Enhancing Spectrum Sensing for Cognitive Radio: Radio Signal Classification using Neural Networks

by

Siddhartha Subray

M.S., Telecommunications, University of Colorado Boulder, 2018

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2022

Committee Members:

Kevin Gifford

Qin Lv

James Curry

Keith Gremban

James Lansford

Subray, Siddhartha (Ph.D., Computer Science)

Enhancing Spectrum Sensing for Cognitive Radio: Radio Signal Classification using Neural Networks

Thesis directed by Dr. Kevin Gifford

With the rapid development of numerous wireless network technologies and the growing number of wireless devices in use globally, sharing the radio frequency spectrum has become a challenge that must be addressed. In recent years, methods for detecting and classifying features in photos, audio, and other types of data have been developed using Deep Neural Networks (DNN). DNN classification algorithms have demonstrated the ability to analyze audio signals with a similar structure accurately for a variety of applications including music recognition, speaker identification, earthquake detection, and sound localization. Recently, DNNs have found applications in the wireless networks domain, and radio frequency (RF) signal identification and classification is one of ideal applications for this machine learning (ML) technology. Given that widely used wireless technologies such as Wi-Fi, LTE, and 5G-NR share modulation schemes, it is beneficial to discern the type of signal, rather than simply identifying the modulation scheme of a signal in order to improve spectrum sensing capabilities. In this dissertation, a novel input feature engineering approach for processing signal I/Q data is proposed and evaluated using different types of supervised neural network architectures, such as the Deep Feedforward Neural Network (DFNN), Deep Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) Neural Network, to detect and classify between 5G-NR, LTE, and Wi-Fi transmissions. The dissertation demonstrates that the proposed feature engineering approach significantly outperforms existing methodologies and that with the appropriate input features, simple neural network architectures can achieve high signal classification accuracy.

Dedication

To my parents, who encouraged and supported me to pursue doctorate studies.

Acknowledgements

I want to thank the National Aeronautics and Space Administration (NASA), National Institute of Standards and Technology (NIST) and the University of Colorado Boulder, for supporting my doctoral studies, Dr. Kevin Gifford for serving as my academic advisor, supporting my research and helping me decide on the thesis topic, Dr. James Curry and Dr. Keith Gremban for their insights on the research process and methodologies, Dr. Qin Lv for her advice on machine learning and data science, Dr. James Lansford for his insights on spectrum sharing and sensing techniques, Dr. Stephen Braham from Simon Fraser University, Canada, for his insights on wireless networks, Dr. Adam Wunderlich and Dr. Aric Sanders from NIST for their advice on tuning SDR receivers and collecting high-quality signal data, Dr. André Rosete for his guidance, my fellow PhD students and friends; Stefan Tschimben, Evariste Some, Ganesh Byrandurga Gopinath, Siddharth Rao, Ramya Rao, Suhas Vyas, and Bharath Yoganarasimha. Finally, I want to thank my parents for their support and encouragement, without whom I would have never had this opportunity.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Over the previous decade, the quantity of wireless devices available has increased at an exponential rate. According to Cisco's Annual Internet Report, by 2023, more than 70% of the world's population will have mobile connectivity, and the number of mobile devices will reach 13.1 billion, including 1.4 billion with 5G connectivity [1]. In today's digital mobile society, it is critical to meet the demands of wireless users. The radio frequency spectrum is a precious resource that must be managed well in order for a large number of wireless devices and technologies to coexist. "According to FCC (2003a) notice of proposed rulemaking and order, spectrum utilization varies from 15% to 85% with wide variance in time and space," suggesting that the current spectrum scarcity is primarily due to inefficient static spectrum management rather than a physical shortage of spectrum, as stated by the authors in [2]. The existing radio spectrum policy assigns frequency bands to specific users for a specified period of time and only licensed users are authorized to access the spectrum while the unlicensed users are barred from utilizing allocated frequency bands even when it is not in use by licensed users [3]. This results in an inefficient use of this valuable resource. As a consequence, academic, commercial, and government researchers have been investigating new paradigms for utilizing the wireless spectrum with greater efficiency.

One such paradigm that aids in the sharing and coexistence of different wireless technologies, such as the 3rd Generation Partnership Project (3GPP) 5th Generation Technology (5G), Long-Term Evolution (LTE), and IEEE 802.11 (Wi-Fi), operating in the same spectrum of radio frequencies is spectrum sensing. Spectrum sensing enables wireless devices to monitor spectrum

usage in a certain band and allows the RF transmitters and receivers to alter their communication parameters [4] to enable shared spectrum access among different wireless protocols and technologies. As early as 2003, the Federal Communications Commission (FCC) recognized the value of opportunistic spectrum access, also termed Dynamic Spectrum Access (DSA), in allowing licensed and unlicensed users to share spectrum [5]. As the spectrum sharing paradigm evolves, spectrum sensing is becoming increasingly important. Spectrum sensing algorithms and signal classification neural networks are critical tools for achieving efficient spectrum sharing among various wireless network technologies.

Broadly speaking, spectrum sensing can be divided into two types: narrowband sensing and wideband sensing. Narrowband refers to a frequency range in which the frequency response is flat, implying that fading is consistent across the signal bandwidth. Essentially, the signal bandwidth is less than the coherence bandwidth of the channel. Wideband denotes a frequency range in which the frequency response is not flat, implying frequency selective fading, and the signal bandwidth exceeds the channel's coherence bandwidth [6]. Energy detection [7], eigenvalue-based detection [8], [9], cyclostationary feature-based detection [10], [11], and matching filter detection [12], [13] are examples of narrowband sensing approaches. Nyquist wideband sensing – where wideband signals are sampled at the Nyquist rate – and sub-Nyquist wideband sensing – where wideband signals are sampled at rates below the Nyquist rate – are two examples of wideband sensing approaches [14].

1.1 Spectrum Sharing

Spectrum sharing enables several users to utilize the same frequency bands, hence optimizing the usage of the wireless communications channels while managing interference. Spectrum sharing is vital as demand continues to grow and the wireless channels become congested. Smartphones, the Internet of Things (IoT), military and public safety radios, wearable devices, smart vehicles, and a number of other devices all share data, voice, and images across the same wireless bands of the electromagnetic spectrum. Institutions such as National Institute of Standards and Technology (NIST), the United States (U.S.) measurement authority, research projects at universities, such as

the Passive and Active Spectrum Sharing (PASS) project at the University of Colorado Boulder [15], are developing measurement tools, data and methodologies to enable fair and efficient spectrum sharing. Additionally, forensic tools for detecting and reporting violations, with the goal of establishing a "fingerprint" of spectrum use in time and space, are being developed [16]. Spectrum sharing approaches can be broadly classified [17] as follows:

- Spectrum Sensing (SS): An approach where the transmitters check to see if a wireless channel is in use and transmit when the channel is empty, or seek to move to an unoccupied channel. It is effectively a Listen Before Talk (LBT) approach. SS is susceptible to the 'hidden node' problem, in which a competing device may be actively utilizing the spectrum but at an imperceptible level, leading to interference.

- Cooperative Spectrum Sensing (CSS): A process in which the SS nodes exchange information regarding sensed activity in order to obtain a more comprehensive view of the spectrum usage. CSS minimizes, but does not eliminate the 'hidden node' problem entirely.

- Geo-location Databases (GL-DB): A process in which each device determines its Global Positioning System (GPS) coordinates and queries a database for available spectrum corresponding to its location.

- Beacon Signalling (BeS): A process in which a transmitter repeatedly transmits a known signal to indicate spectrum use.

- Command-and-Control (C2): A process in which master nodes are in charge of actively controlling and coordinating spectrum assignments.

The author in [18] briefly describes the types of spectrum bands and SS approaches. Spectrum can be allocated according to frequency, time, space, or code division. Diverse protocols and technologies that share a band frequently lack a standardized allocation of time and frequency resources. This is particularly true in unlicensed bands, where Bluetooth, Wi-Fi, LTE, and a number of other protocols frequently coexist. These protocols require devices to employ spectrum

sensing in order to make the most efficient use of available spectrum. Shared bands in both unlicensed and licensed spectrum, the technologies deployed in these bands, the spectrum sharing concerns, and current approaches to spectrum sharing, are summarized in order to highlight growing importance of spectrum sharing.

1.2 Licensed bands

Coexistence is also possible in licensed bands between Primary Users (PU) and Secondary Users (SU), where the SU may use the band only when the PU is not present - in this case, the SU must detect the PU using spectrum sensing techniques. These techniques must be robust to avoid interfering with the PU while optimizing channel utilization in the absence of the PU. Effective spectrum sensing techniques must accomplish both objectives.

1.2.1 CBRS band

The FCC designated the spectrum between 3550 MHz and 3700 MHz as the Citizens Broadband Radio Service (CBRS) frequency for shared usage in the United States [19]. The United States Navy operates in this spectrum band, which it utilizes for Air Traffic Control (ATC) aboard aircraft carrier ships. However, the U.S. Navy has little use for the band away from the coasts, and aircraft carriers are rarely present near any given location along the coast, so there is ample opportunity to share this band with telecommunications providers on a time and space basis, provided that the secondary users can quickly cease using the band if an aircraft carrier's ATC radar is detected. A radio Base Station (BS) operating in the CBRS band must have a Spectrum Access System (SAS), which is a frequency coordinator that dynamically handles spectrum sharing, with Environmental Sensing Capability (ESC). ESC employs a straightforward energy detection technique: if the energy in the band exceeds a threshold, the SAS calculates that an aircraft carrier is approaching and redirects all communications usage away from the CBRS band and onto other bands to avoid interfering with its ATC system.

1.2.2 Television White Spaces

The term Television White Spaces (TVWS) refers to frequencies in the Very High Frequency (VHF) and Ultra High Frequency (UHF) television broadcast bands that are either unassigned or unused by existing broadcast or other licensees. Television transmissions use specific channels in the VHF and UHF bands, with channel assignments differing by region. Not all authorized channels are utilized for broadcast in any particular market, resulting in 'White Spaces' where a channel that is not utilized for broadcast may be utilized for other purposes [20]. The IEEE 802.11af [21] and IEEE 802.22 [22] standards were developed with the goal of utilizing Cognitive Radio (CR) techniques to enable sharing of geographically unused spectrum allocated to the television broadcast service in order to bring broadband access to hard-to-reach, low population density areas typical of rural environments. In the United States, the FCC requires TVWS devices to determine their own location using the GPS, consult a geographical database of television broadcasters, and use only those television channels whose broadcasts are not intended to reach the SU location. This is a method of sharing spatial spectrum that does not require spectrum sensing. However, TVWS are underutilized, with relatively few commercial solutions available to capitalize on this spectrum opportunity [23].

1.3 Unlicensed bands

Numerous bands do not need prospective users to register or get permits from regulators in any way. Due to the availability of these unlicensed bands, Wireless Local Area Network (WLAN), Wireless Personal Area Network (WPAN) technologies such as Wi-Fi and Bluetooth have been developed. However, a lack of licenses and allocations forces users to employ coexistence techniques, which invariably entail some type of spectrum sensing.

1.3.1 ISM Unlicensed Band

The Industrial, Scientific, and Medical (ISM) band, which spans 100 MHz between 2.4 GHz and 2.5 GHz, is one of the widely used unlicensed bands. Various technologies and protocols such Wi-Fi, Bluetooth, Zigbee, medical devices [24], WPANs [25], cordless phones, IoT devices [26], wireless sensor networks [27], and scientific measurement instruments utilize this unlicensed band. Spurious and/or unintended emissions, such as those from microwave ovens [28] or leaky interfaces [29] may also be observed in the ISM band. The spectrogram in Figure 1.1 shows the signals captured from a microwave oven, along with Bluetooth and Wi-Fi signals in the 2.4 GHz ISM band.

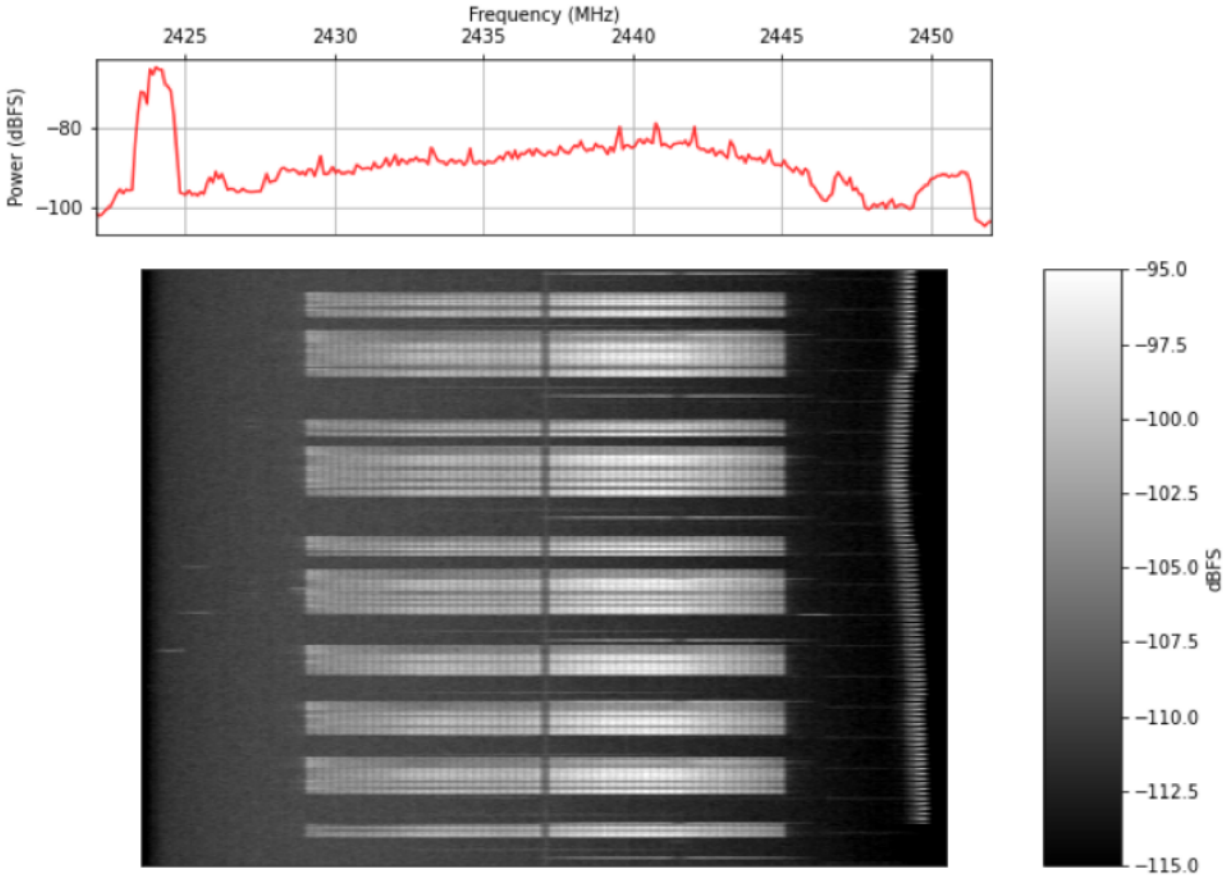

Figure 1.1: Spectrogram of the signals in the 2.4 GHz ISM band

These systems lack a standardized mechanism for planning and allocating time and frequency resources [30], posing a challenge that spectrum sensing mechanisms are designed to address. For instance, Wi-Fi uses a technique called Carrier-Sense Multiple-Access with Collision Avoidance (CSMA/CA) to avoid packet collisions. The CSMA/CA employs two strategies for spectrum sensing: initially, a matching filter is used to attempt to detect a Wi-Fi preamble, which indicates the presence of another Wi-Fi node on the channel. If the matching filter does not detect anything, Energy Detection (ED) is used to detect any additional non Wi-Fi channel occupants, such as Bluetooth broadcasts and microwave oven leakage. If the channel is found to be occupied in either case, the Wi-Fi node seeking to access it will wait a random amount of time within a range specified by the IEEE 802.11 protocol before transmitting again. The Bluetooth Special Interest Group report [31] indicates that approximately 6.4 billion Bluetooth devices, with 70% of those being peripheral devices, would be shipped in 2025 compared to 4 billion devices in 2020, demonstrating that ISM band utilization continues to increase and emphasizes the importance of spectrum sensing and spectrum sharing technologies.

