As a result, a new authentication method should be applied.

Table 1. Malicious attacks at the MAC and Network layers

Layer	Attacks
MAC Layer	MAC spoofing
	Identity theft
	MITM attack
	Network injection
Network Layer	IP spoofing
	IP hijacking
	Smurf attack

RFFI technique is a device fingerprinting authentication scheme that identify wireless devices from their transmitted signals. The device features hidden within emitted signals are originated from hardware defects. RFF is unique because no two devices would have exactly the same physical characteristics. Different from software addresses, it costs enormous efforts to forge radio frequency fingerprints. Meanwhile, RFFI system is a passive non-cryptographic security scheme. It will not introduce supernumerary computational burden and communication latency to IoT devices. Although the technique is still immature, the application of RFFI technique is expected to appropriately remedy some vulnerabilities of identity authentication. At least five potential aspects would benefit from RFFI technique. Firstly, limited computing power devices would have strong authentication ability as high-performance computers do, which further increases the propagation of low-power IoT devices [22]. The IoT market can be driven to grow more positively. Furthermore, mobile apps would not need to perform cumbersome biometric and password authentication for the sake of correctly recognizing the identity of a device. Thirdly, RFFI technique would decrease the manufacturing cost of IoT devices. Extra device firmware and security algorithms are not necessary to be developed. In addition, RFFI technology improves the efficiency of signal transmissions, as ID numbers like MAC address are not required to be involved in the signal header [22]. Finally, nowadays, online banking and mobile payment have been democratized due to the widespread use of smartphones. Promoting RFFI technique would recover unnecessary property losses by prevent attackers from logging into bank accounts.

4 Theory

The project simulates RFFI system in LoRa network. Time synchronization algorithms and CFO estimation and compensation algorithms should be implemented to extract LoRa packets. Convolution neural network is adopted to discern five LoRa devices. This section will introduce the theoretical foundations of aforementioned terms.

4.1 LoRa Network

Long Range (LoRa) network is a modulation technique patented by Semtech that defines the physical layer [26]. The upper layers are implemented by Long Range Wide Area Network (LoRaWAN), which mainly manages communication. According to Figure 4, LoRa has lower bandwidth than Wi-Fi and Cellular, enabling it to have lowest power. Moreover, the transmission range of LoRa is very far. The straight-line distance can reach over 15 kilometres, and the urbanized area can reach 2-5 kilometres [25]. LoRa also features low data rate. These three advantages make LoRa to be very practical in low energy consumption devices. As mentioned above, the project aims of developing RFFI system to support low-power and low-cost IoT devices. Since the objective of inventing LoRa technique is to increase battery lifetime and reduce device cost [27], it would be the most appropriate network to be applied in the project.

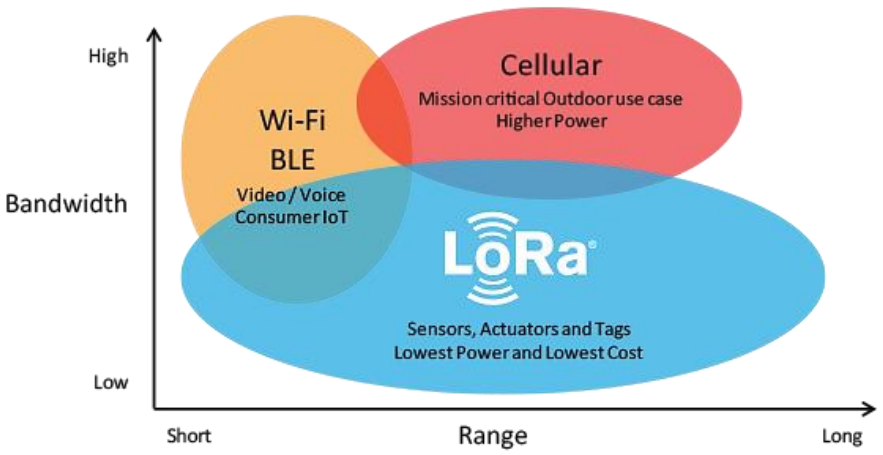


Figure 4. Comparison between Wi-Fi, Cellular and LoRa (taken from [25])

LoRa uses a spread spectrum modulation technique that is based on chirp spread spectrum (CSS) technology to transmit signals [26]. LoRa signals are very resistant to interference and

attenuation. Configuration parameters of LoRa radio are listed in Table 2. CF is the middle frequency of transmission band. In European countries, the prescriptive LoRa frequency must within ISM band frequencies (863 MHz - 870 MHz) [27]. The project operates at 868 MHz as this channel frequency is license exempt and mandatorily implemented in every end-device. SF defines the number of bits in each symbol. It is an integer varying from 7 to 12. Greater SF will result in more time in sending packets. BW is the frequency range of baseband signal. In LoRa, only 125 kHz, 250 kHz and 500 kHz can be selected. Lowest bandwidth allows longest transmission range. CR refers to the number of redundant bits (5-8) encoded in every four transmission bits for the purpose of forward error correction [27]. The smaller the coding rate, the longer it takes to transmit data on air. The values are assigned in the project to ensure long communication range and low transmission time.

Table 2. Configuration parameters of LoRa modulation

Parameter	**Value**
Carrier Frequency (CF)	868 MHz
Spreading Factor (SF)	7
Bandwidth (BW)	125 kHz
Coding Rate (CR)	4/5

A LoRa packet basically contains four parts, as is depicted in Figure 5. A LoRa frame begins at the preamble segment. Preamble is used to perform synchronization between transmitter and receiver [27]. An optional header is followed comprising basic information of LoRa modulation. Payload embraces the transmission data. Cyclic Redundancy Check (CRC) is optionally transmitted at the end.

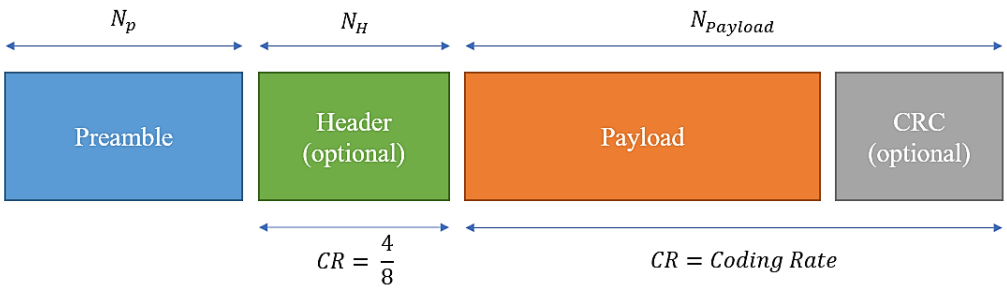


Figure 5. LoRa packet structure (taken from [27])

The difference of waveform between preamble and payload can be viewed in Figure 6. Preamble part of LoRa packet is repeated frequency jumps, while payload part depends on the encoded data. They are both modulated to be linear frequency variations. Ordinarily, there are eight symbols (up-chirps) in preamble, where frequency increases linearly with time. The project selects preamble part to be the inputs because preamble will not change pattern whenever transmission data changes. If payload part is used as inputs, classifier can differentiate devices through the transmission data rather than device fingerprints.

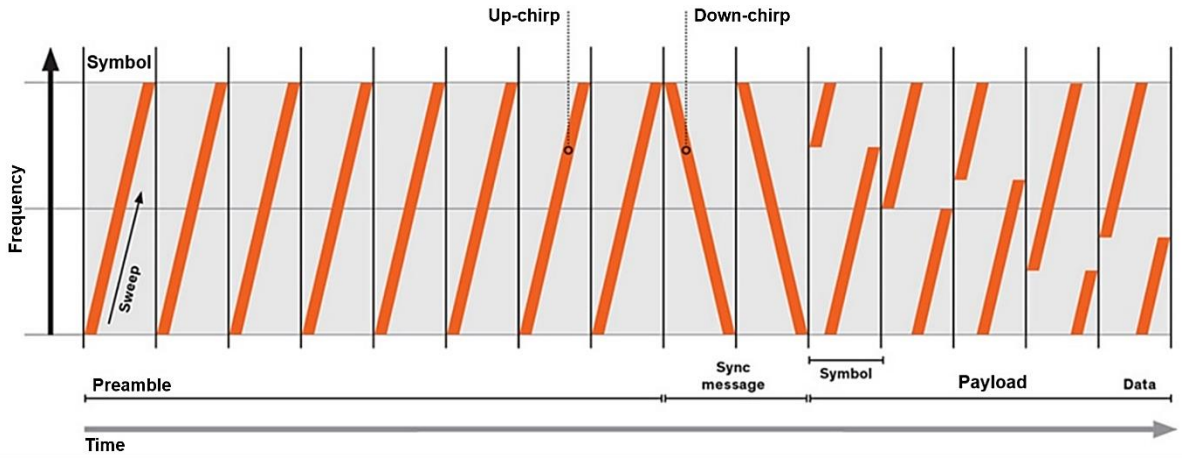


Figure 6. Frequency sweeps of LoRa signal (taken from [28])

The sample length of one LoRa symbol can be calculated as follows:

$$L = \frac{T}{T_s} = \frac{2^{SF}}{B \cdot T_s} \tag{1}$$

where T is the symbol duration, T_s is the sampling duration, SF is the spreading factor, and B is the bandwidth. If SF is 7, B is 125 kHz and f_s is 1 MHz, the sample length of preamble should be 8192. The instantaneous frequency of a sampled baseband up-chirp increases linearly from $-\frac{B}{2}$ to $\frac{B}{2}$, defined as

$$f_{up-chirp}[n] = -\frac{B}{2} + \frac{B}{T} n T_s \tag{2}$$

4.2 Time Synchronization

Under real circumstances, received signal is composed of channel noise and LoRa packets. The time synchronization (TS) algorithm is employed to extract LoRa preambles from received signal. Two algorithms are applied, one to detect the arrival of a LoRa signal, the other to find the exact starting point of the LoRa packet. The proposed Schmidl-Cox algorithm can be used to estimate the start of a LoRa symbol in the time domain [29]. The mathematical expression is given as

$$M[n] = \frac{|\sum_{k=0}^{L-1} r^*[n+k]r[n+k+L]|}{\sum_{k=0}^{L-1} r^*[n+k+L]r[n+k+L]} \tag{3}$$

where L is the sample length of a symbol, $r[n+k]$ is the sampled baseband signal, and $r^*[n+k]$ is its conjugate. The principle of equation (3) is founded on autocorrelation. It generates a discrete sequence M that assigns different values to noise and packets. Values of packets are much greater than which of noise. A threshold can be set to roughly locate the starting position of a packet. However, a small section of channel noise can be ineluctably involved in the extracted packet.

A more precise synchronization algorithm is provided by Robyns et al. [30], based on cross-correlation, which can be mathematically expressed as

$$ind = argmax_{i\in\{0,1,\ldots,L-1\}}(\sum_{n=0}^{L-1} f_{ideal}[n] \cdot \widehat{f_r}[n+i]) \tag{4}$$

where ind is the index of the accurate starting point, $f_{ideal}[n]$ denotes the instantaneous frequency of an ideal baseband up-chirp (according to equation (2)), and $\widehat{f_r}[n]$ indicates the instantaneous frequency of coarse synchronized baseband signal $\hat{r}[n]$. Equation (4) produces a specific location index of $\hat{r}[n]$. After calculating the sample length of preamble (according to equation (1)), a LoRa preamble can be obtained by intercepting $\hat{r}[n]$ from ind to ind + preamble length.

4.3 CFO Estimation and Compensation

Owing to the influence of temperature, instantaneous frequency of the baseband basic chirp will undergo a frequency offset. The frequency offset can vary substantially during the collection process of packets, which will confound the classification model. Carrier frequency offset (CFO) algorithms are employed to estimate and compensate the frequency offset. Because of the linear frequency variation in chirps, calculating the mean frequency value of received preambles can estimate the coarse frequency offset, given as

$$\Delta\hat{f}_{coarse} = \frac{1}{L}\sum_{n=0}^{L-1} f[n] \tag{5}$$

It is not sufficient to just estimate the coarse frequency offset. A teeny frequency offset still remains to be compensated. Based on the repetitiveness of preambles, a sophisticated CFO estimation algorithm is employed to reckon the residual offset. The estimated fine frequency offset is expressed as

$$\Delta\hat{f}_{fine} = -\frac{1}{2\pi T_s L}\angle\left(\sum_{n=0}^{L-1} r'[n]\cdot r'^{*}[n+L]\right) \tag{6}$$

where T_s is the sampling duration, $r'[n]$ is the coarse frequency compensated baseband signal, and $\angle$ returns the angle of the variable. The received signal, $r[n]$, can be compensated after the two frequency offsets have been estimated, mathematically represented as

$$r''[n] = r[n]\cdot e^{-j2\pi\left(\Delta\hat{f}_{coarse}+\Delta\hat{f}_{fine}\right)nT_s} \tag{7}$$

Subsequently, $r''[n]$ is represented as spectrogram and send to the classification model for training and testing.

4.4 Convolution Neural Network

Convolution neural network (CNN) is a branch of deep learning technique that is adept in analysing visual images [31]. The design principles of CNN are in conformity with the vision processing in living organisms [32]. A CNN structure consists of three fundamental domains, which are feature extraction domain, feature classification domain and probabilistic distribution domain, as demonstrated in Figure 7. In feature extraction domain, convolution layer possesses many kernels that respectively perform convolution operations on image matrix to extract features from the input image. Each kernel has different initial weights and bias. A feature map is abstracted after the image matrix passes through one kernel. The rectified linear activation layer (ReLU) does nonlinear mapping to the outputs of the convolution layer. Pooling layer reduces the parameters of feature maps by summarizing neuron clusters at the previous layer to be a single neuron at the next layer [32]. For best feature extractions, plural convolution-pooling pairs are connected adjacently. The closer to the output, the more abstract the features are extracted. For example, the first few layers of convolution kernels extract the hierarchical progressive relationship of stripes, textures, distribution boundaries, object contours. At the terminal, feature maps are flattened for inputting to the next layer.

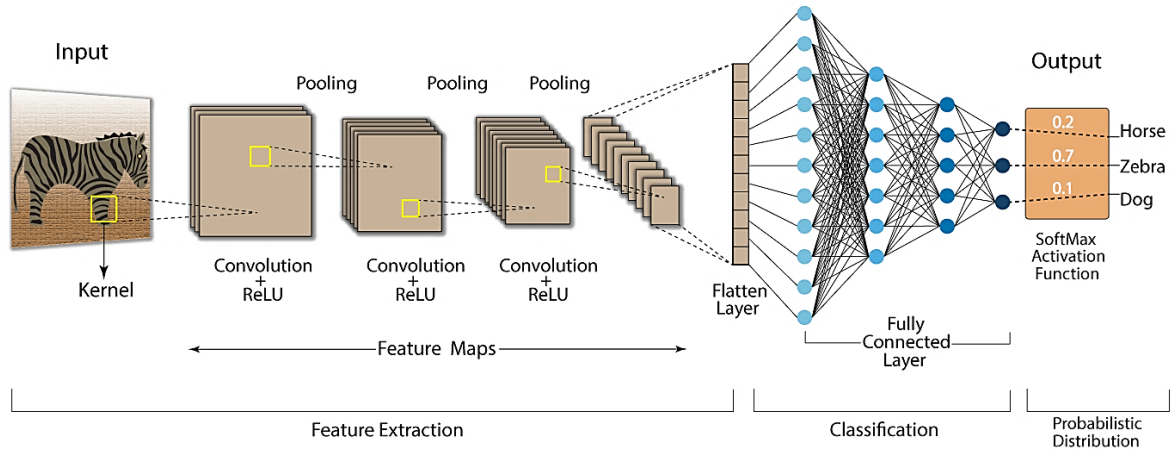


Figure 7. Image classification process of CNN (taken from [31])

In the classification domain, fully connected layer classifies multiple features hidden in feature maps. One neuron at the next layer is connected to all neurons at the previous layer. Output from the former neuron need to multiply the corresponding weight of the connection to be the input of the latter neuron, as shown in Figure 8. An activation function is followed by each neuron to perform logical operations on the sum of all inputs. Eventually, in the probabilistic

distribution domain, SoftMax function is employed to assign possibility to each class. For instance, in Figure 7, an image of zebra is inputted into the CNN model. Three possibilities are distributed to three animals, 0.2 to horse, 0.7 to zebra, 0.1 to dog. The highest probability is treated as the final predicted result. Therefore, in the case, zebra is the recognition result.

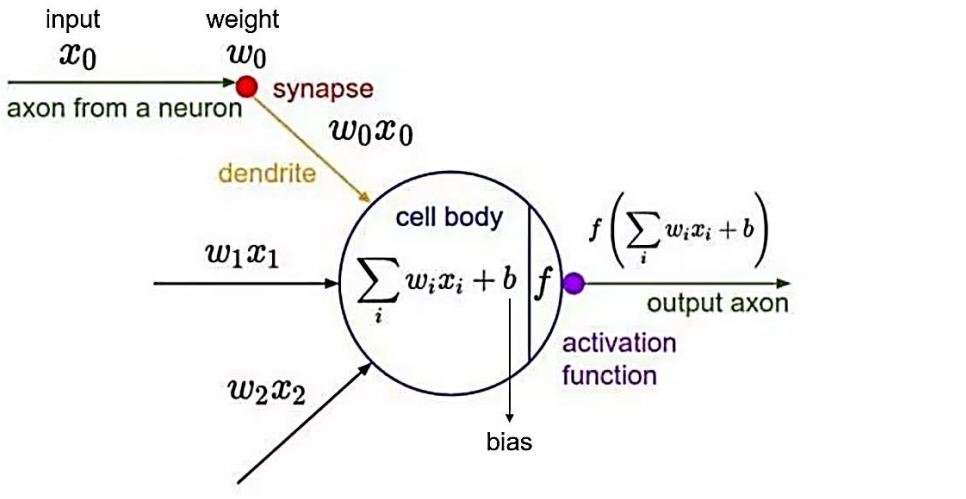


Figure 8. Mathematical model of a neuron (taken from [33])

During the learning process, the predicted possibilities are compared with true possibilities. Errors are calculated and reversely propagated to each layer for updating parameters [32]. By iteratively adjusting the weights and biases, the CNN model learns how to identify one object.

5 Design

This section elucidates the hardware and software designs that were carried out to implement the RFFI system.

5.1 Equipment Layout

The experimental layout of RFFI system was designed, as illustrated in Figure 9. PC is used to actualize the signal transmitting algorithm. The algorithm initialises the LoRa communication network. A transmitter is connected to the PC and continually emits LoRa signals through the antenna according to a predetermined time interval. The transmitter should support the transmission frequency band of 868 MHz. At the other end, 5-10 meters away from the

transmitter, a receiver down-converts the LoRa signals to the baseband and samples them. The LoRa packets composed of discrete I/Q samples are engendered, and are processed by algorithms to obtain unaffected preamble parts. Afterwards, these preambles are represented as spectrograms. They are forwarded to the classification model that has been stored in IoT device to predict the identity of the transmitter based on the received spectrograms. A fingerprint library should be prepared in advance for comparing RF features. The identification results are displayed on screen.

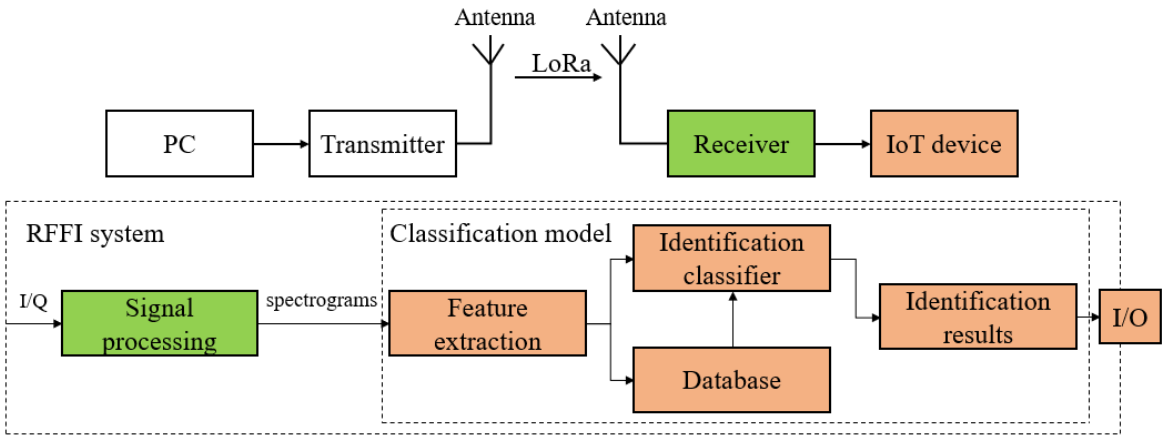


Figure 9. Simulated RFFI system

5.2 Algorithm Design

The signal transmitting algorithm is to be implemented in PC to control the transmitter. It specifies the configuration parameters and transmission data of LoRa modulation. The configuration parameters are aligned with data in Table 2. Transmission data can be either string or bytes. The receiver was programmed to cyclically read samples until the number of samples have reached the predefined sampling length. Normally, the modulus of LoRa I/Q sample is much greater than which of noise I/Q sample on account of high SNR. The RFFI system will first roughly filter out I/Q samples whose modulus is less than a threshold. The threshold can be randomly set as long as it is greater than any modulus of noise I/Q sample. Signal processing algorithm will then be executed on the remaining samples to acquire intact preambles. TS algorithms are adopted to extract LoRa preambles, and CFO algorithms are adopted to compensate the frequency offsets. Preambles are converted from digital time-domain signals to spectrograms in the time-frequency domain. The figures and pixel matrices of the spectrograms are saved in two separate folders. Users can access the figures to check the

pattern of spectrograms, while classification model can access the pixel matrices to learn RF features of the transmitter. In case that no preamble is received, instead of continuing to the inference stage, RFFI system needs to remind users to inspect any transmitting problem. In addition, a graphical user interface should be devised to suffice front-end requirements. It is necessary for users to input some basic parameters of sampling, such as sampling frequency, sampling length and sampling times. Push buttons are placed for user interaction. Meanwhile, the RFFI application is required to prevent users from inputting invalid parameters. For example, if a letter is entered into the line editor, the system should recover from exceptions and pop up a window to warn the users. A robust application should be developed to capture errors and exceptions and report them to the users. Furthermore, users can open the file explorer to select favourite classification model. Final inference result is outputted to be the label of the transmitter. The working process of the designed RFFI system is shown in Figure 10.