1.3.2 U-NII Unlicensed Band

The Unlicensed National Information Infrastructure (U-NII) frequency bands, as defined by the FCC, are part of the radio frequency spectrum from 5.150 - 7.125 GHz. Since radars may use portions of this band, other users must use Dynamic Frequency Selection (DFS) to access those frequencies. Along with IEEE 802.11 technologies such as Wi-Fi 5, Wi-Fi 6, and Wi-Fi 6E, the band is also used by technologies such as LTE-Unlicensed (LTE-U) and License-Assisted Access (LAA) to provide additional downlink capacity for 4G LTE networks [32], and 5G New Radio (5G-NR). LTE-U is a proposed extension of the LTE wireless standard, originally developed by Qualcomm. It is designed to enable cellular network operators to offload some of their data traffic by utilizing the unlicensed 5 GHz frequency range [33]. One of the variants of LTE-U is the LAA, which was standardized by the 3GPP in Rel-13 [34].

1.4 Signal Classification

Signal classification is the process of assessing the type of a signal and is a form of spectrum sensing. A device attempting to utilize shared spectrum needs to first discover whether individual channels and time slots within these channels are already in use by other devices. Accurate spectrum sensing enables more efficient use of the available spectrum and is also beneficial for informing regulators and telecommunications operators on how the spectrum is being used.

Modulation schemes, such as 64 Quadrature Amplitude Modulation (QAM), 16 QAM, and Quadrature Phase Shift Keying (QPSK) are shared by wireless network technologies such as 5G-NR, LTE, and Wi-Fi, and recognizing the Radio Frequency (RF) signal's technology provides more information about the band's occupancy than only recognizing the modulation process of the signal. Current wireless classification methodologies primarily focus on identifying signals based on modulation type, with the categorization of wireless network technologies becoming more popular in recent publications.

Multi-class classification of signals is needed to identify and differentiate the transmissions from various technologies such as Wi-Fi, LTE, and 5G-NR. To that end, it is possible to train a Neural Network (NN) using In-phase and Quadrature (I/Q) samples or features extracted from I/Q data, at various Signal-to-Noise ratio (SNR) values, and surpass the classification capabilities of analytical spectrum sensing methods. A NN is a subset of Machine Learning (ML) that mimics the way biological neurons operate and aims to recognize the underlying relationships in a dataset. NN are capable of inferring complex and nonlinear interactions between dependent and independent variables making them highly suitable for identifying the patterns in a signal waveform.

1.5 Organization

Chapter 1 describes the importance of spectrum sensing, provides a summary of common spectrum sharing bands, along with spectrum sensing and sharing approaches. Chapter 2 describes the current literature in spectrum sensing and signal classification using ML, limitations

of the current research, and contributions of the dissertation. Chapter 3 describes complex signals, Software-Defined Radio (SDR) and direct conversion receiver, the widely used ML and NN architectures in this domain, and the proposed novel feature engineering approach for signal classification. Chapter 4 presents the results of the dissertation. Finally, conclusions and future work are presented in chapter 5. The data collection process and the list of acronyms are found in the appendix.

Chapter 2

Literature Review

2.1 Spectrum Sensing

The ability to receive, detect, and identify signals in a spectrum environment is a fundamental component of many wireless systems. Spectrum sensing is used in a wide variety of applications in telecommunications and security, including interference mitigation, and cyber resilience. Spectrum sensing techniques enable the SU to determine the PU's presence or absence on a particular frequency channel. Conventional sensing techniques include:

- Energy Detection [7]: A technique where the energy of the sample is calculated by averaging the squared magnitude of the Fast Fourier Transform (FFT) over the number of samples and the result is the compared to a predetermined threshold. The PU is considered present if the energy level exceeds the threshold; otherwise, the PU is considered to be absent. Numerous methodologies have been developed to enhance energy detection performance through the use of dynamic thresholds [35], [36].

- Eigenvalue-based Detection [8],[9]: Eigenvalue-based or Covariance-based detection technique determines the presence of a PU signal by analyzing the sample covariance matrix of the received signal and performing a Singular Value Decomposition (SVD). This is determined by examining the structure of the received signals' covariance matrix. The PU signals are correlated and can be distinguished from background noise. The eigenvalues of this matrix can be calculated using the SVD method. The ratio of the maximum to

minimum eigenvalues is then calculated and compared to a threshold value to determine the presence or absence of the PU.

- Cyclostationary Feature Detection [10], [11]: The detection is based on certain features of the received signal. Properties of the transmitted signal, such as the modulation rate and carrier frequency, are periodic and are referred to be cyclostationary characteristics. This approach outperforms energy-based detection techniques in terms of sensitivity. Additionally, because these techniques are capable of discriminating between signals and noise, they are less subject to noise uncertainty and so have a lower likelihood of false alarm than energy detection-based systems.

- Matched-Filter Detection [12], [13]: A method where the incoming signal is compared to pre-allocated and pilot samples acquired from the same transmitter. The test statistic is computed using these pilot samples and is then compared to a threshold. If the signal exceeds the threshold, it is considered present.

The radio frequency spectrum is a finite resource that must be managed efficiently. The FCC policies regarding the radio spectrum are largely based on static management, in which the radio spectrum is divided up and distributed for use via licenses. Due to the dramatic increase in wireless communication, this static management has resulted in a spectrum shortage. Cognitive radio has the potential to address this issue by dynamically managing the radio spectrum in order to ensure equitable access to all users [37],[38]. A cognitive radio is capable of sensing its environment and determining the presence of signals, and then adapting its parameters to maximize its transmission throughput without interfering with the signals of licensed users. This function enables unlicensed secondary users to sense the radio spectrum and identify available channels. The ability to accurately detect the primary user's signal is critical for secondary users to utilize idle licensed spectrum in cognitive radio systems.

As early as the 1980s, researchers examined methods for classifying wireless signals based on the modulation schemes. The author in [39] mentioned a method for real-time categorization

of Amplitude-Shift Keying (ASK), Frequency-Shift Keying (FSK), and Phase-Shift Keying (PSK). The approach was validated using computer simulations and found to be reliable and robust in the presence of expected perturbing effects such as noise, mismatches in the center frequency or bandwidth, and cross-talk between adjacent frequency channels. In [40], the authors proposed algorithms that incorporated time domain parameters in addition to the standard envelope and instantaneous frequency to detect the modulation type of a signal. The simulations, which included Gaussian noise perturbation, mismatch in the center frequency and bandwidth, supported theoretical arguments that the new parameters improve modulation type discrimination under noisy conditions.

The authors in [41] proposed a method for determining the type of digital modulation scheme based on the signal's known amplitude, phase, and frequency differences. The three properties were normalized and used to classify the waveforms according to their digital modulation scheme using a decision tree. In [42], the authors suggested a method for classifying MPSK modulation schemes based on particular predicted values for each MPSK modulation scheme, and then comparing the quasi-log likelihood ratio for the waveform to determine the PSK modulation order.

However, conventional sensing techniques such as energy detection and matched filter detection have several limitations such as the inefficiency of energy detectors at low SNR values, and impracticality of matched filter detection [43]; hence, several researchers have investigated the use of machine learning classifiers to detect the primary user's state, particularly in the context of cooperative spectrum sensing.

2.2 Machine Learning for Spectrum Sensing and Signal Classification

In this section, an overview of the literature on spectrum sensing and signal classification using ML is provided. Classification is a type of supervised learning technique used in machine learning. It is the process of dividing a collection of data into distinct categories. The objective of classification predictive modeling is to approximate the mapping function between discrete input variables and discrete output variables and ascertain the category to which the new input data

belongs. A classification problem may be a binary classification problem, in which data is classified into two classes, or a multi-class classification problem, in which data is classified into more than two classes.

For modulation scheme categorization, a combination of Spectral Correlation Density (SCD) and a neural network was provided by authors in [44]. Combining analytical feature extraction with a neural network to classify waveforms based on the obtained features was discovered to be a useful method. M. Hong et al. [45] proposed using the SCD function to categorize numerous users. Authors in [46] proposed that spectrum sensing based on ML techniques could provide high detection accuracy with low complexity that adapt to the environment by optimizing additional features.

O'Shea et al. [47] used a Convolutional Neural Network (CNN) to classify 11 modulation schemes, and the results were compared to Naive Bayes, Deep Neural Networks, K-nearest neighbor (KNN), Support Vector Machine (SVM), and a decision tree. Residual Neural Network (ResNet) was used to classify 24 modulation schemes by the authors in [48]. Their samples were subjected to fading and carrier frequency offset due to multipath fading. In [49], the authors demonstrated the successful use of a CNN as a feature extractor and a bidirectional Long Short-Term Memory (LSTM) to classify sources of transient radio interference and compared it to a method that used a SVM instead of an LSTM. They found that the CNN-LSTM classifier outperformed the CNN-SVM classifier in terms of classification accuracy for all classes. Li et al. [50] proposed utilizing a combination of CNN with one dimension for feature extraction and a Gated Recurrent Unit (GRU) to classify signals into 31 categories (4 modulation types, different channel coding methods, different frequency bands). Authors in [51] proposed a deep learning-based signal detector that utilizes the underlying structural information of modulated signals and demonstrates its ability to achieve state-of-the-art detection performance without prior knowledge of the channel state or background noise. Additionally, the effect of modulation scheme and sample length on performance was analyzed, and finally, the cooperative detection system based on deep learning was demonstrated to outperform conventional cooperative sensing methods.

In [52], authors proposed the use of distributed deep reinforcement learning for CSS in cognitive radio networks with correlated fading to learn the optimal CSS. To reduce the size of the solution space in large networks, the coordination graph technique is utilized to decompose the problem into a max-plus problem. Multiple machine learning-enabled solutions were proposed by authors [53] to address the challenges associated with complex sensing models in cooperative spectrum sensing for non-orthogonal multiple access transmission mechanisms. These solutions include unsupervised learning algorithms such as K-Means clustering and Gaussian mixture models, as well as supervised learning algorithms such as Directed Acyclic Graph - Support Vector Machine (DAG-SVM), K-Nearest Neighbor, and Neural Networks. Additionally, the sensing accuracy were examined in terms of the number of secondary users, the volume of training data, the average signal-to-noise ratio of receivers, the ratio of primary users' power coefficients, as well as the training and test times. The authors observed that supervised learning algorithms achieved better sensing accuracy than unsupervised learning algorithms, and DAG-SVM algorithm performed the best among the tested supervised learning algorithms.

Authors [54] used CNN to extract features from the observed signal to improve sensing performance. More accurately, a novel two-dimensional dataset of the received signal was created, and three CNN models, including LeNet [55], AlexNet [56], and VGG-16 [57], were trained for CSS. Additionally, sensing performance comparisons were made between the three CNN-based CSS schemes. Four supervised machine learning algorithms were compared in [58] based on their Receiver Operating Characteristic (ROC) and Area under the ROC Curve (AUC). In a high-noise environment, the Naive Bayes classifier, SVM, the Gradient Boosting Machine (GBM), and the Distributed Random Forest (DRF) all achieved a more accurate sensing than energy detection and their performances were evaluated in a variety of channels with varying signal-to-noise ratios. Their simulations demonstrated that all four algorithms outperformed the Neyman-Pearson classic detection method [59], and that the Naive Bayes Classifier was the most suitable algorithm of the four for spectrum sensing.

The authors [60] develop an STFT-CNN method for spectrum sensing based on the Short-

Time Fourier Transform (STFT) and CNN. The proposed method takes advantage of the signal samples' time-frequency domain information to achieve state-of-the-art detection performance. The method is particularly well-suited for signals from a variety of primary users and does not require a priori information. In their experiment, the authors assume that primary users transmit QPSK signals. Additionally, the proposed algorithm's signal-to-noise ratio robustness and generalization ability are analyzed. At -15 dB SNR, their proposed method achieves a detection probability of 90.2% with a false alarm probability of 10%.

2.2.1 Wireless Network Technology Classification

The authors in the above literature categorized wireless signals primarily on the basis of modulation schemes and the signals were not categorized according to the radio technologies commonly used for communication, such as IEEE 802.11 (Wi-Fi) and 3GPP LTE/5G. Identifying the access technology rather than just the modulation scheme provides additional information about the occupancy of a radio band, enabling more efficient spectrum management and detection of interference.

In this section, the current literature on classifying the signals based on the wireless network technology via machine learning is discussed. The literature is categorized based on the type of signals classified corresponding to the two major wireless technology standards, i.e., IEEE 802 wireless standards and 3GPP standards, as shown in figure 2.1.

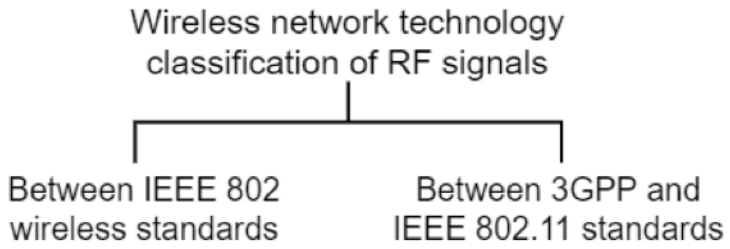

Figure 2.1: A taxonomy of commercial wireless network technology of RF signals

Signal classification scheme 1 - between IEEE 802 wireless standards: In this category, the

signals are classified between IEEE 802.11 (Wi-Fi), IEEE 802.15.1 (Bluetooth), and IEEE 802.15.4 (Zigbee) standards. Signal classification scheme 2 - between 3GPP and IEEE 802.11 standards: In this category, the signals are classified between IEEE 802.11 and LTE standards.

2.2.2 Between IEEE 802 Wireless Standards

The authors of [61] employ a random forest classifier technique in conjunction with a SVM to recognize and classify IEEE 802.15.1, IEEE 802.15.4, and IEEE 802.11 b/g/n signals in order to establish the source of interference in the 2.4 GHz ISM band. The authors generated the signals with Commercial off-the-Shelf (COTS) hardware in a variety of controlled and uncontrolled conditions, primarily using the Received Signal Strength Indicator (RSSI) data from the signal bursts for time and frequency domain feature extraction. Signals with varying SNR were used for training the models and the models were trained in different locations such as an office area, lab area with equipment, etc., and while the accuracy level depended on the test environment, on average they achieved greater than 90% classification accuracy. The performance of SVM and random forest classifier models were comparable in their experiments. The authors of [62] employ CNN to classify signals in the 2.4 GHz ISM band into Bluetooth, Zigbee, and IEEE 802.11 b/g technologies. Their sensing bands include ten Bluetooth channels, two Zigbee channels, and three Wi-Fi channels, totaling 15 output classes. The model was trained on IQ data that had been converted to the frequency domain through the FFT. The input SNR was adjusted from -20 dB to +20 dB. Classification of the Zigbee and Bluetooth channels achieved 95% accuracy on average for SNR greater than -7 dB and classification of Wi-Fi channels achieved 95% on average for SNR greater than 0 dB. The CNN model was designed by the authors based on the paper referenced in [47]. The authors of [63] use CNN to recognize and classify IEEE 802.11n, Zigbee, and Bluetooth signals. COTS devices were used to transmit the signals, and a spectrum analyzer was used to acquire the IQ data. For training the model, the authors used a range of scenarios, including homogeneous Wi-Fi, Zigbee, and Bluetooth, as well as heterogeneous Wi-Fi and Zigbee, Wi-Fi and Bluetooth, and Zigbee and Bluetooth. The SNR was varied between 0 and 30 dB, and a classification

accuracy of approximately 92% was achieved for SNR greater than 10 dB. Additionally, the authors compared the models' performance using a variety of machine learning algorithms, including SVM, Naive Bayes Classifier and others, and found that CNN produced the best results. The authors of [64] distinguished between IEEE 802.11 b/g/n protocols using KNN and Naive Bayes models. RSSI values were utilized to identify the pattern of a channel's activity and inactivity for the various IEEE 802.11 protocols, and thus classify them. The I/Q data was obtained using a spectrum analyzer, and the signals were transmitted using COTS hardware. Their results showed that Naive Bayes provided better identification accuracy of about 85.9% compared to the K-nearest neighbor's accuracy of 82.05% in a heterogeneous environment.

2.2.3 Between 3GPP and IEEE 802.11 Standards

The authors in [65] classify between IEEE 802.11g and LTE signals using CNN. COTS hardware was used to generate the data and the model was trained with both the I/Q samples and FFT transformation of the I/Q samples separately. The SNR of the samples ranged from 0 dB to 45 dB. A classification accuracy of greater than 90% was achieved when the SNR was greater than 10 dB and the accuracy was as high as 99% when SNR was 45 dB. The CNN model trained with FFT data performed better than the model trained with raw I/Q samples, especially at low (less than 15 dB) SNR values. The authors then used this capability for enhancing coexistence between simulated LTE and Wi-Fi transmissions in a shared spectrum. In their experiment, the LTE transmissions initially lasted for 20 milliseconds (ms) and remained idle for 2 ms and as soon as the neural network identified the presence of Wi-Fi transmissions during the idle time, the LTE system reduced its transmission duration from 20 ms to 10 ms and increased its idle time from 2 ms to 10 ms to allow more transmit opportunity (TxOP) for Wi-Fi. In [66], the authors classify between DVB-T, LTE, and Wi-Fi technologies using random forest decision trees, a fully connected neural network (FNN) and a CNN across multiple heterogenous environments and to study the difference in performance between manual and automatic feature extractions. The authors have achieved an accuracy of up to 99%.

In [67], the authors used various autoencoders, such as a deep standard autoencoder, LSTM autoencoder, and variational autoencoder to distinguish between LTE and IEEE 802.11 ac, IEEE 802.11 ax signals as a way of detecting anomalous RF signals in the wireless spectrum and compared the performance of well-known autoencoder architectures for the classification task. They used real-world LTE I/Q data, captured using an SDR, to train all the autoencoder models and tested against various combinations of LTE and IEEE 802.11 ax, IEEE 802.11 ac signals, including multiple modulation and coding scheme (MCS) values, and a precision of 99.9% and a recall of 88.1% were achieved in their models. The authors identified that the exponential linear unit (ELU) was the best activation function for this task when compared to other activation functions and one of their models required a training time of only 47 seconds to achieve an F1 score of 0.93, making it suitable for online training and deployment.

Table 2.1: Summary of the current literature for wireless network technology classification

Wireless technology classified	Input data	Model(s) used	Classification metric(s)	Maximum performance achieved
IEEE 802.11 b/g/n, Bluetooth, Zigbee [61]	RSSI values	SVM, Random forest classifier	Accuracy	96%, 98% respectively
IEEE 802.11 b/g, Bluetooth, Zigbee [62]	I/Q values	CNN	Accuracy	99%
IEEE 802.11n, Bluetooth, Zigbee [63]	I/Q values	CNN	Accuracy	93%
IEEE 802.11 b/g/n [64]	RSSI values	Naive Bayes, KNN	Accuracy	85.9%, 82.05% respectively
LTE, IEEE 802.11g [65]	I/Q, FFT	CNN	Accuracy	99%
IEEE 802.11, LTE, DVB-T [66]	RSSI, I/Q, FFT, Spectrogram	DFNN, Random Forest Classifier, CNN	Accuracy	87.4%, 88.6%, 99%, respectively
IEEE 802.11ac, IEEE 802.11ax, LTE [67]	I/Q values with phase and amplitude features	Deep standard, LSTM and variational autoencoder	Precision, Recall, F1 score	99.9%, 88.1%, 0.93, respectively

2.3 Limitations in the current methodologies

The current literature do not consider newer protocols such as IEEE 802.11ax and 5G. The majority of the literature focuses on older IEEE 802.11 protocols, such as IEEE 802.11g, and there is little information on how to use neural networks to classify newer communication protocols. Fur-

thermore, the majority of the available literature focus on classifying IEEE 802 Wireless Standards, and there is limited research on identifying and classifying 3GPP and IEEE wireless standards.

In literature where the authors classify between 3GPP and IEEE wireless standards, the over-the-air signal data captured is not representative of various real-world scenarios, such as active wireless traffic between a server and a client. For instance, in [66], the authors captured Wi-Fi beacons and compared them to LTE recordings from a base station. Additionally, current methodologies are heavily reliant on the signal strength of the data being captured and do not perform well in scenarios where cellular and WLAN signals are transmitted at comparable signal strengths. The field of signal classification based on wireless network technology is relatively new, with little research conducted, enabling opportunity for further research.

2.4 Contributions of this Dissertation

The use of neural networks for spectrum sensing and classification of RF signals is gaining prominence. As a result, it is essential to evaluate a variety of well-known neural network architectures and input feature engineering techniques for performing this task efficiently. The contributions of this dissertation are as follows:

2.4.1 Novel Input Feature Proposal and Evaluation

Current literature in this domain mainly use images, raw I/Q signal data, or RSSI values as the input features. This dissertation proposes a unique technique for feature engineering I/Q data and compares it to established methods. The purpose of analyzing various input features is to determine the optimal technique for processing the I/Q data in order to achieve efficient signal classification.

2.4.2 Baseline Model for RF Signal Classification using Neural Networks

Current literature in this domain mainly use the CNN for the classification of signals. Moreover, they do not consider the latest iteration of the WLAN and 3GPP technologies such as the

IEEE 802.11ax and 5G (or 5G-NR), respectively. Most of the literature use simulated RF signals, which do not represent the real-world signals accurately. This dissertation evaluates various well-known NN architectures such as the Deep Feedforward Neural Network (DFNN), CNN, and Recurrent Neural Network (RNN); in particular, LSTM, to classify between real-world Wi-Fi (IEEE 802.11ax), LTE, and 5G signals captured over-the-air and identify an effective NN architecture, if any. Capturing over-the-air signals enables neural networks to be trained on data that accurately represents signals in the real world and helps to establish a baseline model for the classification task.

Chapter 3

Methodology

3.1 Overview

This chapter describes the equipment, the methodologies used to obtain real-world RF signal datasets, the ML algorithms used in the domain of wireless technology classification and finally, the NN architectures used in the dissertation. A supervised classification approach is adopted for the wireless network technology classification, where the NN models are trained on labeled datasets. The required wireless signals were generated using COTS hardware and the transmissions were captured in the form of I/Q data using an SDR. The experiment was set up such that the signals captured from different protocols and technologies had an overlapping modulation scheme, which in this case is 64QAM, to ensure that the wireless access technology can be classified even with the same modulation scheme of the signals. The following signals were captured over-the-air for this dissertation:

- Wi-Fi: IEEE 802.11ax
- LTE
- 5G-NR

The data collection process is described in Appendix A. The I/Q data were captured at a sampling rate of 20 MS/s for all the signals.

3.2 Software-Defined Radio (SDR)

For the majority of radio's existence, tasks such as tuning to a certain frequency and demodulation were performed by physical hardware with sophisticated circuitry. This meant that a radio's capabilities were either extremely limited and predetermined before development, or were hidden under a layer of complexity that necessitated the use of trained operators. SDR abstracts away the majority of this complexity. While an SDR still requires an antenna and amplifier, it digitizes the raw data, which are then processed on a computer. The USRP^{TM} B200mini-i SDR, shown in Figure 3.1, was utilized to capture the RF signals for this research. The B200mini-i has a frequency range of 70 MHz to 6 GHz and an industrial-grade Xilinx Spartan-6 XC6SLX75 Field Programmable Gate Array (FPGA). The RF front end uses the Analog Devices AD9364 Radio-Frequency Integrated Circuit (RFIC) transceiver and supports up to 56 MHz of instantaneous bandwidth. The device is powered by a high-speed Universal Serial Bus (USB) 3.0 connection, which allows data to be streamed to the host computer and has a 12-bit Analog-to-Digital Converter (ADC). As with other USRP SDRs, the B200mini-i is interfaced via the open-source USRP Hardware Driver (UHD) Application Programming Interface (API).

Figure 3.1: USRP^{TM} B200mini-i SDR [68]

3.2.1 In-phase and Quadrature (I/Q) data

Quadrature signals, often referred to as I/Q signals, or IQ data, are frequently employed in RF applications. They serve as the foundation for modulation and demodulation of complex RF signals, both in hardware and software, as well as for complex signal analysis. When two periodic signals differ in phase by 90 degrees, they are said to be in 'quadrature' The signal that is 'in phase' or the reference signal is referred to as 'I', whereas the signal that is 90 degrees out of phase is referred to as 'Q'.

A sinusoidal wave, $s(t)$, is given by the equation:

$$s(t) = A * cos(2\pi Ft + \phi) \tag{3.1}$$

where, A is the amplitude, F is the frequency, t is the time, and ϕ is the phase shift. A sine wave and a cosine wave together form a quadrature signal due to their 90 degree phase shift as shown in Figure 3.2. By convention, the cosine wave is considered to be the in-phase component, whereas the sine wave is defined as the quadrature component.

Let $x(t)$ be the modulated RF carrier. It is given by:

$$x(t) = I * cos(2\pi ft) + Q * sin(2\pi ft) \tag{3.2}$$

where, I,Q are the amplitudes of the cosine and sine wave, respectively.

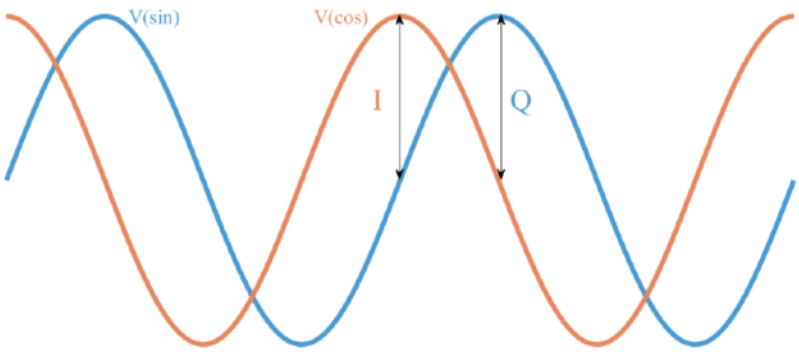

Figure 3.2: Sine and Cosine wave [69]

Modulated RF signals may be created using time-varying I and Q signals. With the I(t) and Q(t) baseband signals, an RF signal with any form of modulation may be generated by varying the amplitudes of the cosine and sine waves that are summed together. Figure 3.3 illustrates the process of quadrature modulation.

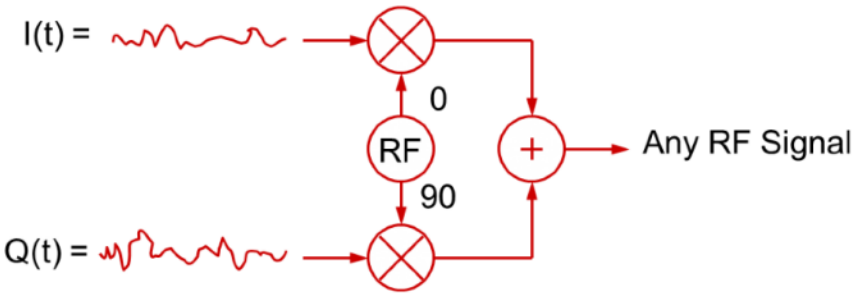

Figure 3.3: Quadrature modulation process [70]

To demodulate an RF signal, the same method is used in reverse. I(t) and Q(t) baseband signals can be created by combining an RF signal with a quadrature signal from the local oscillator. Figure 3.4 illustrates the process of quadrature demodulation.