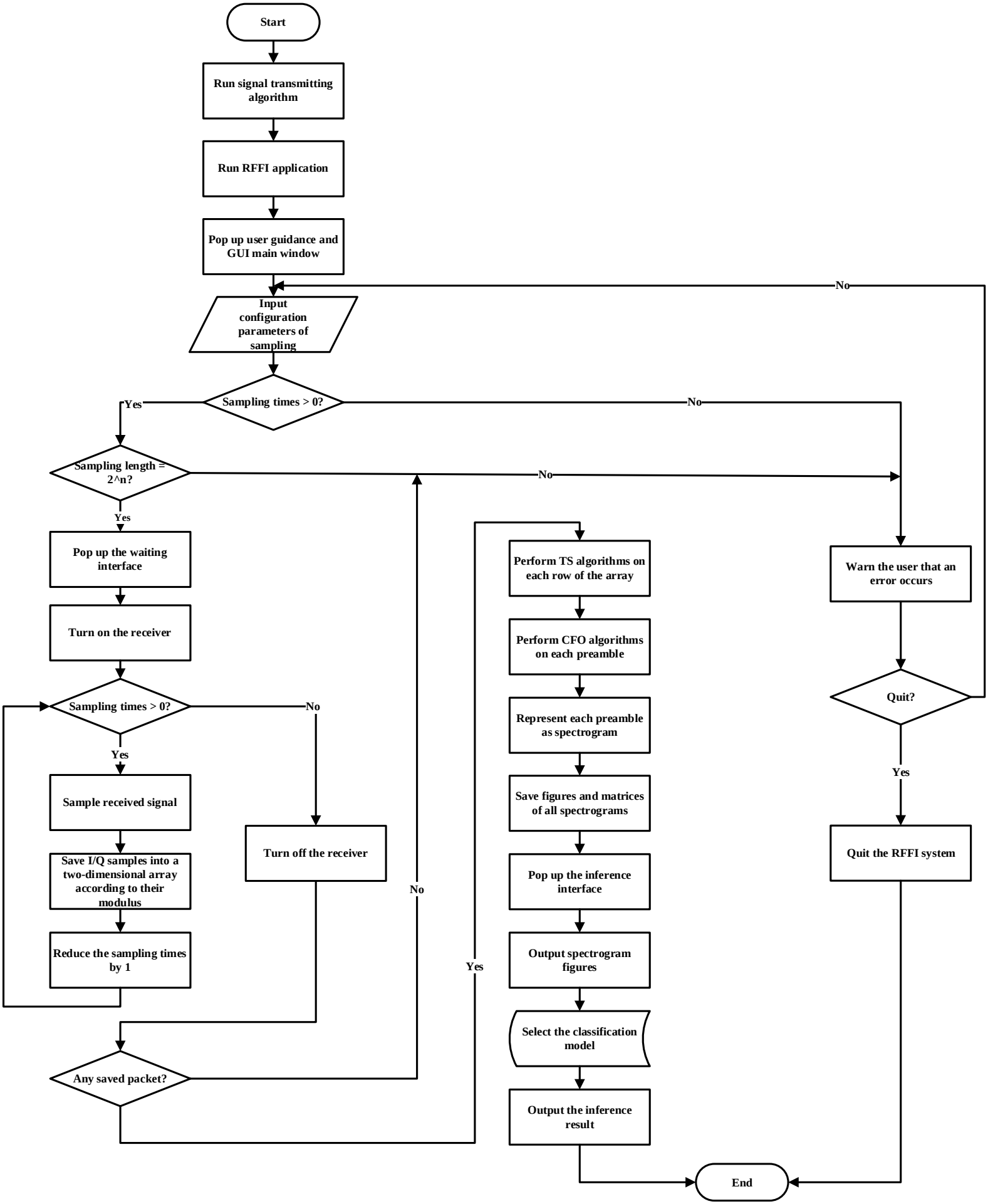


Figure 10. Whole flow chart of RFFI system

5.3 Neural Network Architecture

CNN is recommended to be the classification model of the RFFI system due to its powerful graphics cognitive ability. What is more, CNN itself is an assemblage of feature extraction, database and identification classifier. Totally, three NN models were designed, as displayed in Figure 12. Convolution layers are employed to extract features. Batch normalization layer is employed followed by convolution layer to normalize the inputs of each layer, reducing the internal covariate shift. The rectified linear activation layer (ReLU) is adopted after batch normalization. ReLU function turns all negative values in pixel matrix to zero and remains non-negative values unchanged. It has been proved that ReLU enables NNs to converge more quickly and compute more efficiently than other activation functions like sigmoid and tanh [31]. Max pooling layers are used to reduce the resolution of feature maps. The pooling kernel abandons all other elements in each feature map except for the maximum element, as the greatest value holds the most important features. Utilization of pooling layers can decrease number of parameters and inhibit overfitting. Following the feature extraction layers, dropout layer randomly deactivates neurons to avoid overfitting. Fully connected layer is exploited to classify images into different categories. Leaky ReLU is a special type of ReLU that has a definable slope in the negative domain. Ultimately, SoftMax layer maps outputs to a probabilistic distribution. Respective mathematical expression and function graph of activation functions are depicted in Figure 11.

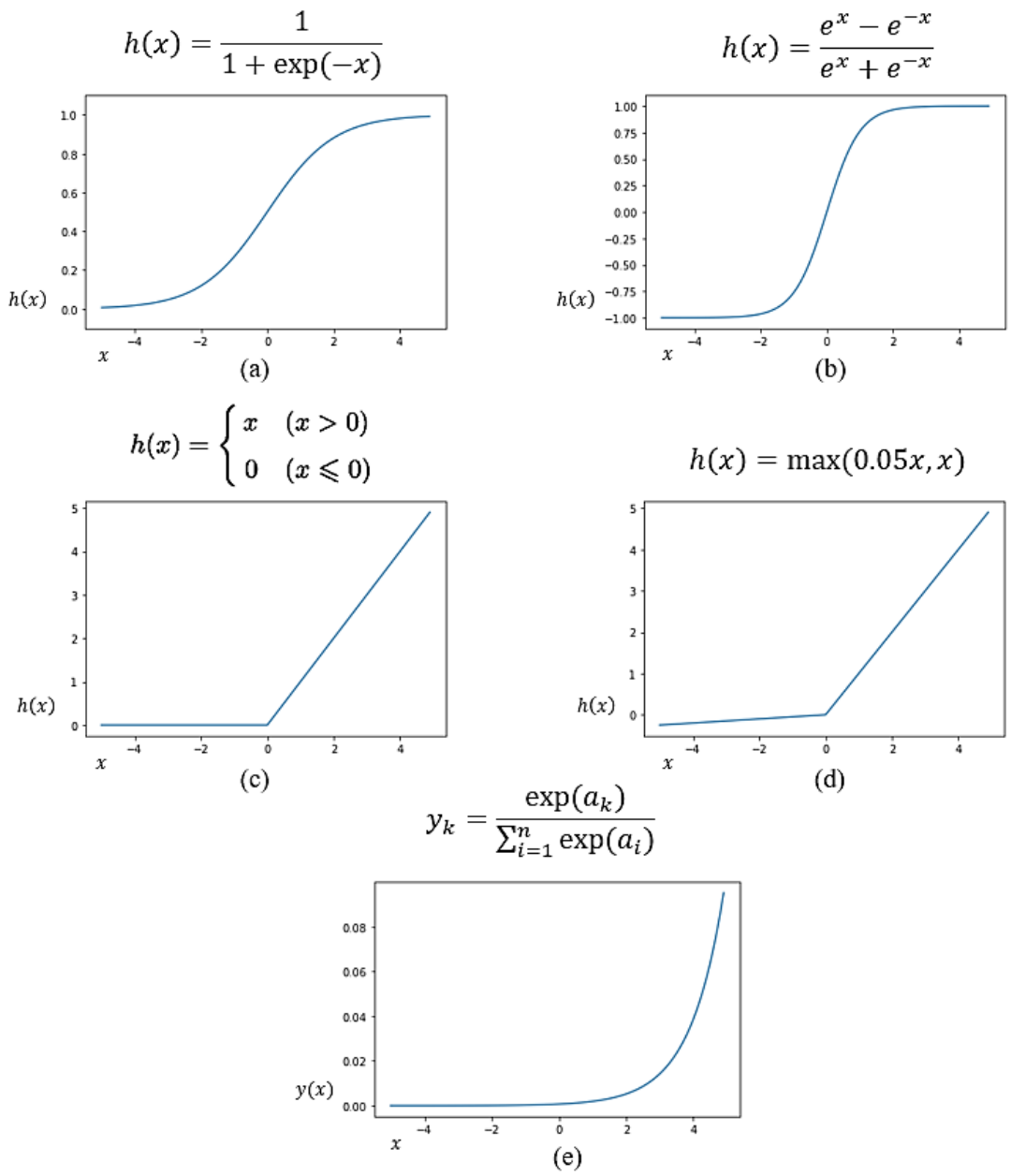


Figure 11. Activation functions. (a) Sigmoid. (b) Tanh. (c) ReLU. (d) Leaky ReLU. (e) SoftMax.

CNN models (a) and (b) were designed for spectrogram inputs. The first CNN model (a) owns fewer convolution layers compared with model (b), which means it possesses fewer learnable parameters but weaker feature extraction ability. Size of convolution kernels is 3x3 to ensure having a centre pixel. Number of kernels is integer power of 2 for supporting the parallel computing of CPU and GPU. Parameters were determined based on the rule of thumb. Both models would be tested to contrast classification accuracy. Classification model (c) is called multilayer perceptron (MLP), which is almost entirely composed of fully connected layers. It

was designed to investigate the ability difference between MLP and CNN on learning RF spectrograms.

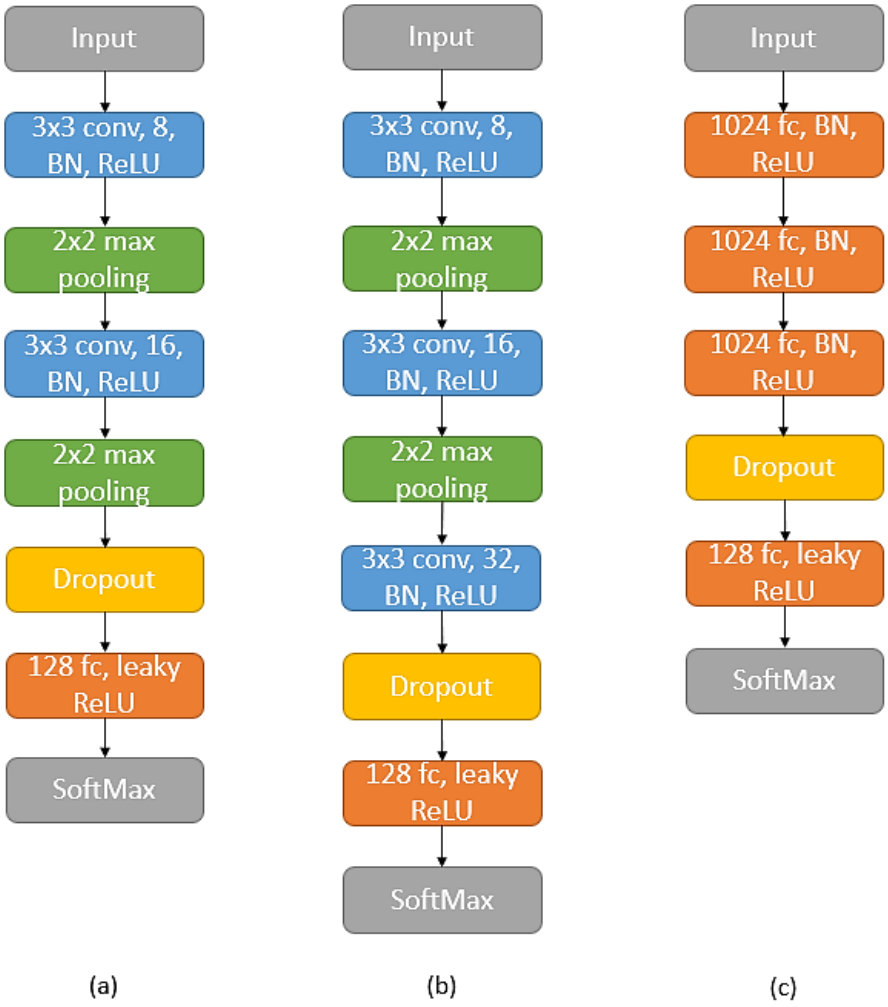


Figure 12. Designed NN architectures. (a) CNN with two convolution layers. (b) CNN with three convolution layers. (c) MLP with four fully connected layers.

6 Experimental Method

6.1 Apparatus Selection

As mentioned in the previous section, the indispensable RF development equipment for the final project contains a signal transmitter, a signal receiver, a PC and a low-power IoT device. PC had been already prepared. It is embedded with an Intel i7-9750H processor, which is powerful enough to do deep learning computations. The transmitter was decided to be Pycom LoPy4 development board. LoPy4 is a compact but powerful transceiver that supports LoRa communication. MicroPython scripts can be written to control LoPy4 devices. In the project, Atom coding platform was installed in PC to be the text editor of MicroPython. It features on a succinct and intuitive graphical user interface. The signal transmitting algorithm referred to the official website of Pycom [34]. Basic configuration parameters of LoRa modulation were set according to Table 2. The number of symbols in preamble was eight. The transmitter was programmed to transmit one byte data once after each 0.1 second. For the purpose of exploring the learnability of the RFFI system on different RF fingerprints, five LoPy4 boards were bought. They were all labelled with numbers from 1 to 5. An antenna was connected to each LoPy4 board for transforming analog current pulses into electromagnetic waves. Expansion board was assembled for connecting to the PC. Figure 13 shows the overview of the experimental transmitter.

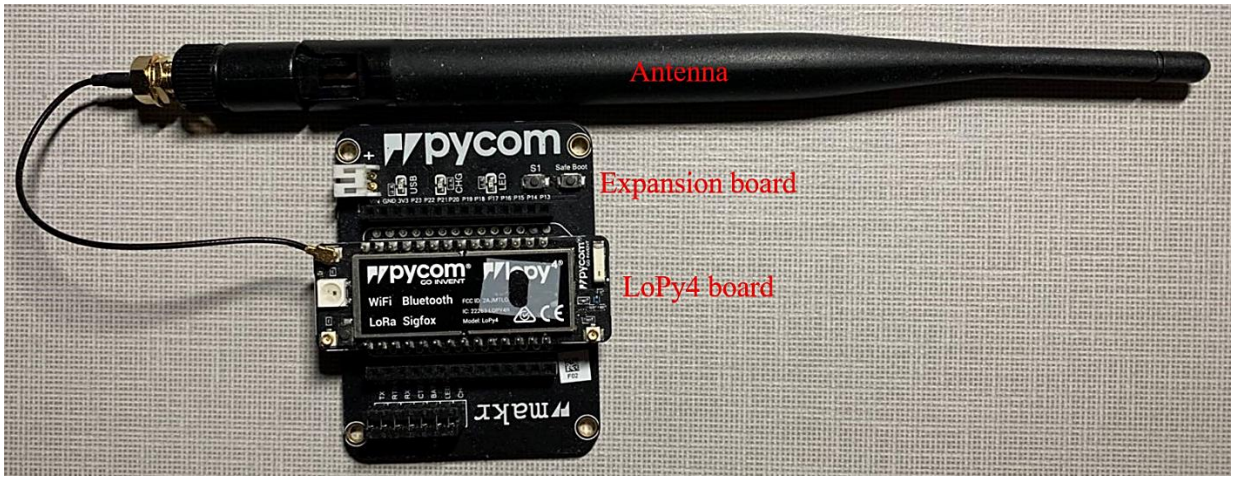


Figure 13. LoPy4 development kits for experiment

Realtek RTL2838U (RTL-SDR) was selected to be the signal receiver. It is a low-priced software defined radio equipped with RTL2838U chipset, stemming from DVB-T TV tuner

dongle. The electrical components of RTL-SDR, such as demodulator and tuner, are implemented by means of software instead of dedicated hardwired circuits. The receiver can tune into signal frequencies from 24MHz to 1850MHz. Figure 14 presents the internal structure of RTL-SDR. Mixer is embedded to down-convert LoRa signals from transmission band to the baseband. ADC samples and digitizes baseband signals to produce discrete-time and discrete-amplitude signals. The sampling frequency was 1MHz, which was much greater than twice the highest frequency (125 kHz), to satisfy Nyquist theorem. DSP mathematically manipulates inputs to generate I/Q samples. It was known that LoPy4 boards could receive LoRa signals as well. However, LoPy4 was not used as the receiver because it would introduce its own physical layer features, thereby, polluting original RF fingerprint of the transmitter. Besides, LoPy4 kits are more expensive than an SDR, which is conflicting with the project objective.

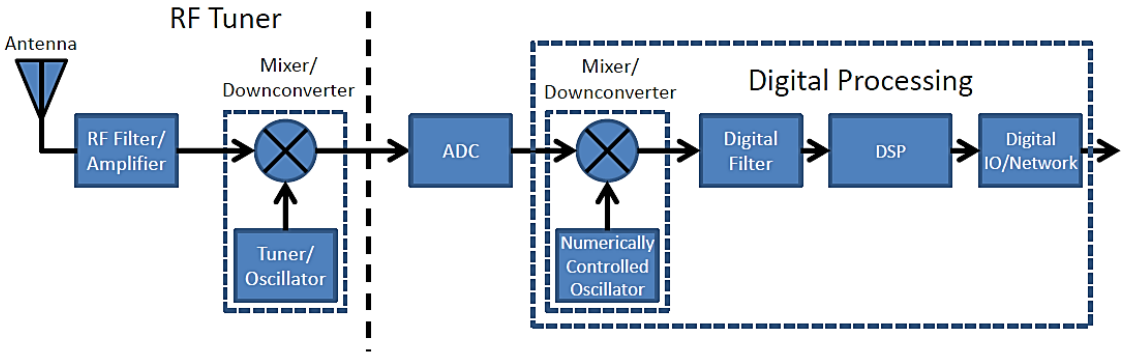


Figure 14. Demodulation process of RTL-SDR (taken from [35])

6.2 Algorithm Implementation

Signal processing algorithms and NN models were implemented by Python. The mathematical principles of signal processing algorithms are ground on the equations proposed in the Theory section. TensorFlow and Keras libraries were imported in scripts to create the designed NN architectures. They have ready-made layer functions that are convenient to invoke. After the NN models in Figure 11 had been created, 5500 packets were continuously gathered from five LoPy4 transmitters, of which 1100 packets from each transmitter. Each packet was labelled with a number that equalled to its source transmitter. The CFO shift of each packet was recorded in a database. 4500 packets (900 packets from each transmitter) were randomly selected as the training set. 500 packets (100 packets from each transmitter) were the validation

set to adjust the hyperparameters of the model and make a preliminary assessment of the learning ability. The rest 500 packets were the test set.

NN models were trained in PC with hyperparameters listed in Table 3. The inputted spectrograms were generated to have 256 pixels in height and 63 pixels in width, with one channel. Adaptive moment estimation (Adam) was selected to be the optimizer. The initial learning rate was set small in order to prevent weights from changing too much, resulting in the gradient descent overshooting the optimal convergence range. Additionally, model with the initial learning rate of 0.0003 costs less training time. As is illustrated in Figure 15, model performances with different initial learning rates were tested in the experiment. NNs with an initial learning rate of 0.0003 converges to the local minimum faster than other learning rates. Conversely, setting too low initial learning rates would lead NN to be deprived of learning ability and retard the training process.

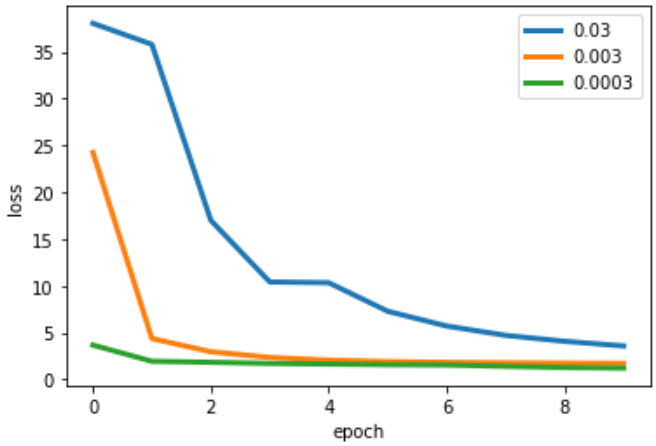


Figure 15. Training loss in 10 epochs of three learning rates

To progressively approaching to the plateau point, after every 10 epochs, if the validation loss stopped improving, the learning rate would decrease with a drop rate of 0.3. Loss and accuracy of training and validation in 60 epochs were depicted to evaluate the generalization ability of each NN model on RF fingerprints. It was anticipated that the test accuracy reached as high as possible, best above 90% but avoiding overfitting. All trained NN models were saved in PC afterwards.