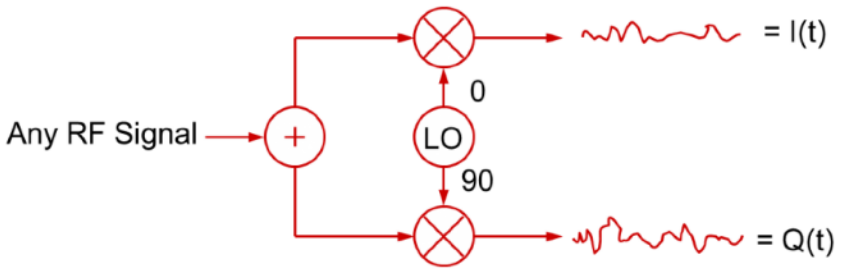

Figure 3.4: Quadrature demodulation process [70]

3.2.1.1 Phasor Representation

The I/Q convention is a way of describing the magnitude and phase of a signal, which leads to the concept of complex numbers and their ability to be represented on a complex plane. When a sinusoidal wave is depicted as a vector in the complex plane, it is known as phasor. The I/Q data

is represented using a complex number in the form, $I + Qj$. The Figure 3.5 is known as a 'phasor diagram', where a complex number is treated as a vector. The magnitude and phase of the signal is calculated as follows:

$$Magnitude = \sqrt{I^2 + Q^2} \tag{3.3}$$

$$Phase = \tan^{-1}(Q/I) \tag{3.4}$$

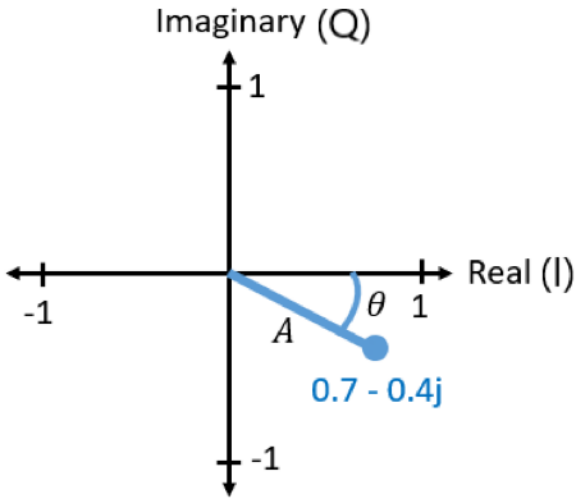

Figure 3.5: Phasor diagram [71]

3.2.2 Direct-Conversion Receiver

SDRs typically use a Direct-Conversion Receiver (DCR) architecture. The block diagram of a DCR is shown in Figure 3.6.

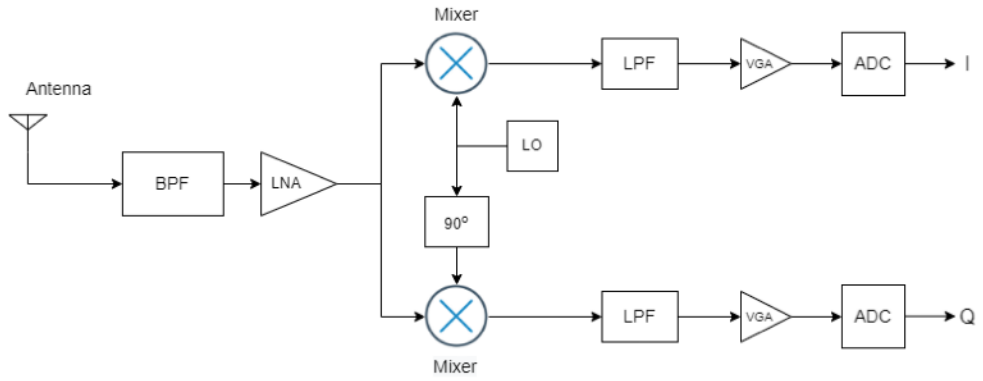

Figure 3.6: Architecture of a Direct-Conversion Receiver

A direct-conversion receiver, sometimes referred to as a homodyne or zero-IF receiver, is a particular form of receiver design. Receivers that use direct conversion architecture convert an RF signal to a 0 Hz signal. They are sometimes regarded as low-cost solutions due to the fact that they require minimal components. Typically, direct-conversion receivers filter and amplify an RF input signal received. The signal is then combined with a Local Oscillator (LO) signal of the same frequency as the RF input signal. Thus, the input signal is transformed to a 0-Hz signal at the mixer's output. Although the combination of the RF and LO signal frequencies appears at the mixer's output, this product is filtered away by the mixer's low-pass filter. The demodulated baseband output is then processed. Direct-conversion receivers are frequently combined with two mixers to form an I/Q demodulator. Both mixers are driven by the same LO. However, the LO signals to each mixer are 90 degrees out of phase. Following I/Q demodulation, the signal is input to an ADC to obtain the digital representation of the I and Q components.

3.2.3 Nyquist-Shannon Theorem

The Nyquist-Shannon sampling theorem is a key signal processing theorem that acts as a link between continuous-time and discrete-time signals. It presents a necessary condition for a sample rate at which a discrete sequence of samples may capture all of the information contained in a continuous-time signal with finite bandwidth. The theorem states, "If a function $x(t)$ contains no

frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart" [72]. Alternatively, it can be stated as: the sampling rate must be at least twice the bandwidth of a signal. It is important to note that a baseband signal's bandwidth is equal to its maximum frequency, and the bandwidth of a bandpass signal is equal to the difference between its upper and lower bounds [73]. Due to the fact that a complex signal has twice the information, namely the I and Q components, as compared to a real signal sample, the Nyquist theorem for complex sampling requires that the sampling rate be at least equal to the signal's bandwidth instead of twice the bandwidth.

3.2.4 Noise and Clipping

Noise is defined as any unwanted signal (typically random) that is added to the desired signal, causing it to deviate from its original value. All electrical systems generate noise. There are numerous types of noise [74]:

- Thermal Noise: A temperature-dependent noise generated by the physical movement of electrons within an electrical conductor.
- Flicker Noise: Also known as 1/f noise, occurs in nearly all electronic devices and is caused by a variety of factors, including impurities in a conductive channel, noise generation and recombination in a transistor as a result of base current, and so on. It has a power density that is inversely proportional to the signal's frequency.
- Burst Noise: Electronic noise generated by device defects in semiconductors, which makes it random and mathematically unpredictable. It is characterized by abrupt transitions between two or more levels.
- Quantization Noise: It's produced when an infinite number of analog voltages are converted to a finite number of digital codes. Quantization noise is the effect of representing a continuous analog signal with a discrete number. Quantization noise is the term used to describe the rounding error [75].

These types of noise can enter the signal chain in a variety of ways [74], including:

- ADC (adds thermal and quantization noise).
- Amplifiers (causes thermal and 1/f noise).
- Voltage references
- Non-ideal power supply
- Internal or external clocks
- Defective Printed Circuit Board (PCB)
- Sensors

Clipping is a type of distortion that occurs when a signal exceeds a specified threshold. It can occur when a signal is recorded by a sensor with a limited data range, or at any other point in the transformation of an analog or digital signal. It can be caused by exceeding the input differential voltage range of the amplifier. For example, clipping and distortion in a signal captured by an SDR is observed when the gain on the SDR is set too high and too close to a transmitter due to the SDR operating in a non-linear region as shown in Figure 3.7 and Figure 3.8.

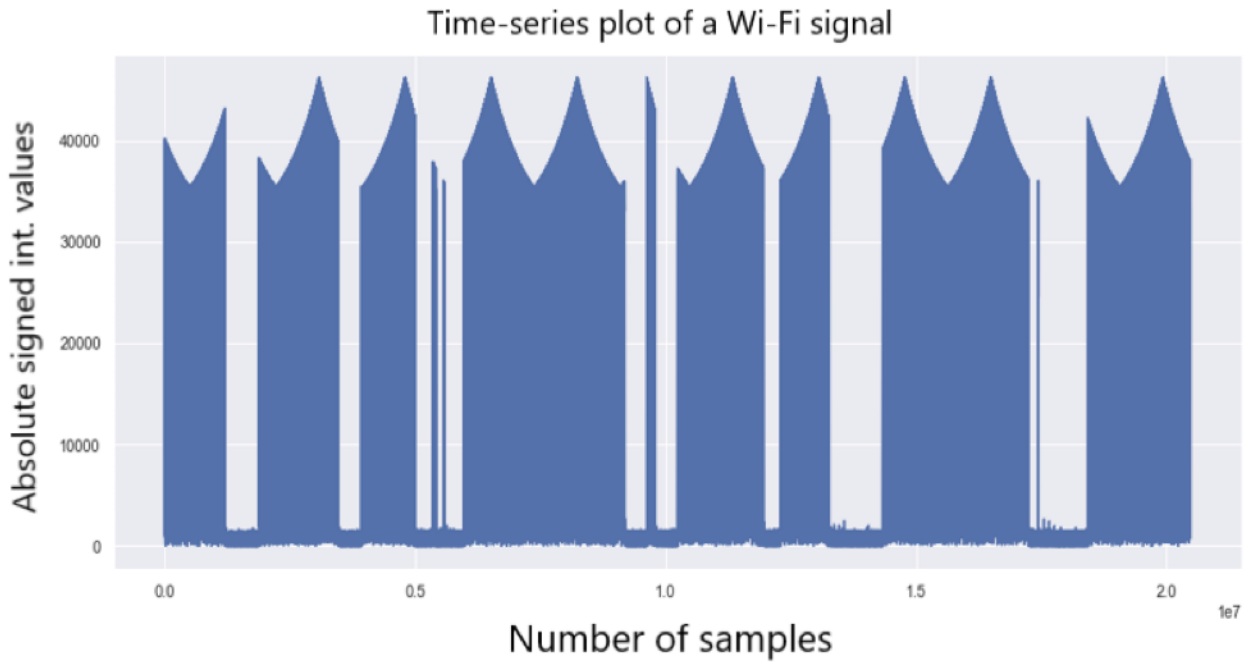

Figure 3.7: Example of a distorted signal

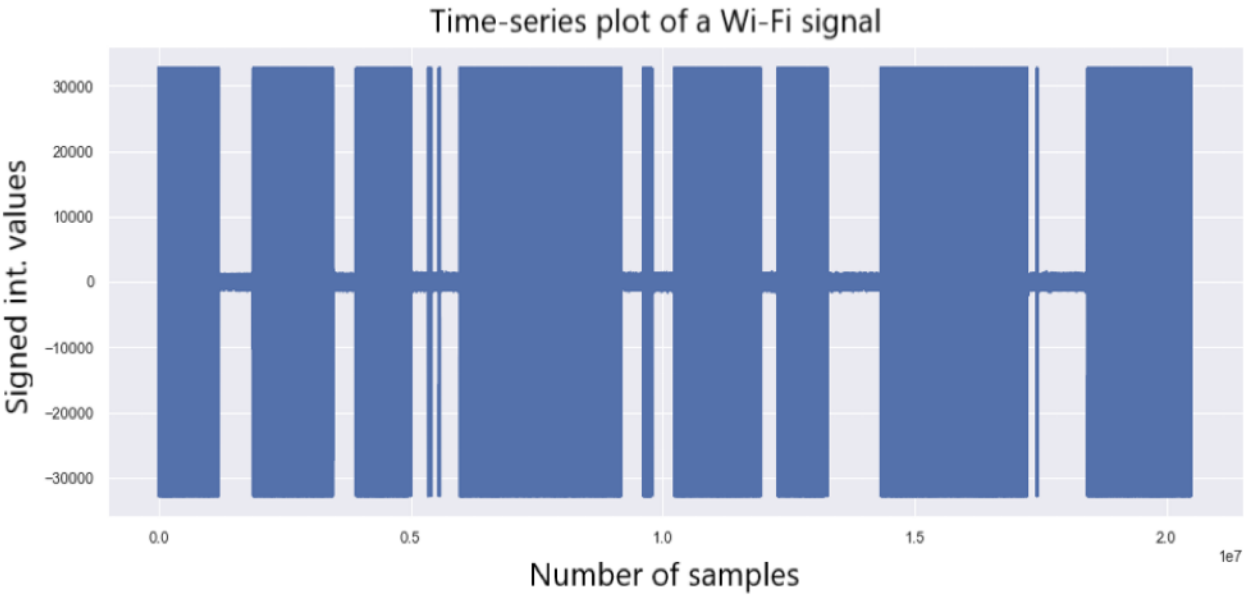

Figure 3.8: Example of a clipped signal

3.2.5 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is an algorithm to transform a signal in the time-domain into a signal in the frequency-domain. It represents a signal in terms of its constituent frequencies. If $x[n]$ is a sequence of N complex numbers, the DFT of $x[n]$ is given by:

$$X[j] = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi ijn}{N}} \tag{3.5}$$

The output, $X[j]$, is a sequence of N complex numbers that represent the amplitude and phase of a sinusoidal wave with a frequency of k/N cycles per time unit. The DFT is computed using the FFT algorithm since it is an efficient method for computing the DFT as it reduces the time-complexity of the computation from $O(N^2)$ to $O(NlogN)$.

3.3 Machine Learning Algorithms

ML is a method for computers to learn and improve without being explicitly programmed. ML algorithms are well-known for their ability to solve problems requiring pattern recognition, anomaly detection, and prediction using historical data. ML is becoming ubiquitous and is finding applications in many fields in recent times.

The following is a summary of the most frequently used machine learning methods utilized by researchers in the field of wireless network technology classification of signals:

3.4 Support Vector Machine

SVM is a supervised machine learning technique that can be used for data classification and regression analysis. The method was first published in 1963 [76] and has since evolved. One of the most popular versions that is being used today was introduced by Vapnik and Cortes [77] in 1995. The SVM algorithm's purpose is to generate a decision boundary (hyperplane) that divides various classes while maximizing the separation distance (margin). The data for categorization can be separated linearly or non-linearly.

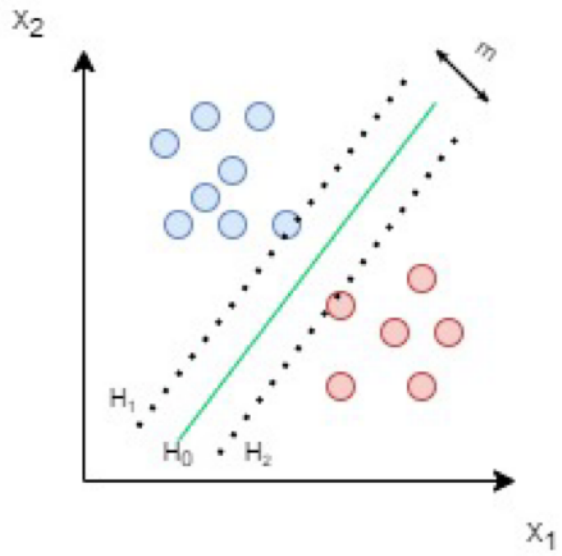

Figure 3.9: SVM margin for linear binary classification

Figure 3.9 shows the hyperplane (H_o) and margins (H_1, H_2) for an SVM trained for classifying two linearly separable data. The samples on the margin are called the support vectors. If there are n elements in a dataset, the dataset can be represented as:

$$\{(x_i, y_i) \mid x_i \in R^p, y_i \in (-1, 1)\}_{i=1}^{n} \tag{3.6}$$

where x_i is a vector of dimension p and y_i indicates the class 1 or -1 to which the data x_i belongs.

If H_o is the hyperplane, its equation is given by:

$$H_o = w^T x = 0 \tag{3.7}$$

where w is a vector normal to the hyperplane.

Given H_o, two other hyperplanes H_1 and H_2 can be selected such that H_o is equidistant from H_1 and H_2, which are given by:

$$H_1 = w^T x = 1; \quad H_2 = w^T x = -1 \tag{3.8}$$

Data points that are on or above the boundary H_1 have label 1 and data points that are on or below the boundary H_2 have label -1. The margin m between the two hyperplanes is given by:

$$m = \frac{2}{||w||} \tag{3.9}$$

The goal of the algorithm is to maximize the margin m. Finding the optimal hyperplane is an optimization problem and among the possible hyperplanes, the hyperplane with the smallest $||w||$ is selected as it has the largest margin.

3.5 Naive Bayes Classifier

This is a Bayes' theorem-based categorization algorithm. It is assumed that the presence of a particular feature in a class is independent of any other feature. If x, given by $x = \{x_1, x_2....x_n\}$, is a feature vector of length n is to be classified, the probability of x belonging to a class A is given by:

$$p(A|x) = \frac{p(A) * p(x|A)}{p(x)} \tag{3.10}$$

where $p(A|x)$ is the probability of class A given input x, $p(A)$ is the probability of class A, $p(x|A)$ is the probability of input x given class A, and $p(x)$ is the probability of input x.

3.6 K-Nearest Neighbor

KNN is an algorithm that classifies data based on a similarity measure, i.e., distance function. The algorithm assumes that similar things exist in close proximity. If x is the input to be classified, the class to which it belongs is given by the most common class among its K nearest data points, measured by a distance function. One of the distance functions is the Euclidean distance. The Euclidean distance between two points, x and y, is given by:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.11}$$

where x_i and y_i are the input point and an existing data point respectively and n is the number of features in the data.

The distance is calculated for all the existing data points and the input data is assigned a class which is the most common among the K nearest data points.

3.7 Artificial Neural Networks

An Artificial Neural Network (ANN) is a subset of ML designed to solve complex problems such as speech recognition, pattern recognition, make predictions and so on. ANNs use artificial neurons that are modeled loosely after the human brain to interpret the input data, perform calculations, and provide an output. Neurons are the building blocks of neural networks and were first introduced in 1958 [78].

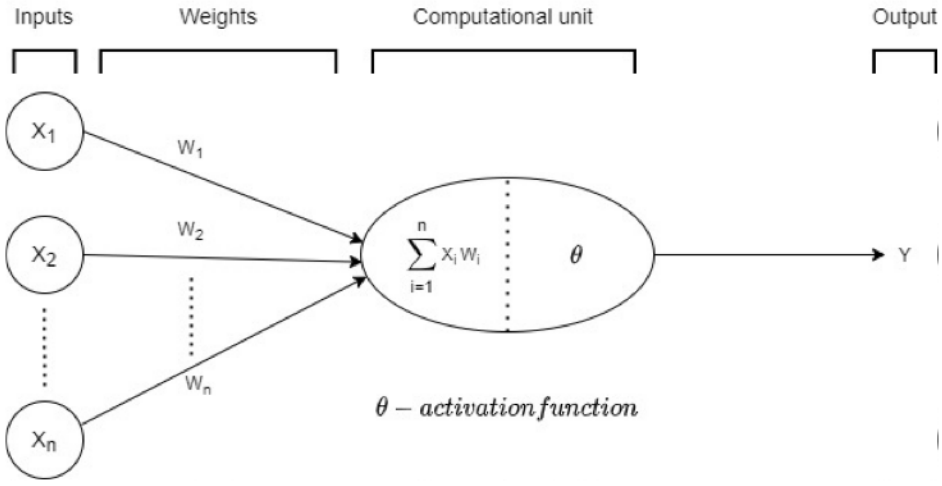

Figure 3.10: Components of an artificial neuron (perceptron)

An artificial neuron, shown in Figure 3.10, consists of the following components:

- Inputs – the data input in the form of a vector

- Weights – the values that determine the strength of the input connection to the neuron and are updated during training
- Computational unit – calculates the summation of the product of the inputs and weights and passes it as an input to an activation function, which generates a value depending on the type of activation function used. Some of the activation functions include sigmoid function [79], softmax function [80], Rectified Linear Unit (ReLU) [81], and hyperbolic tangent.
- Output – the value output by the activation function

The neural networks are constructed from 3 types of layers: the input layer consisting of the input data, the hidden layer consisting of the neurons and the output layer for the output. Neural networks with multiple layers are called deep neural networks. There are several types of neural network architectures as shown in Figure 3.11:

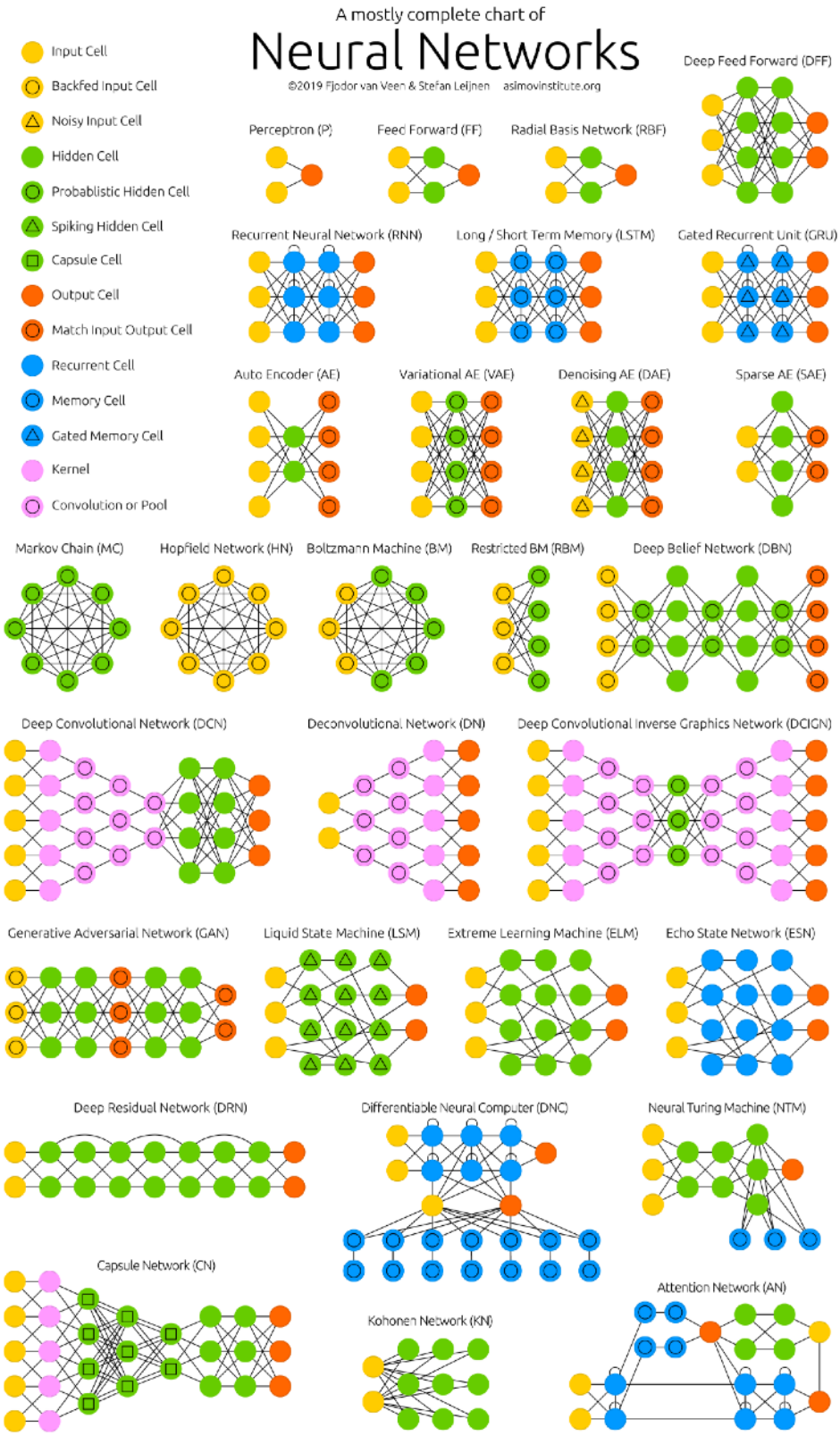

Figure 3.11: Types of neural network architectures [82]

The following are some of the widely used architectures of neural networks, and the architectures evaluated for the purpose of signal classification in this dissertation:

3.7.1 Deep Feedforward Network

Introduced in 1965 [83], the DFNN, also known as a Multilayer Perceptron (MLP), was the first deep neural network to be developed.

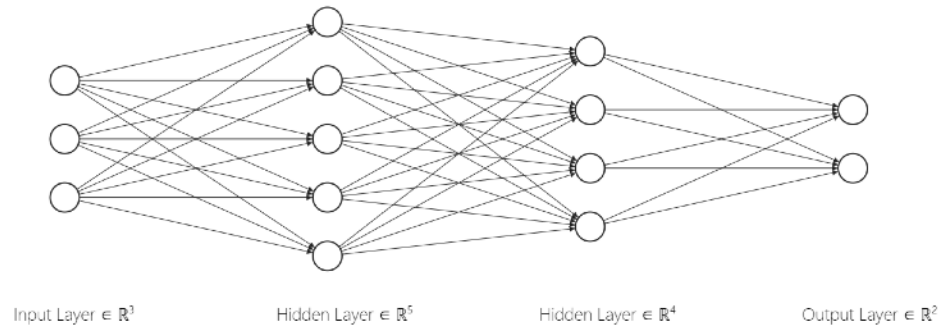

Figure 3.12: Deep feedforward network architecture

As the name suggests, there are multiple layers of perceptrons that are fully connected to each other between the adjacent layers and the input data is fed from the left of the neural network to the right. The training of the MLP model proceeds in two phases:

- In the forward phase, the input signal propagates forward, neuron by neuron, through the network and an output is obtained at the end of the network. The output 'y' at a neuron 'm' is a function of the input signals, associated weights, and its activation function, given by:

$$y_m = h\left(\sum_{i=1}^{n} x_i w_i + b\right) \tag{3.12}$$

where x_i and w_i represent the i^{th} input and weight connected to the neuron, respectively, b is the bias unit, h is the activation function and n is the number of inputs in the previous layer.

- In the backward phase, the neurons are trained using the backpropagation algorithm [84]. An error signal is generated by comparing the output of the network (estimated output) with the desired output. The error 'e' at the m^th neuron is given by:

$$e_m = z_m - y_m \tag{3.13}$$

where z_m is the expected output and y_m is the actual output at the neuron.

The resulting error value is propagated backward from the output layer towards the input layer and the weights are adjusted using the gradient descent algorithm [85].

The training process is repeated until the mean square error (MSE) of the neural network is minimized.

3.7.2 Convolutional Neural Network

The CNN is a prominent image processing algorithm that was first introduced in 1989 [55]. Although the approach is most commonly used for image analysis, it can also be utilized for other data analysis and classification challenges. CNNs are adept at seeing patterns in data and extracting relevant information.

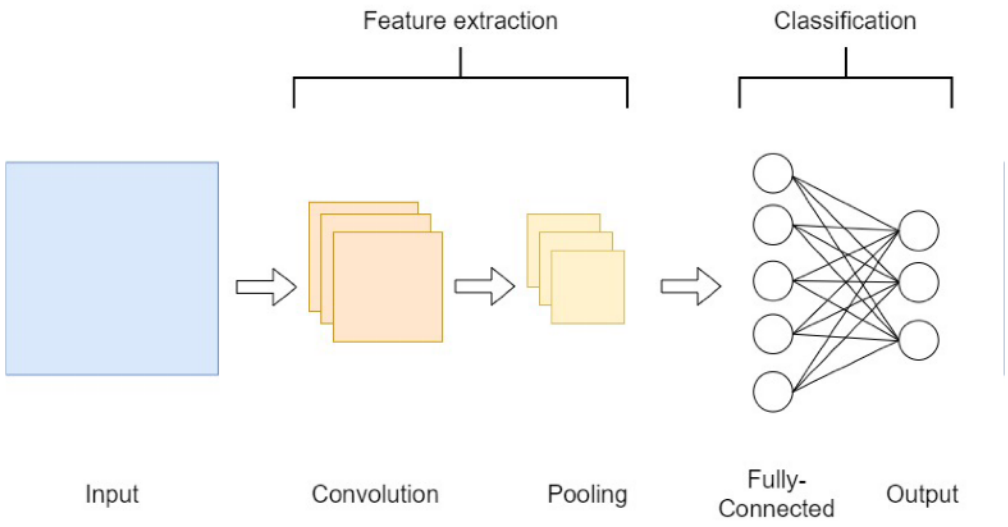

Figure 3.13: Architecture of a CNN

In a CNN, a convolution layer follows the input layer as shown in Figure 3.13. The convolution layer consists of a set of filters. Filters are relatively small matrices compared to the input matrix, for example a 3x3 matrix, and are initialized with random numbers. The filters slide over the input matrix at defined steps known as strides. A stride of 1 implies that the filter shifts by 1 unit. As the filter slides over the input matrix, the dot products between the filter and the input matrix are computed. The convolution operation between a 2D input matrix 'x' and a 2D filter matrix 'c' is given by:

$$(x * c)_{i,j} = x[i,j] * c[i,j] = \sum_{m}\sum_{n} = x\,[m,n]\,c\,[i-m]\,[j-n] \tag{3.14}$$

where m,n are the height and width of the filter, respectively.

The dot product is then passed to an activation function, such as the Rectified Linear Unit (ReLU), and the output is stored. The filter moves to the next input patch and the process is repeated until the filter covers the entire input matrix. The ReLU function is given by:

$$h\,(a) = max\,(0, a) \tag{3.15}$$

where a is the activation function's input.

The output of the convolution layer is a feature map that detect the patterns in the input. The output 'y' at l^{th} neuron is given by:

$$y_l = h\left((x * c)_{i,j} + b_l\right) \tag{3.16}$$

where h is the activation function and b_l is the bias unit.

The feature map is followed by a pooling layer. The pooling layer consists of a small matrix, such as 2x2 matrix, which is used to reduce the dimensions of the feature map while retaining the important information. The most common type of pooling is max pooling, which selects the highest value from a patch of the feature map it slides over. The number of convolution and pooling layers is a configurable parameter and is not limited to one. Following the pooling layer, the CNN consists of a fully connected layer. The output from the pooling layer is flattened to a vector and is fed to the fully connected layer. The final layer of the CNN is the output layer.