Table 3. Training hyperparameters of NNs

Parameter	**Value**
Input Shape	(256,63,1)
L2 Regularization Factor	0.0001
Initial Learning Rate (LR)	0.0003
Dropout Rate	0.5
Negative Slope Coefficient	0.05
LR Drop Rate	0.3
LR Drop Period	10 epochs
Batch Size	32
Total Epochs	60

Finally, TensorFlow Lite was applied to optimize the post-trained NN models. It helps developers run TensorFlow models on IoT devices, supporting on-device machine learning inference. Optimized models can have diminutive binary file size. The compression method relies on the post-training quantization that appropriately reduces the floating-point number of the parameters without affecting the model accuracy. Detailed realization steps can be viewed in Figure 16. After converting NN models into TensorFlow Lite format, the compressed NN models were deployed to IoT device. Raspberry Pi 4 (RPi) was bought as the low-power IoT device. RPi is a microcomputer, whose operating system is based on Linux. In the project, an AC/DC adaptor with an output of 5V/3A was used to power the RPi. RTL-SDR was connected to RPi through USB. GUI application was developed by PyQt5 and run in RPi subsequently for inference. Five LoPy4 devices were tested sequentially. To determine the success of the RFFI system, any inference result should be consistent with the label of the tested transmitter. contain

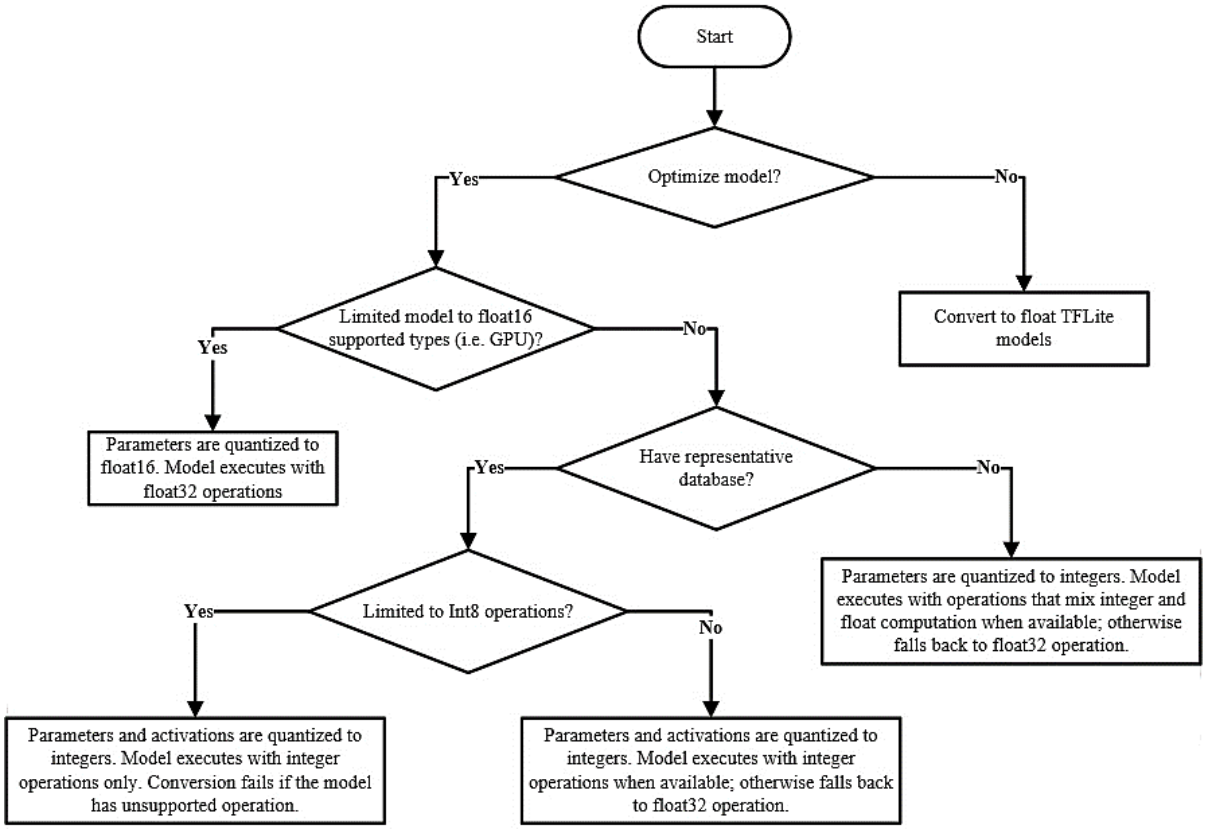


Figure 16. The post-training quantization of TensorFlow Lite (taken from [36])

7 Results and Calculations

Since the final project has been carried out for 17 weeks, many critical achievements were made.

7.1 LoRa Preamble Capture

Firstly, the TS algorithms have been implemented in Python script. A blended packet, in which the 3000th to 7000th samples were LoRa signal, others being noise, was generated to test the coarse TS algorithm. The result is delineated in Figure 17. It can be observed that the M values of first 0th to 2000th and last 7000th to 8000th samples are very low, almost approaching to 0.0. There shows a significant increase in M values from the 2000th to 3000th samples. For the LoRa signal part, the calculated M values are relatively high, reaching approximately 1.0. Apparently, equation (3) distributes highly differentiated values to noise and LoRa signal. Therefore, it is feasible to set a threshold for M value to coarsely estimate the start of a LoRa

signal. The threshold can take random value, as subsequent fine TS algorithm can accurately locate the starting point. However, it is recommended to set threshold more than 0.9 to reduce the computational burden of fine TS algorithm. In the experiment, the threshold was confirmed to be 0.93. Tested best threshold range for M value is from 0.90 to 0.95. Values above 0.95 may sometimes filter out important samples of the LoRa signal.

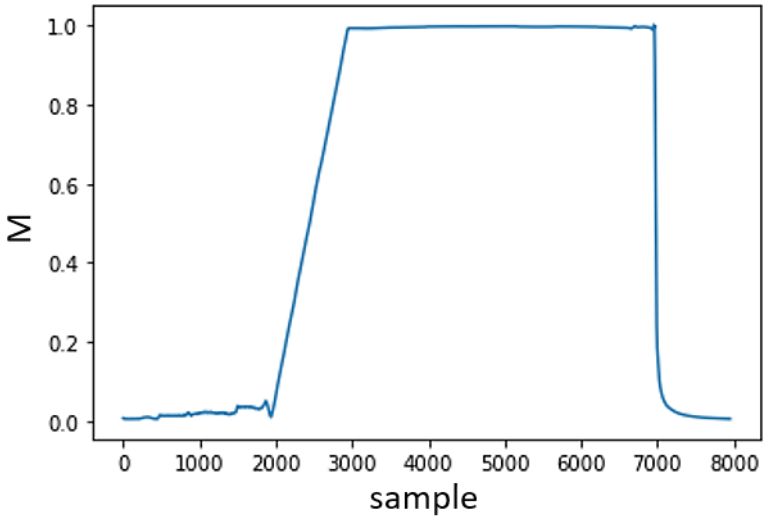


Figure 17. M values of 8000 samples

The fine TS algorithm was implemented to locate the exact starting point. Figure 18 presents the extracted packet of baseband LoRa preamble. There are eight symbols in the preamble. The experimental symbol duration is 1.024ms. The frequency variation of each symbol increases linearly with time. Total sequence length of preamble is 8192. Obviously, there exists a frequency offset that shifts the average frequency of preamble to a negative number. These results are in line with the theoretical expectations.

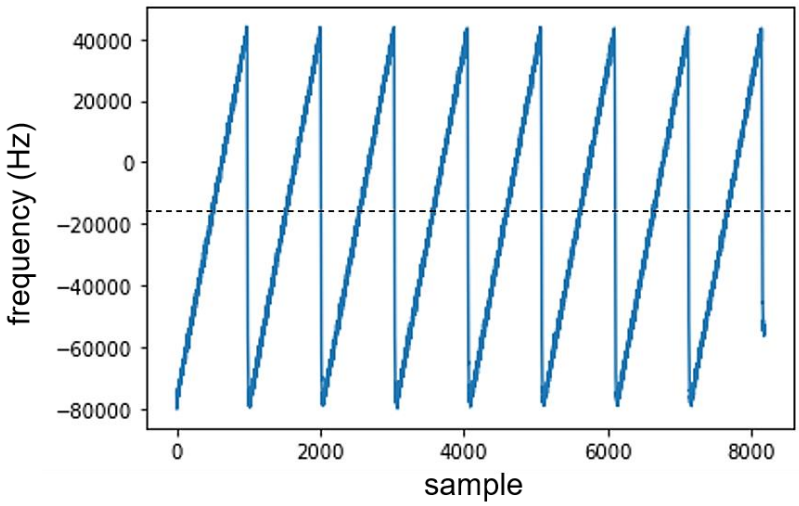


Figure 18. Frequency swings of baseband preamble after TS

Secondly, CFO estimation and compensation algorithms were realized by Python to compensate the frequency offset. The experiment has consecutively measured carrier frequency offsets of the first transmitter in 20 minutes, as elucidated in Figure 19, where time interval between two adjacent packets is 2 seconds. It can be viewed that the CFO variation decreases rapidly in the first 5 minutes, then it gradually slows down. After approximate 16 minutes, CFO tends to be stable. These results are understandable because temperature gradually increases, and will ultimately maintain at a maximum limitation. Meanwhile, it was investigated that both LoPy4 and SDR would be influenced by temperature and contributed drifts to the preamble frequency. CFO variations of other four transmitters were also measured and showed similar phenomena. The CFO variation range of each device is different.

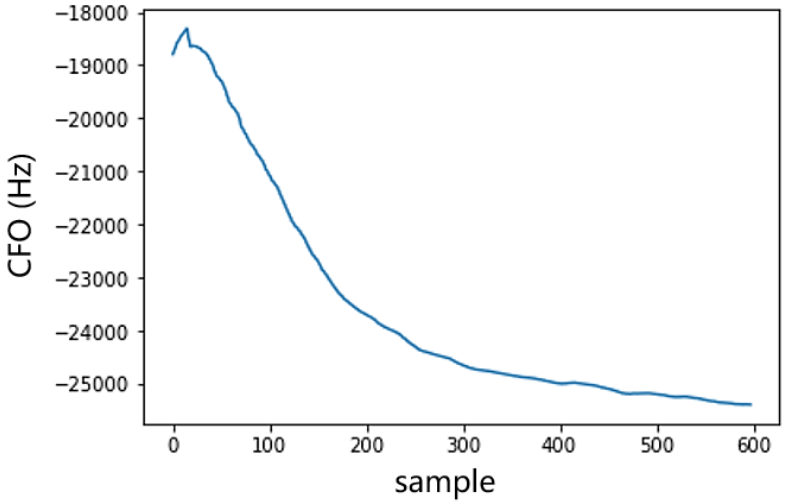


Figure 19. Estimated CFO variation of 600 samples in 20 minutes

The compensated baseband LoRa preamble can be viewed in Figure 20. The average frequency of preamble is compensated to be approximately zero. The frequency baseband was measured to be 125 kHz. Consequently, it is sufficient to prove the correctness of the CFO algorithms.

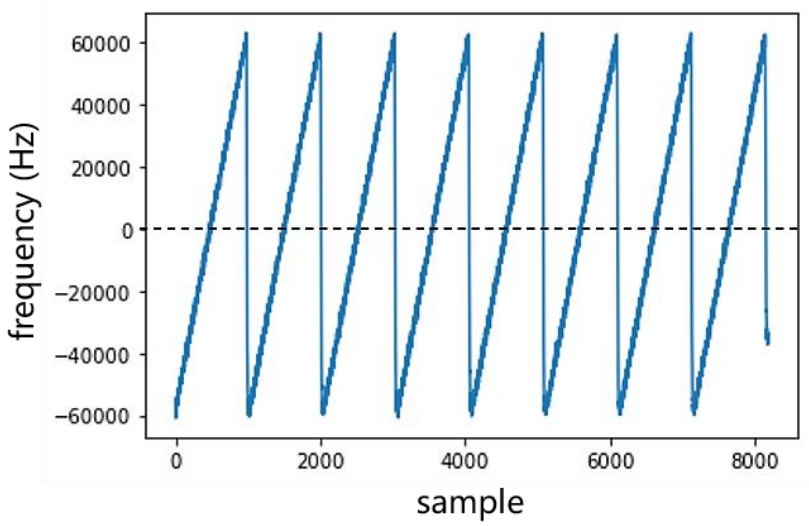


Figure 20. Compensated frequency swings of baseband preamble

Table 4 summarizes the information of the LoRa signals transmitted in the experiment.