3.7.3 Recurrent Neural Network

The RNN is a model that has feedback loops and uses inputs from the previous stages to influence the current values. These models are useful for processing sequential data, such as audio, video, and so on. The RNNs feed the output of units as inputs to the units in the next time-step as depicted in Figure 3.14.

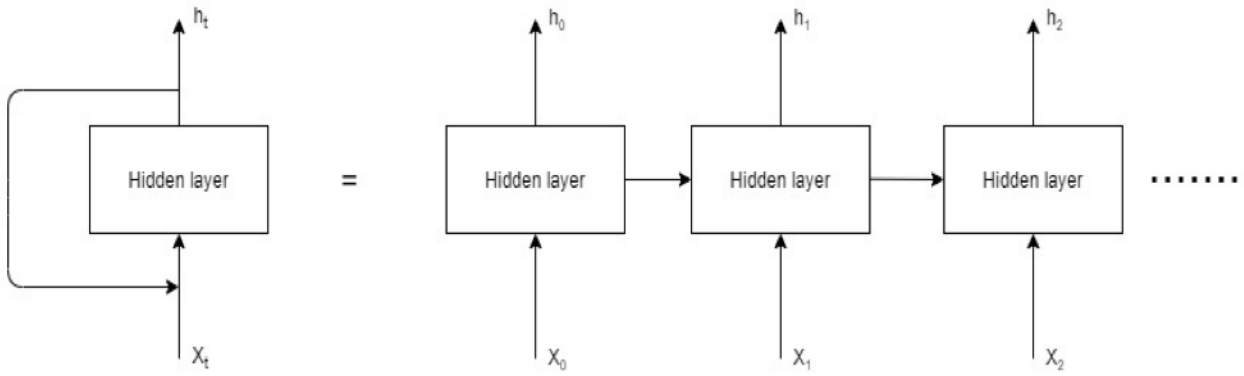

Figure 3.14: Architecture of a traditional RNN

In traditional RNNs, the repeating units have a simple structure, such as a single tanh layer as shown in Figure 3.15:

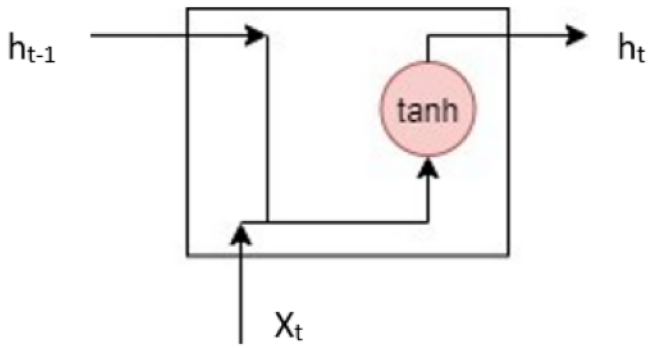

Figure 3.15: Repeating unit in a traditional RNN

One of the problems of traditional RNNs is the exploding gradients problem. The gradient is the value that is used to update the neural network's weight. As the sequence of units get longer, the chain of parameter multiplication gets longer and when several weights are multiplied, the losses become highly sensitive to the weights resulting in steep slopes in the cost function. Updating the weights with a large slope values causes the weights to become too large, which negatively affects the optimization of the model. The other problem traditional RNNs suffer from is the vainishing gradient problem. The vanishing gradient problem is when the gradient shrinks as it backprop-

agates through time. As a result, the difference between the updated weights and the previous weights become small in the initial layers of the model resulting in optimization issues in the earlier layers. This causes the RNN to have a short-term memory. To alleviate the exploding gradients and short-term memory problems found in the traditional RNNs, Long Short-Term Memory LSTM and GRU architectures were created. LSTMs and GRUs are special forms of RNNs that use gated modules to retain the important values in a sequence of data points. These architectures are widely used in applications such as speech analysis.

3.7.3.1 Long Short-Term Memory

Introduced in 1997 [86], the LSTM is a special form of RNN that is designed to store information for long periods of time compared to a traditional RNN. The structure of an LSTM unit/cell is as shown in Figure 3.16:

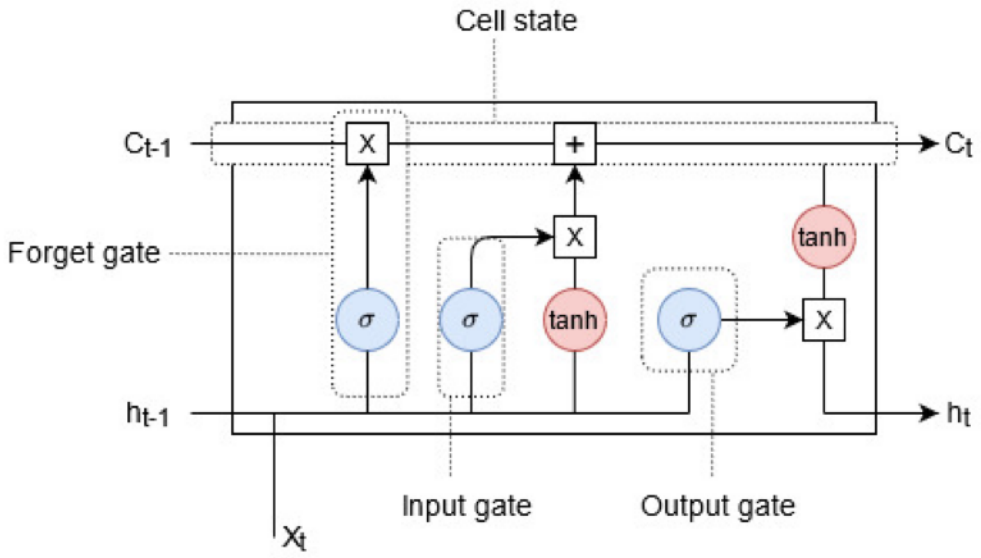

Figure 3.16: Repeating unit in an LSTM

In the first step, the forget gate layer decides the information to be discarded from the previous cell state C_{t-1}. The input vector X_t is concatenated with the previous unit's hidden state vector

h_t. The output of the forget gate layer is given by:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3.17}$$

where W_f are the weights associated with the forget gate layer, h_{t-1} is the previous cell's hidden state (or output), x_t is the input to the current cell, b_f is the bias unit.

In the second step, the new information to be stored is decided. The input gate layer decides the values to be updated, the tanh gate layer creates a vector with new values that could be added to the cell state. The output of the input gate layer is given by:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{3.18}$$

The output of the tanh gate layer, adjacent to the input gate layer, is given by:

$$\overline{C_t} = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{3.19}$$

where W_i, W_C are the associated weights and b_i, b_C are the bias units of the input gate layer and the tanh gate layer respectively.

In the third step, the cell state is updated. The previous cell state C_{t-1} is multiplied by f_t to discard previous information and the resulting value is added to the product of i_t and C_t.
The new cell state is given by:

$$C_t = f_t * C_{t-1} + i_t * \overline{C_t} \tag{3.20}$$

The output from the output gate layer o_t is used to determine the portion of cell state to be used in the final output. It is given by:

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{3.21}$$

where W_o and b_o are the weights and bias associated with the output gate layer respectively.

The cell output h_t is given by:

$$h_t = o_t * tanh\,(C_t) \tag{3.22}$$

LSTM uses the gates to select the information to be stored or discarded during the input sequence processing and the cell output h_t can be used for predictions.

3.8 ML Model Evaluation

Evaluation metrics are vital for selecting machine learning models and a critical part of ML since the techniques are used to determine the efficiency with which a model predicts outcomes or classifies input data. An evaluation metric is a mathematical quantifier of the model's quality. The following metrics will be used to evaluate the classification efficiency of the neural networks described in Section 3.7:

- Accuracy: The outcome of a classification task is either correct (True) or erroneous (False). Classification accuracy is expressed as a percentage of correctly classified data:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{3.23}$$

  Accuracy is suitable for evaluation only when the dataset is balanced, i.e., when there are equal number of samples in each class. For example, in a sample of 100 items, if 95 belong to class A and 5 belong to class B, a prediction of class A yields 95% accuracy. However, this model is no better than a model without any predictive capabilities. Hence, accuracy cannot be used with imbalanced datasets.

- Confusion Matrix: Given an N-class classification model, a confusion matrix is a N * N matrix that summarizes the classification model's performance. One axis contains the

predicted classes/labels by the model, while the other axis contains the actual classes. There are four possible outcomes from a confusion matrix:

* True Positive (TP): Sample is predicted as positive and is truly positive
* False Positive (FP): Sample is predicted as positive, however, it is truly negative
* False Negative (FN): Sample is predicted as negative, however, it is truly positive
* True Negative (TN): Sample is predicted as negative and is truly negative

The confusion matrix, shown in Figure 3.17, can be summarized as follows:

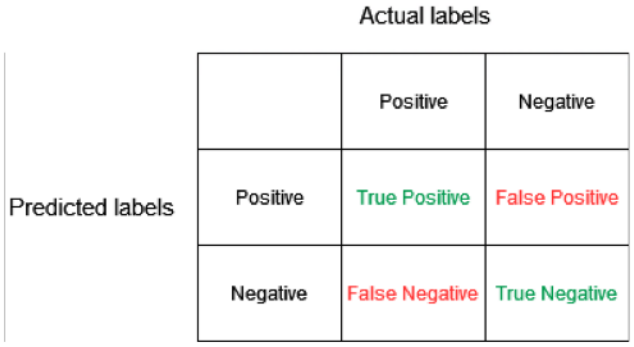

Figure 3.17: Confusion Matrix

A false positive is known as a Type-I error and a false negative is known as a Type-II error.

- ROC: Is a graphical representation, shown in Figure 3.18, of a classifier system's diagnostic capability as its discriminating threshold is varied. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds. TPR is given by:

$$TPR = \frac{TP}{TP + FN} \tag{3.24}$$

FPR is given by:

$$FPR = \frac{FP}{FP + TN} \tag{3.25}$$

- AUC: Measures the two-dimensional area underneath an ROC curve and is a performance metric that aggregates a model's performance over all possible categorization thresholds. An AUC score of 0.0 indicates that a model's predctions are all incorrect and a score of 1.0 indicates that all predictions are correct.

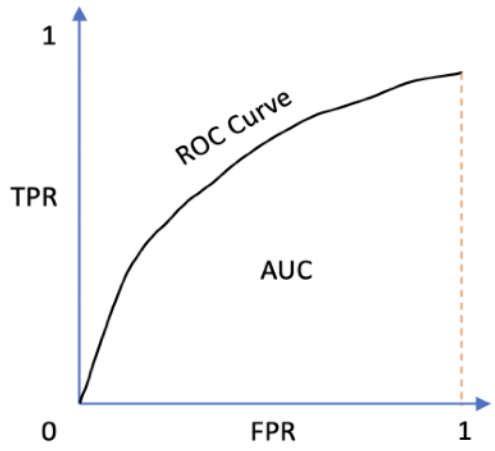

Figure 3.18: ROC curve and AUC

- Precision and Recall: Precision measures the proportion of positive prediction results that are correct. Precision is given by:

$$Precision = \frac{TP}{TP + FP} \tag{3.26}$$

Recall measures the proportion of actual positives that were predicted correctly. Recall is given by:

$$Recall = \frac{TP}{TP + FN} \tag{3.27}$$

Precision and Recall metrics are useful when the dataset is imbalanced.

- F1 score: Indicates both the precision and robustness of a model and is defined as the harmonic mean of precision and recall. It is given by:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.28}$$

The F1 score metric is used when a dataset has imbalanced classes.

3.9 Signal Classification Methodology

The ability of neural networks to recognize and classify signals is highly dependent on the quality of the captured data. The accuracy of classification is affected by parameters such as the signal strength of the transmitter, SDR gain and location during data collection, as well as the date rate between a server and client. For example, it is straightforward to distinguish between Wi-Fi and LTE when a Wi-Fi's signal strength is greater than LTE's or vice versa. In general, due to the proximity of a wireless access point compared to an LTE base station, over-the-air recordings of Wi-Fi IQ data have a higher amplitude than LTE IQ data, and using the IQ data directly as an input to the neural network results in a high classification accuracy (greater than 85%). Additionally, it is straightforward to distinguish between two technologies when one of the captures include wireless data traffic and the other only has synchronization signals, such as a Wi-Fi beacon. In these instances, good classification results can be obtained without calculating and utilizing signal properties such as FFT, signal amplitude or phase features as an input to the neural networks. However, these examples represent only a small portion of any real-world scenario. In situations where the signal strengths of various signals are comparable and the recordings include wireless data traffic, which more accurately reflects real-world over-the-air signals, using IQ data alone is insufficient to achieve a good classification result, since the IQ data readings from various signal sources have similar values, and the neural networks struggle to differentiate the signals.

As discussed in Chapter 2, the majority of authors used either raw IQ data or spectrograms as input to their machine learning models. The authors in [66] demonstrated that using raw IQ data as the input to neural networks produced results similar to those obtained when using spectrograms. The authors in [65] demonstrated that calculating the FFT for the IQ data and utilizing it as an input resulted in improved classification accuracy. The authors demonstrated in [67] that using raw IQ data in conjunction with amplitude and phase values resulted in high accuracy. In this dissertation, the different feature engineering techniques were compared, and it was found that the input features used in [67] were better for classifying real-world data than the

other methods. However, the results were unsatisfactory because the classification accuracy was less than 80% in the majority of the classification scenarios and model training took a significant period of time. While providing the neural networks with signal features such as amplitude, phase, and FFT enhances classification accuracy, these features alone are insufficient for understanding the underlying pattern of signals from different wireless technologies. Hence, there is a need for novel approaches to process the IQ data that can extract the underlying pattern of a wireless signal.

3.9.1 Proposed IQ Feature Engineering

The goal of the proposed feature engineering process is to determine the underlying pattern of the IQ data. The method allows for a robust and reliable signal classification across a variety of real-world scenarios, including classifying signals with the same modulation schemes, similar signal strengths, similar wireless traffic, etc., for determining the wireless network technology using the IQ data of a signal. The proposed novel feature engineering for processing IQ data is as follows:

Algorithm 1 Novel Feature Engineering for IQ Data Processing

$i \leftarrow 0$
$j \leftarrow N$ $\quad$ *where*, $500 \leq N \leq 1000$
while $j \leq length\,of\,IQ$ **do**
  $FeatureA \leftarrow \mu(IQ[i] : IQ[j])$
  $FeatureB \leftarrow \sigma(IQ[i] : IQ[j])$
  $i \leftarrow i + N$
  $j \leftarrow j + N$
end while
$Feature\,1 \leftarrow Real(FeatureA)$
$Feature\,2 \leftarrow Imag(FeatureA)$
$Feature\,3 \leftarrow \sqrt{Real(FeatureA)^2 + Imag(FeatureA)^2}$
$Feature\,4 \leftarrow \tan^{-1}\left[\frac{Imag(FeatureA)}{Real(FeatureA)}]\right]$
$Feature\,5 \leftarrow FeatureB$

The proposed feature engineering approach greatly accelerates signal data processing by lowering the total number of samples in the data by a factor of N, while identifying signal variations by calculating the mean and standard deviation N samples at a time. N is a parameter that specifies the number of data points to consider when extracting the signal's pattern and aggregating the

values. In this case, where real-world signals are classified, it was determined through experimentation that a value of N between 500 and 1000 is optimal for the data recorded for the dissertation; values less than 500 or greater than 1000 did not produce as good a result because the mean and standard deviation calculations would be based on either too few or too many samples, respectively, resulting in an inaccurate representation of the underlying pattern.

3.9.2 Neural Network models

The data for training the models were captured using the Universal Software Radio Peripheral (USRP) B200mini-i SDR and include over-the-air IQ recordings of Wi-Fi (IEEE 802.11ax), LTE, and 5G. The signals were transmitted such that they had a common modulation scheme, 64 QAM, and were captured at a sampling rate of 20 million samples per second for a duration of 1 second each. The models were implemented in Python 3 using the Keras library, a high-level API for neural networks, with the TensorFlow framework in the back-end. The machine used for the implementation is the Dell Aurora R11 desktop, which has an Intel core i9-10900K CPU, NVIDIA RTX 3090 GPU, and 32 GB of RAM. The dataset was divided into training, validation and test datasets. Out of the 20 million samples captured per signal category, 70% were used for training, 15% for validation and 15% for testing the models. The DFNN, CNN, and LSTM models were developed to evaluate various prevalent NN architectures and determine the optimal architecture for signal classification, if any.

The NN architectures used in the dissertation for binary classification of signals, i.e., Wi-Fi vs. LTE, Wi-Fi vs. 5G, LTE vs. 5G, with the novel input features are summarized in Tables 3.1, 3.2, and 3.3.

Table 3.1: Deep Feedforward Network Architecture for Binary Classification

Layer	Output dimensions	Activation function
Input	(None, 5)	n/a
Dense	(None, 512)	ReLU
Dense	(None, 350)	ReLU
Dense	(None, 256)	ReLU
Dense	(None, 128)	ReLU
Dense	(None, 64)	ReLU
Dense	(None, 32)	ReLU
Dense	(None, 1)	Sigmoid

Optimizer: Adam

Loss function: Binary Crossentropy

Table 3.2: Deep Convolutional Network Architecture for Binary Classification

Layer	Output dimensions	Activation function
Input	(None, 5, 1)	n/a
Conv1D	(None, 4, 64)	ReLU
MaxPooling1D	(None, 2, 64)	ReLU
Conv1D	(None, 2, 48)	ReLU
Conv1D	(None, 2, 48)	ReLU
Flatten	(None, 96)	n/a
Dense	(None, 64)	ReLU
Dense	(None, 32)	ReLU
Dense	(None, 1)	Sigmoid

Optimizer: Adam

Padding: Same

Loss function: Binary Crossentropy

Table 3.3: Deep LSTM Network Architecture for Binary Classification

Layer	Output dimensions	Activation function
Input	(None, 1, 5)	n/a
LSTM	(None, 1, 256)	ReLU
LSTM	(None, 1, 128)	ReLU
LSTM	(None, 1, 64)	ReLU
LSTM	(None, 1, 32)	ReLU
LSTM	(None, 1, 16)	ReLU
Dense	(None, 1)	Sigmoid

Optimizer: Adam

Loss function: Binary Crossentropy

The NN architectures used in the dissertation for multi-class classification of signals, i.e., Wi-Fi vs. LTE vs. 5G, with the novel features as input are summarized in Tables 3.4, 3.5, and 3.6.

Table 3.4: Deep Feedforward Network Architecture for Multi-class Classification

Layer	Output dimensions	Activation function
Input	(None, 5)	n/a
Dense	(None, 512)	ReLU
Dense	(None, 350)	ReLU
Dense	(None, 256)	ReLU
Dense	(None, 128)	ReLU
Dense	(None, 64)	ReLU
Dense	(None, 32)	ReLU
Dense	(None, 3)	Softmax

Optimizer: Adam

Loss function: Sparse Categorical Crossentropy

Table 3.5: Deep Convolutional Network Architecture for Multi-class Classification

Layer	Output dimensions	Activation function
Input	(None, 5, 1)	n/a
Conv1D	(None, 4, 64)	ReLU
MaxPooling1D	(None, 2, 64)	ReLU
Conv1D	(None, 2, 48)	ReLU
Conv1D	(None, 2, 48)	ReLU
Flatten	(None, 96)	n/a
Dense	(None, 64)	ReLU
Dense	(None, 32)	ReLU
Dense	(None, 3)	Softmax

Optimizer: Adam

Padding: Same

Loss function: Sparse Categorical Crossentropy

Table 3.6: Deep LSTM Network Architecture for Multi-class Classification

Layer	Output dimensions	Activation function
Input	(None, 1, 5)	n/a
LSTM	(None, 1, 256)	ReLU
LSTM	(None, 1, 128)	ReLU
LSTM	(None, 1, 64)	ReLU
LSTM	(None, 1, 32)	ReLU
LSTM	(None, 1, 16)	ReLU
Dense	(None, 3)	Softmax

Optimizer: Adam

Loss function: Sparse Categorical Crossentropy

Chapter 4

Results

This chapter presents the results of signal classification using neural networks. The neural network models were trained and tested on IQ data collected over-the-air from Wi-Fi (IEEE 802.11ax), LTE, and 5G signals, and the IQ data were classified according to the wireless access technology. The Wi-Fi Access Point (AP) and LTE eNB hardware were configured and network clients, or User Equipment (UE), were connected to the radios to set up the Wi-Fi and LTE network connections. The test-beds were used for generating user traffic and capturing the Wi-Fi and LTE signals of the same modulation scheme over-the-air. Since 5G-NR radios are subjected to export control as of this writing and have access restrictions, the test UE was connected to a 5G network deployed to the general public by telecommunications service providers, such as T-Mobile, to generate user traffic and capture over-the-air 5G-NR signals. Figure 4.1 shows the generalized block diagram for capturing the real-world over-the-air signals. Appendix A contains detailed information about the data collection procedure.

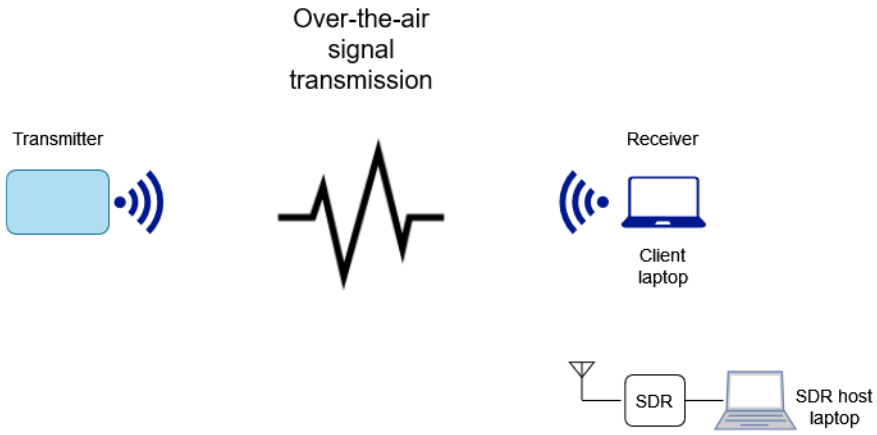

Figure 4.1: RF signal data capture overview

Two classification categories were used to evaluate the models: binary classification and multi-class classification. Given that feature engineering enabled improved classification performance than using only the IQ data, as demonstrated by the authors in [67], [18], and [65], the classification performance of the NN models when using the proposed novel feature engineering approach was compared to the performance when using one of the best existing feature engineering techniques: IQ data, amplitude, and phase features as input, as described in Section 3.9. It can be observed that using the proposed features reduces the training time significantly while increasing the classification accuracy due to the reasons mentioned in Section 3.9.1.

4.1 Binary Classification

In this category, the IQ data was classified between Wi-Fi and LTE, Wi-Fi and 5G, 5G and LTE, respectively, to evaluate the classification efficiency in a scenario where a spectrum is shared between two different types of wireless access technologies. The performance of the various NN models described in Section 3.9.2 were evaluated using the metrics described in Section 3.8 for the above three scenarios.

4.1.1 Deep Feedforward Network Performance

The performance of the DFNN model for the above three classification scenarios with the two input feature approaches are shown in Figures 4.2 and 4.3.

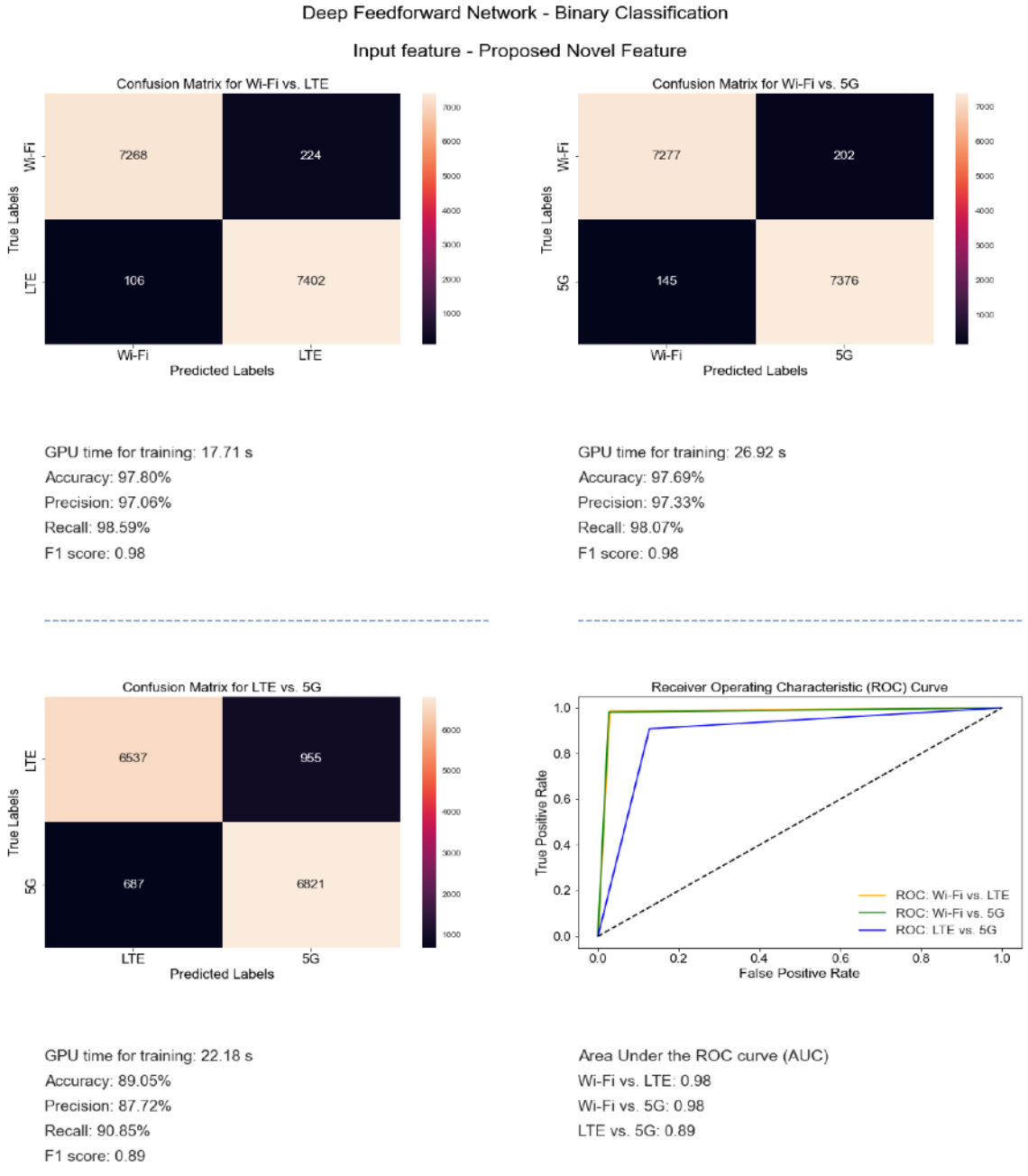

Figure 4.2: DFNN binary classification performance with the proposed novel input features

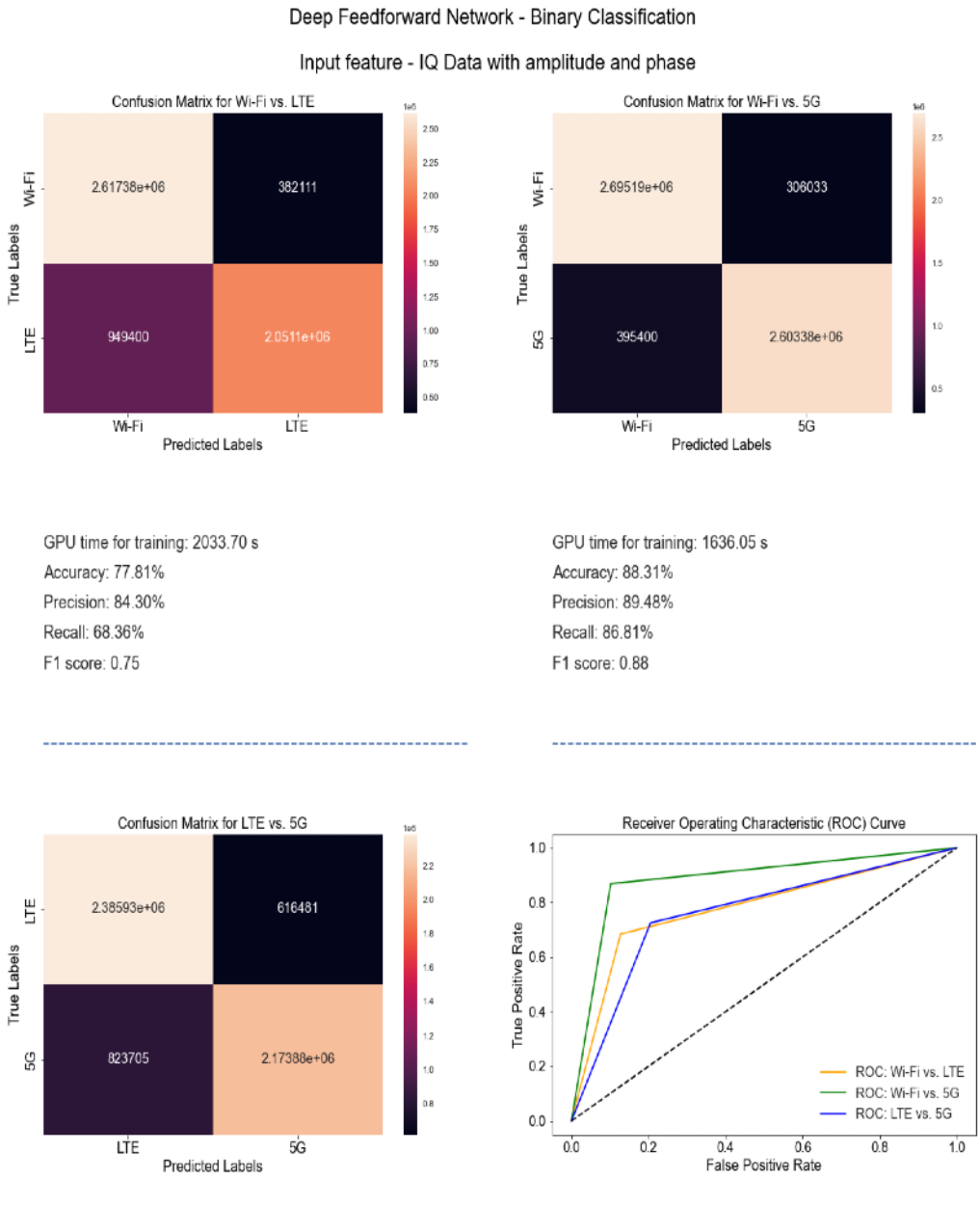

Figure 4.3: DFNN binary classification performance with I/Q, amplitude and phase input features

It can be observed from the above figures that the DFNN performance for signal classification is significantly higher when using the novel input features when compared to using IQ data with amplitude and phase features. With the proposed input features, an F1 score of 0.98 for both Wi-Fi vs. LTE and Wi-Fi vs. 5G classification and a score of 0.89 for LTE vs. 5G classification was achieved, whereas an F1 score of 0.75 for Wi-Fi vs. LTE and LTE vs. 5G classification and a score of 0.88 for Wi-Fi vs. 5G classification was achieved. The computational time needed for training the models using the proposed input features was under 30 seconds per scenario, whereas the model needed as high as 2847 seconds (47 minutes) per scenario for training when using the alternate input features.