Table 4. Measured data of LoRa signal

Parameter	**Value**
Explicit header	Enabled
CRC	Enabled
Chip Rate	125000chips/s
Data Rate	5.47kbps
Preamble Length	8 symbols
Preamble Duration	12.544ms
Symbol Time	1.024ms
Symbols in Frame	13
Payload Length	1 byte
Payload Duration	13.312ms
Time on Air	25.856ms

Duty Cycle	One message every 3 seconds

7.2 Neural Network Performance

Thirdly, the designed NN models in Figure 12 were trained with 4500 packets and validated with 500 packets. The accuracy and loss in 60 epochs of training and validation of three models were shown in Figure 21. With the increasement of epoch, each model manifests a tendency of accuracy rising and loss dropping. It can be attested that received spectrograms truly contain learnable RF fingerprints. Different models have disparate performance on the training and validation sets.

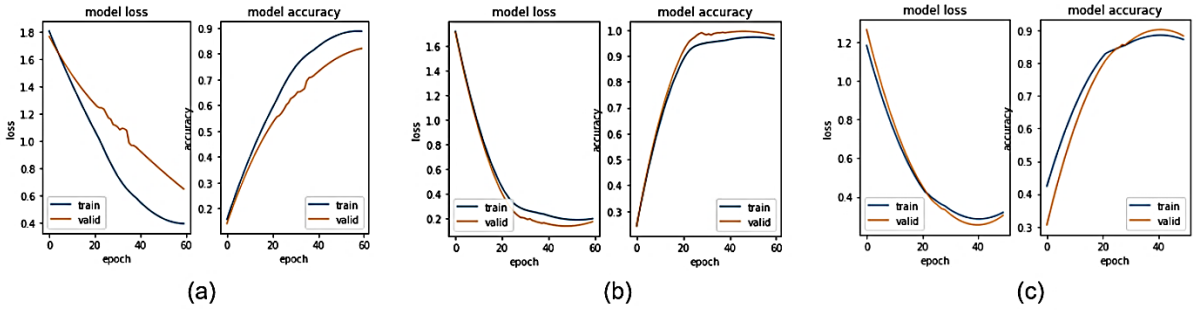


Figure 21. Generalization performance of NN model (a), (b) and (c)

500 packets were fed into trained NN models for testing. The attributes and test results of each NN are documented in Table 5. Conclusions can be drawn that model (a) has fewer parameters and less training time than model (b), but the test accuracy of model (a) is much lower than which of model (b). Two convolution layers are not adequate to extract full features of spectrograms. CNN model with four convolution layers was tested likewise, whose test accuracy stabilised at approximate 95%. Increasing the number of convolution layers would not have a substantial impact on the test results. Therefore, three convolution layers are considered to be enough for classifying five LoPy4 devices. Model (c) is the MLP, which has much more parameters and training time than other models. The final test accuracy of MLP is 10.88% lower than which of CNN (b). As a result, in the project, MLP is inferior to CNN in classifying spectrograms. CNN model (b) is determined to be the perfect classification model for RFFI system, as it causes moderate computational complexity and training delay to the system and achieve the highest test accuracy.

Table 5. Test accuracy and loss, number of parameters, training time of different models

Model	Type of NN	Accuracy	Loss	Number of Parameters	Training Time (min)
(a)	CNN	78.41%	1.1317	1,968,197	22
(b)	CNN	95.60%	0.2925	3,939,045	29
(c)	MLP	84.72%	0.6290	18,759,429	38

The confusion matrix of the test results of CNN model (b) is demonstrated in Figure 22. Usage of confusion matrix can intuitively observe the model performance. The spectrograms on the diagonal are predicted to be the left-most true labels. Bottom predicted labels of other spectrograms mismatch the true labels on the left. According to data in Figure 22, 478 spectrograms were predicted to be correct. Residual 22 spectrograms were predicted to be incorrect. The test results were gratifying as they have met the requirements of specification report.

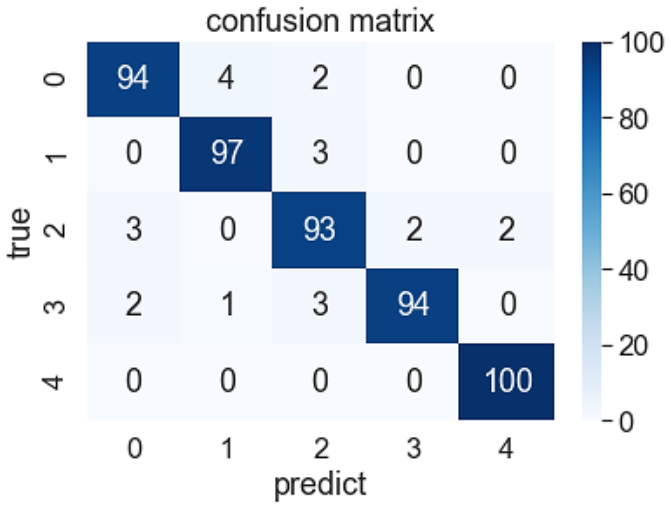


Figure 22. Prediction results of CNN model (b) on 500 spectrograms

To verify the validity of CFO algorithms, test packets were collected from the next three days, and respectively inputted to the trained CNN model (b). The confusion matrixes of classification results in each day were documented, as arranged in Figure 23. Evidently, test accuracies of three days stay steady at nearly 95%. It can be firmly convinced that impact of CFO drift on classification accuracy was eradicated.

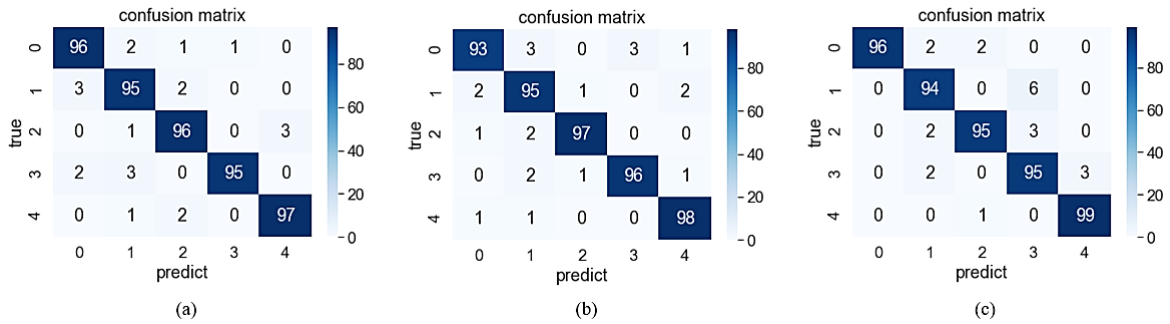


Figure 23. Experimental results. (a) Day II. (b) Day III. (c) Day IV

The trained CNN model was converted from h5 format to TensorFlow Lite format. File size compressed from 46224KB to 3853KB, which was 12 times smaller than the original size. The conversion model was copied to RPi for inference.

7.3 RFFI Application

Finally, GUI application for RFFI system was developed using PyQt5 in RPi. Figure 24 displays the designed user interface, where users can input basic settings of sampling and select suitable parameters for SDR. The debug system will censor whether invalid inputs exist whenever packet receiving button is clicked. If no exception occurs, connected SDR will then be automatically turned on and read signal packets based on the sampling settings.

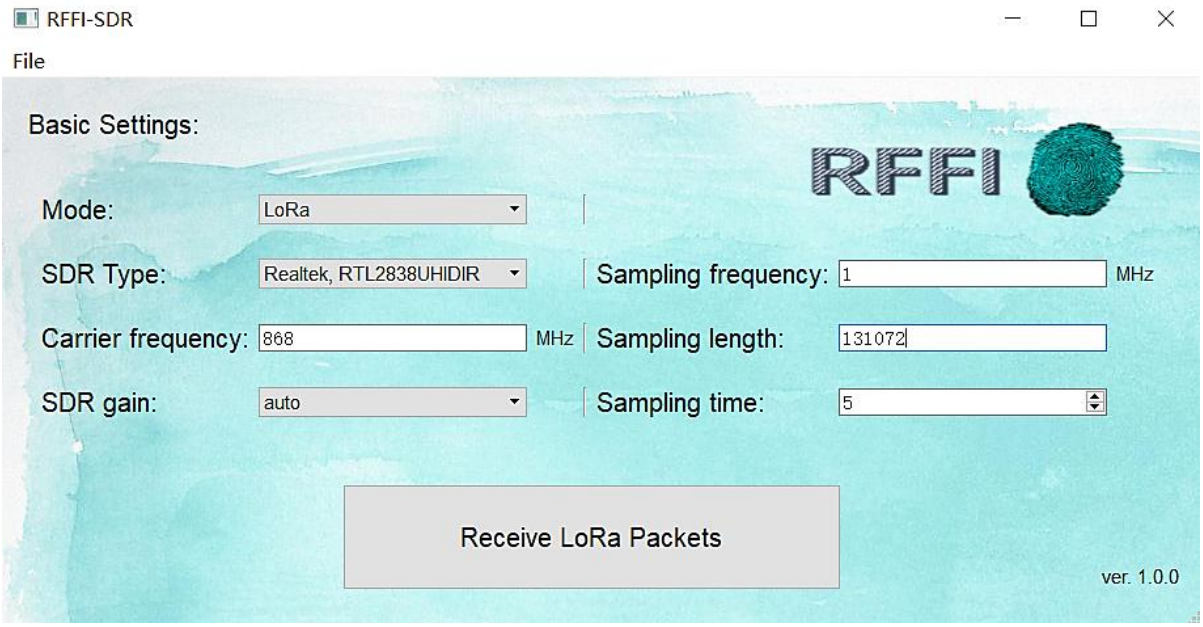


Figure 24. Main window layout of RFFI application

For the inference phase, as shown in Figure 25, users can inspect the pattern of each received spectrogram, and choose stored CNN model for inference. All received spectrograms will be inferred and the label with the most occurrences is considered the inference result, which is outputted in the bottom bar. It has been tested that the classification model can correctly identify five LoPy4 devices. The inference accuracy can reach 95% if 20 spectrograms are analysed. Data packets collected in different periods would not decline the inference accuracy.

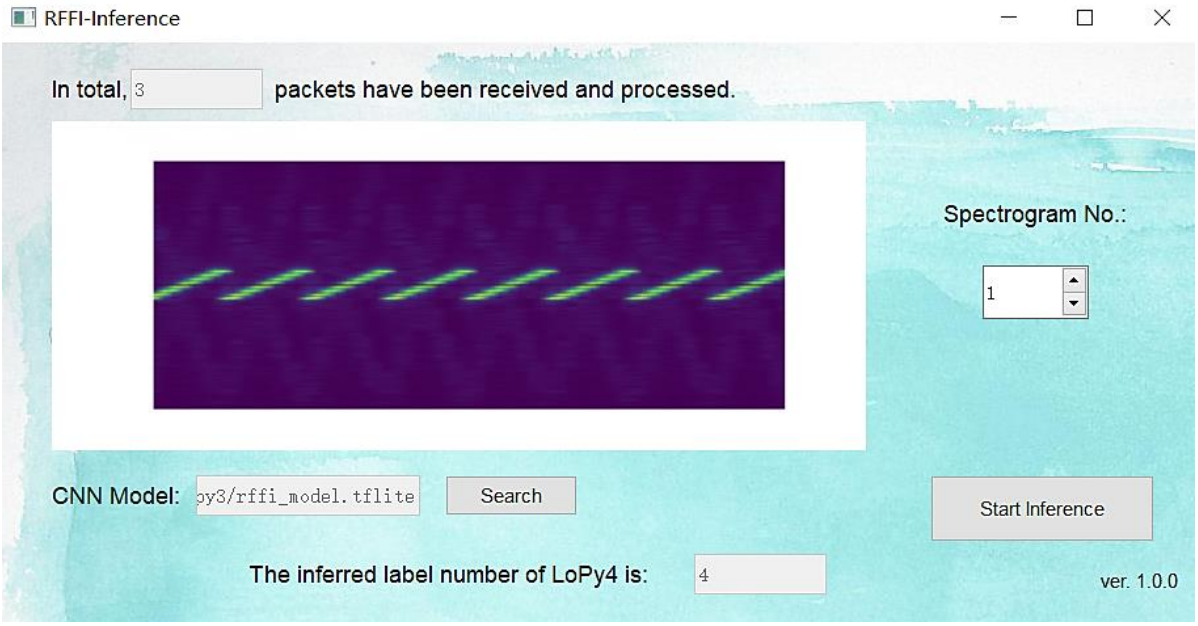


Figure 25. Interface layout for inference

8 Discussion

With the intensification of cyber threats, it is foreseeable that device authentication schemes utilizing software addresses are ineluctably subjected to identity forgery and theft. More recently, radio frequency fingerprint identification technique was proposed as a substituted authentication strategy with great potentialities. In the present work, a deep-learning based RFFI system was built as the preliminary prototype for LoRa device classification. Applying time synchronization and CFO compensation to procure preambles, and CNN to be the feature classifier, the methodology was confirmed to be practical and reliable. Results indicate that every experimental LoPy4 transmitter imposes a distinctive influence on frequency fluctuations. Regular pattern can be extracted from substantial spectrograms to constitute a

device fingerprint. The developed RFFI application was assessed to be efficacious in identifying wireless devices under certain prerequisites.

In comparison to previous literature [11], [37], exploited signal processing methods have exhibited two principal advantages. First of all, clear and unabridged LoRa preambles were extracted from composite signal. Moreover, the undesirable phenomenon, which emerged a significant degradation in classifier performance when test packets gathered on different days were fed for inference, was avoided. In addition to this, erstwhile studies [11], [14], [38] directly used I/Q samples as CNN inputs, while in the project, evidences show that representing I/Q samples as spectrograms is also feasible. Three NN architectures were designed and compared in accordance with respective classification performance to verify the rationality of other researches [8]-[10]. Groundbreakingly, a robust GUI application for RFFI system was developed, which provides a rudimentary template for future market applications.

Although the project has revealed serval important discoveries, there still exists some limitations and deficiencies. Firstly, wireless channel conditions such as multipath and shadow fading were not investigated meticulously. Experiments conducted from different locations demonstrated that, if keeping SDR in the same location but deploying LoRa devices in the adjacent room, the collected test packets only leaded classifier to achieve approximately 32% accuracy. The alteration of channel conditions can significantly impact the classification accuracy. Possible mitigation methods are to tentatively generate signal with artificial noise that enables classifier to adapt various channel conditions [37], or by manufacturing a FIR filter at the transmitter side to compensate channel distortion [39]. Secondly, it was experimentally found that changing SDR and antenna would invalidate the classifier to some extent. Reason was speculated that the physical characteristics of SDR and antenna have been learnt by CNN model as well. Envisaged solutions include developing additional algorithms to thoroughly eliminate these characteristics or applying transfer learning to the CNN model. Thirdly, situations of unconfident predictions have not been considered. If two transmitters have very similar characteristics, merely treating the greatest possibility as the predicted result is not rigorous. A proposed method is to exclude those outputted probabilities that are obviously impossible according to the CFO uniqueness [8]. Furthermore, all experiments were carried out with the condition that the antenna of the receiver was parallel to which of the transmitter. When two antennas were placed perpendicular to each other, the classification accuracy diminished strikingly. In addition, if two or more LoPy4 devices are simultaneously

transmitting LoRa signals. A farraginous packet will be received by the RFFI system, which can perplex the classifier. Incontrovertibly, a novel algorithm is requisite to be found in the future to address these problems. Moreover, low bad packet rate is an important indicator to warrant a high-accuracy classifier. In the experiments, fragmentary preambles can be received and be mixed into the training set. These bad preambles will degenerate the generalization degree of NN models. Preferably, the bad packet rate should be controlled below 0.2%. Recommended improvement is that the receiver performs cross-correlation algorithm on both ends of the preamble to inspect whether it is standards-compliant. Finally, due to the scalability of IoT, RFFI system should expeditiously acquire the ability to classify newly-joined devices. Long-term continuous training on spectrograms is not realistic. Thanks to Siamese neural network, it is constituted by two associated neural networks that can measure the similarity of two inputs [40]. With merely a small number of training samples, Siamese network is able to correctly determine whether an unknown spectrogram belongs to a category based on the contrastive loss. As a consequence, on the basis of the original specifications having been completed, some further experiments should be conducted.

In summary, notwithstanding the mentioned limitations, the research has made some preliminary achievements in the field of RFFI technique. Scientifically, the existence of RF fingerprints was verified. A RFFI system was developed on five LoPy4 devices, with approximate 95% inference accuracy. It is undoubted that the experimental results and developed application can be used as a transition for succeeding researches. The future investigations will focus on ameliorating algorithms to sustain high accuracy under various external conditions, and expanding the classification scope using more LoPy4 devices.

9 Conclusions

In conclusion, a deep-learning based RFFI system for classifying LoRa devices was actualized successfully. Five LoPy4 transmitters were deployed to emitting LoRa signals. RTL-SDR was the receiver and Raspberry Pi was exploited as the IoT device. The report has comprehensively expounded potential shortcomings of software addresses applied in conventional device authentication schemes. RFFI technique was introduced detailly as a practicable scheme to improve IoT security of limited computational ability devices. The project utilized TS algorithms to ascertain the start of a LoRa symbol. Successive I/Q samples with a sequence

length of 8192 were extracted as the LoRa data packet. Furthermore, it had been observed that received preamble was frequency shifted and the CFO offset varies incessantly during signal acquisition period. CFO estimation and compensation algorithms were employed to compensate the frequency drift for eliminating performance degradation. Experiments conducted on other days shew the stabilization of classification accuracy. 1100 packets were collected from each transmitter and represented as spectrograms. Three NN architectures were constructed to be the classification models of RFFI system. CNN with three convolution layers was proved to outperform other two NN models in classifying signal spectrograms. It was experimentally validated that radio frequency spectrograms of transmitters contain learnable features that can be extracted by neural networks. A GUI application was developed for implementing front-end operations. The robustness of application was guaranteed by debugging any possible failure. The proposed RFFI system can finally achieve 95.60% accuracy on classifying five LoRa devices. On the other hand, some scientific contributions and potential limitations were discussed. Alternative solutions were suggested to mature the RFFI technique in the future.

References

[1] L. Coetzee and J. Eksteen, "The Internet of Things - promise for the future? An introduction", in *2011 IST-Africa Conference Proceedings*, Gaborone, Botswana, 2011, pp. 1-9.

[2] S. Chung, J. H. Kim and Y. Kim, "Pragmatic approach using OAuth mechanism for IoT device authorization in cloud," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2018, pp. 1-4, DOI: 10.1109/ICACCCN.2018.8748856.

[3] P. Stokes, "6 Internet of Things (IoT) security technologies on the contemporary market", Sep, 2018. [Online]. Available: https://medium.datadriveninvestor.com/6-internet-of-things-iot-security-technologies-on-the-contemporary-market-8b302d04d1d0. (Accessed Mar. 21, 2021)

[4] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[5] Q. Xu, R. Zheng, W. Saad and Z. Han, "Device Fingerprinting in Wireless Networks: Challenges and Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94-104, Firstquarter 2016, DOI: 10.1109/COMST.2015.2476338.

[6] A. Sivanathan et al., "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics," in *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745-1759, 1 Aug. 2019, DOI: 10.1109/TMC.2018.2866249.

[7] H. Liu, Y. Wang, J. Liu, J. Yang, Y. Chen and H. V. Poor, "Authenticating Users Through Fine-Grained Channel Information," in *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 251-264, 1 Feb. 2018, DOI: 10.1109/TMC.2017.2718540.

[8] G. Shen, J. Zhang, A. Marshall, L. Peng, X. Wang, “Radio Frequency Fingerprint Identification for LoRa Using Deep Learning,” PhD dissertation, Dept. Electronic & Elec. Eng., Univ. of Liverpool, Liverpool, UK, 2020.

[9] S. Gopalakrishnan, M. Cekic and U. Madhow, "Robust Wireless Fingerprinting via Complex-Valued Neural Networks", 2019.

[10] R. Das, A. Gadre, S. Zhang, S. Kumar and J. M. Moura, "A Deep Learning Approach to IoT Authentication", *Proc. of the IEEE International Conference on Communications (ICC)*, pp. 1-6, 2018.

[11] A. Al-Shawabka et al., "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, 2020, pp. 646-655, DOI: 10.1109/INFOCOM41043.2020.9155259

[12] S. D. Andrews, “Extensions to radio frequency fingerprinting,” Ph.D. dissertation, Virginia Tech, 2019.

[13] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, “Accurate and efficient wireless device fingerprinting using channel state information,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1700–1708.

[14] S. Riyaz, K. Sankhe, S. Ioannidis and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," in *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146-152, Sept. 2018, DOI: 10.1109/MCOM.2018.1800153.

[15] F. Meneghello, M. Calore, D. Zucchetto, M. Polese and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182-8201, Oct. 2019, DOI: 10.1109/JIOT.2019.2935189.

[16] F. Meneghello, M. Calore, D. Zucchetto, M. Polese and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182-8201, Oct. 2019, DOI: 10.1109/JIOT.2019.2935189.

[17] IoT.Business.News, “Global IoT device connections to reach 11.7 billion in 2020, surpassing non-IoT devices for the first time”, Nov, 2020. [Online]. Available: https://iotbusinessnews.com/2020/11/20/03121-global-iot-device-connections-to-reach-11-7-billion-in-2020-surpassing-non-iot-devices-for-the-first-time/ (Accessed Mar. 25, 2021)

[18] Snigdha, “Everything You Should Know About the IoT Before Hopping On-board,” Mar. 2021. [Online]. Available: https://www.appypie.com/everything-you-should-know-about-internet-of-things (Accessed Mar. 25, 2021)

[19] ReportLinker, “The IoT in manufacturing market size is projected to grow from USD 33.2 billion in 2020 to USD 53.8 billion by 2025, at a Compound Annual Growth Rate (CAGR) of 10.1%,” Dec. 2020. [Online]. Available: https://www.globenewswire.com/news-release/2020/12/22/2149124/0/en/The-Iot-in-manufacturing-market-size-is-projected-to-grow-from-USD-33-2-billion-in-2020-to-USD-53-8-billion-by-2025-at-a-Compound-Annual-Growth-Rate-CAGR-of-10-1.html (Accessed Mar. 25, 2021)

[20] M. Smith and E. Lostri, “The Hidden Costs of Cybercrime,” Dec. 2020. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf (Accessed Mar. 25, 2021)

[21] Symantec Norton Department, “NortonLifeLock Cyber Safety Insights Report,” Jan. 2021. [Online]. Available: https://uk.norton.com/nortonlifelock-cyber-safety-report (Accessed Mar. 25, 2021)

[22] J. Yu et al., "Radio Frequency Fingerprint Identification Based on Denoising Autoencoders," *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Barcelona, Spain, 2019, pp. 1-6, DOI: 10.1109/WiMOB.2019.8923325.

[23] IPA, “Business Outline: Creating a Secure and Reliable IT-based Society,” 2012. [Online]. Available: https://www.ipa.go.jp/english/about/outline/security/01.html (Accessed Mar. 26, 2021)

[24] L. Irwin, “UK cyber-crime rate has doubled in the past five years,” July, 2020. [Online]. Available: https://www.itgovernance.co.uk/blog/uk-cyber-crime-rate-has-doubled-in-the-past-five-years (Accessed Mar. 27, 2021)

[25] Sensors – Energy Dynamics, “WE ARE LORAWAN SOFTWARE DEVELOPERS,” 2021. [online]. Available: https://www.energy-dynamics.com.au/lorawan-sensors/ (accessed 27th March 2021)

[26] Semtech, “LoRa™ Modulation Basics”, May, 2015. [Online]. Available: https://web.archive.org/web/20190718200516/https://www.semtech.com/uploads/documents/an1200.22.pdf (Accessed Mar. 27, 2021)

[27] T. Bouguera, J. F. Diouris, J. J. Chaillout, R. Jaouadi, A. Guillaume, “Energy consumption model for sensor nodes based on LoRa and LoRaWAN,” *Sensors 18(7):2104,* France, Jun. 2018, DOI: 10.3390/s18072104

[28] V.S. Shridhar, “Tata Communications’ Countrywide Internet of Things Will Manage the Chaos in India’s Booming Cities,” Jan. 2019. [Online]. Available: https://spectrum.ieee.org/telecom/internet/tata-communications-countrywide-internet-of-things-will-manage-the-chaos-in-indias-booming-cities (Accessed Mar. 27, 2021)

[29] T. M. Schmidl and D. C. Cox, “Robust frequency and timing synchronization for OFDM,” *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, 1997

[30] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, “A multi-channel software decoder for the LoRa modulation scheme,” in *Proc. Int. Conf. Internet Things, Big Data Secur. (IoTBDS),* Mar. 2018, pp. 41–51.

[31] K.E. Swapna, “Convolutional Neural Network | Deep Learning,” Aug. 2020. [Online]. Available: https://developersbreach.com/convolution-neural-network-deep-learning/ (Accessed Mar. 28, 2021)

[32] G. Lou and H. Shi, "Face image recognition based on convolutional neural network," in *China Communications*, vol. 17, no. 2, pp. 117-124, Feb. 2020, DOI: 10.23919/JCC.2020.02.010.

[33] J. Vandoni, “Ensemble Methods for Pedestrian Detection in Dense Crowds,” Image Processing [eess.IV]. University of Paris-Saclay, Oct. 2019. English. NNT: 2019SACLS116.

[34] “Pycom go invent”, accessed Apr. 1, 2020. [Online]. Available: https://docs.pycom.io/firmwareapi/pycom/network/lora/ (Accessed Mar. 28, 2021)

[35] J. Banks, “High end software defined radio: getting closer to the ideal,” Dec. 2016. [Online]. Available: https://www.curtisswrightds.com/news/blog/high-end-software-defined-radio-getting-closer-to-the-ideal.html (Accessed Apr. 1, 2021)

[36] TensorFlow Lite, “Post-training quantization,” accessed Apr. 2, 2020. [Online]. Available: https://tensorflow.google.cn/lite/performance/post_training_quantization (Accessed Apr. 2, 2021)

[37] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelee, and B. Preneel, “Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning,” in *Proc. ACM Conf. Security Privacy Wireless Mobile Netw. (WiSec)*, Boston, MA, USA, Jul. 2017, pp. 58–63.

[38] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, “Deep learning for RF device fingerprinting in cognitive communication networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp.160–167, 2018.

[39] F. Restuccia, S. D’Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, et al., "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Finger-printing Algorithms", *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, pp. 51-60, 2019.

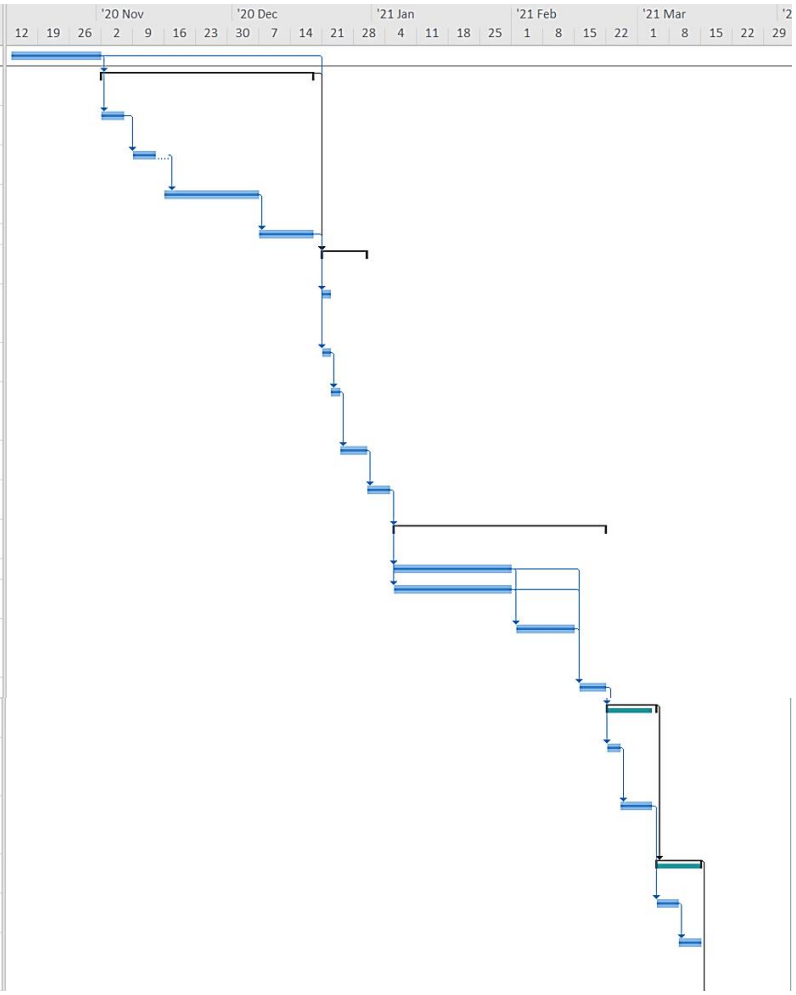
[40] L. Vizváry, D. Sopiak, M. Oravec and Z. Bukovčiková, "Image Quality Detection Using The Siamese Convolutional Neural Network," *2019 International Symposium ELMAR*, Zadar, Croatia, 2019, pp. 109-112, doi: 10.1109/ELMAR.2019.8918678.

A Appendix-1: Work Packages, Deliverables, and Milestones

	Task	Deliverable	Milestone
Workpackage-1	T-1. Literature Review	The related knowledge of LoRa, RFFI and CNN has been acquired.	Specification Report
Workpackage-2	T-2. LoPy4 Configuration	LoRa transmission is implemented in MicroPython script.	Completion of SDR-based Physical Layer of LoRa
	T-3. SDR Configuration	Time synchronization is implemented in Python script.	
		CFO estimation and compensation is implemented in Python script.	
	T-4. Data Collection	5500 data packets are collected from five LoPy4 boards.	
Workpackage-3	T-5. PC Software Installation	CNN model is implemented in Python script with TensorFlow and Keras libraries.	Completion of Convolution Neural Network Model
	T-6. CNN Structure Designing		
	T-7. TensorFlow/Keras Language Learning		
	T-8. CNN Algorithm Implementation		
Workpackage-4	T-9. Performance Evaluation	A high classification accuracy is reached.	CNN Model Training and Testing
Workpackage-5	T-10. RPi Software Installation	RPi can invoke trained CNN model and correctly infer source device of one RFF spectrogram.	Completion of TensorFlow Lite script
			Bench Inspection
	T-11. RPi TensorFlow Lite Implementation		Final Report Submission

B Appendix-2: GANTT Chart

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Read related literature**	14 days	2020/10/13	2020/11/2	
2		**Design an SDR-based LoRa communication system**	**35 days**	**2020/11/2**	**2020/12/18**	**1**
3		Start programming the LoPy4 using MicroPython	5 days	2020/11/2	2020/11/6	1
4		Receive signals through RTL-SDR	5 days	2020/11/9	2020/11/16	3
5		Implement the TS and CFO algorithms	15 days	2020/11/16	2020/12/7	4
6		Gather data packets	10 days	2020/12/7	2020/12/18	5
7		**Implement neural network on PC**	**8 days**	**2020/12/21**	**2020/12/30**	**2,1**
8		Install requisite software and extension libraries	2 days	2020/12/21	2020/12/22	6
9		Design the structure of neural network	2 days	2020/12/21	2020/12/22	6
10		Learn algorithms of TensorFlow and Keras libraries	2 days	2020/12/23	2020/12/24	9
11		Implement algorithms of CNN	4 days	2020/12/25	2020/12/30	10
12		**Evaluate the performance of RFFI**	3 days	2020/12/31	2021/1/5	11
13		**Make further improvements**	**33 days**	**2021/1/6**	**2021/2/22**	**12**
14		Optimize the algorithms	18 days	2021/1/6	2021/2/1	12
15		Correct potential mistakes	18 days	2021/1/6	2021/2/1	12
16		Investigate more architectures of CNN to increase accuracy	9 days	2021/2/2	2021/2/15	15,14
17		Test more LoPy4	4 days	2021/2/16	2021/2/22	16,15,14
18		**Invoke trained CNN model for inference**	**9 days**	**2021/2/23**	**2021/3/5**	**17**
19		Operate Linux system and install requisite software in RPi	3 days	2021/2/23	2021/2/25	17
20		Use TensorFlow Lite to invoke and optimize CNN model	5 days	2021/2/26	2021/3/5	19
21		**Design user interface for RFFI system in RPi**	**7 days**	**2021/3/6**	**2021/3/16**	**18**
22		Download pyqt5 library for GUI designing	3 days	2021/3/6	2021/3/10	20
23		Implement the GUI application with Python in RPi	3 days	2021/3/11	2021/3/16	22



C Appendix-3: Python Scripts