4.1.2 Deep Convolutional Network Performance

The performance of the CNN model for the above three classification scenarios with the two input feature approaches are shown in figures 4.4 and 4.5.

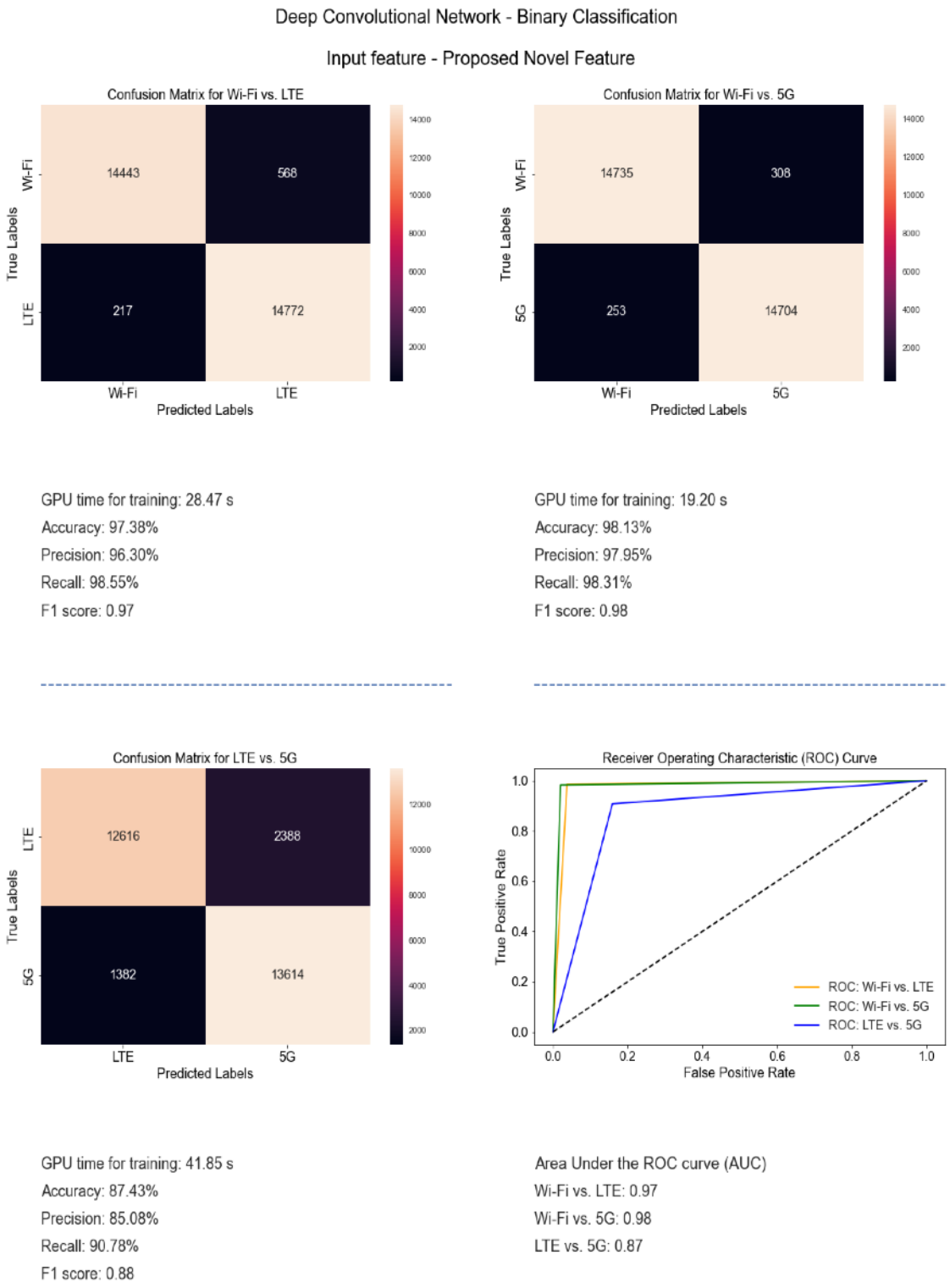

Figure 4.4: CNN binary classification performance with the proposed novel input features

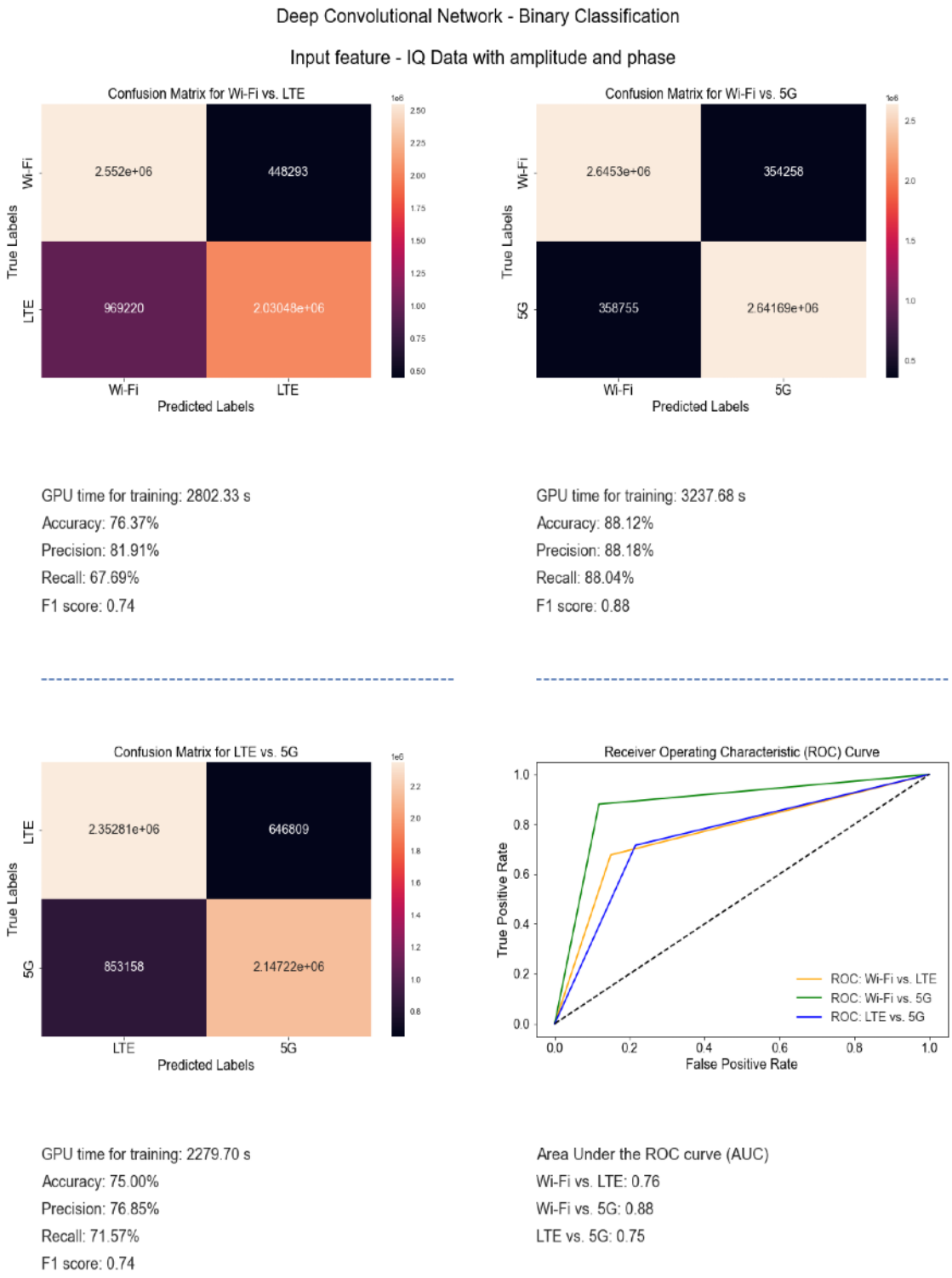

Figure 4.5: CNN binary classification performance with I/Q, amplitude and phase input features

It can be observed from the above figures that the CNN performance for signal classification is significantly higher when using the novel input features when compared to using IQ data with amplitude and phase features. With the proposed input features, an F1 score of 0.97 for Wi-Fi vs. LTE classification, 0.98 for Wi-Fi vs. 5G classification and a score of 0.88 for LTE vs. 5G classification was achieved, whereas an F1 score of 0.74 for Wi-Fi vs. LTE and LTE vs. 5G classification and a score of 0.88 for Wi-Fi vs. 5G classification was achieved. The computational time needed for training the models using the proposed input features was under 45 seconds per scenario, whereas the model needed as high as 3237 seconds (54 minutes) per scenario for training when using the alternate input features.

4.1.3 Deep LSTM Network Performance

The performance of the LSTM model for the above three classification scenarios with the two input feature approaches are shown in Figures 4.6 and 4.7.

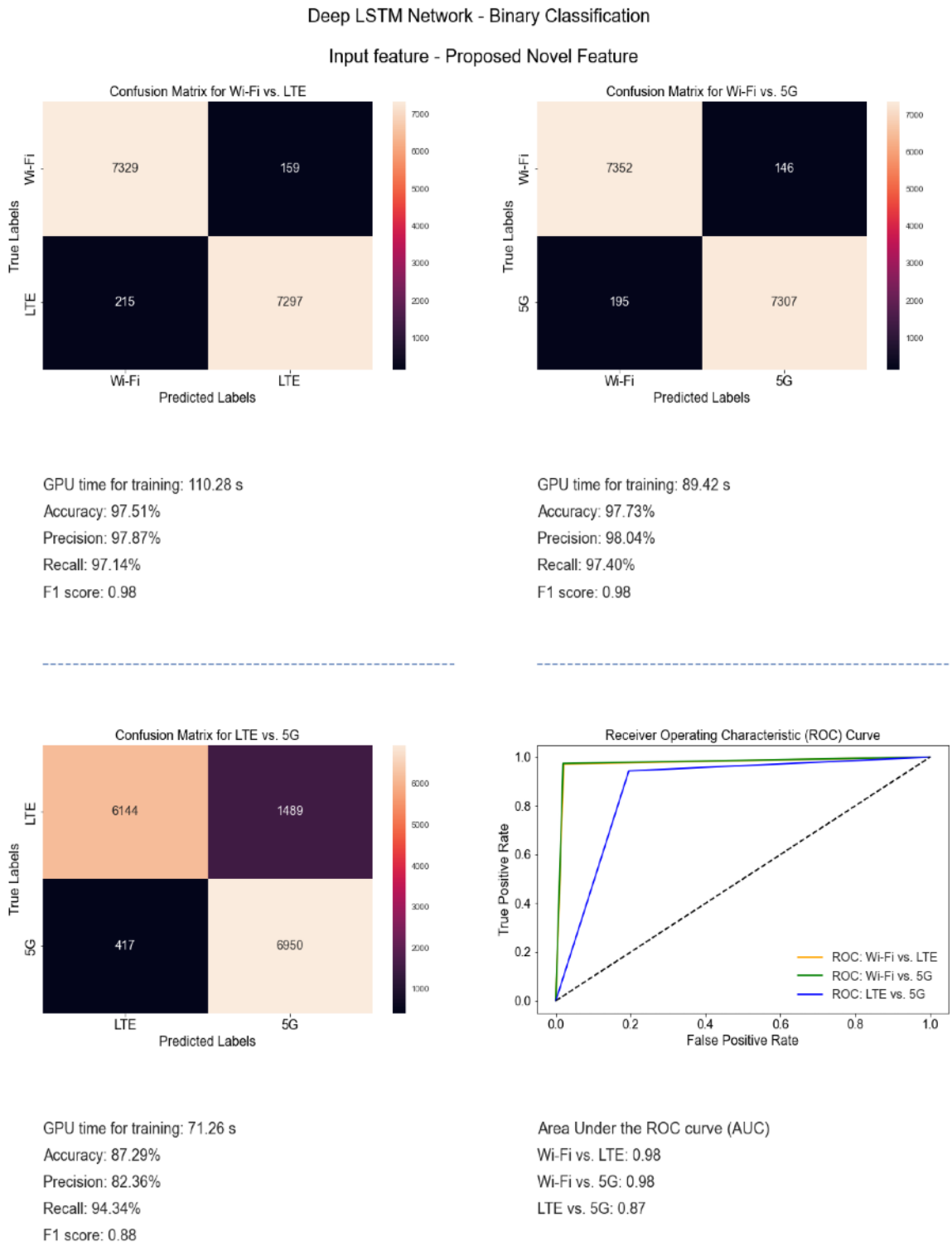

Figure 4.6: LSTM binary classification performance with the proposed novel input features

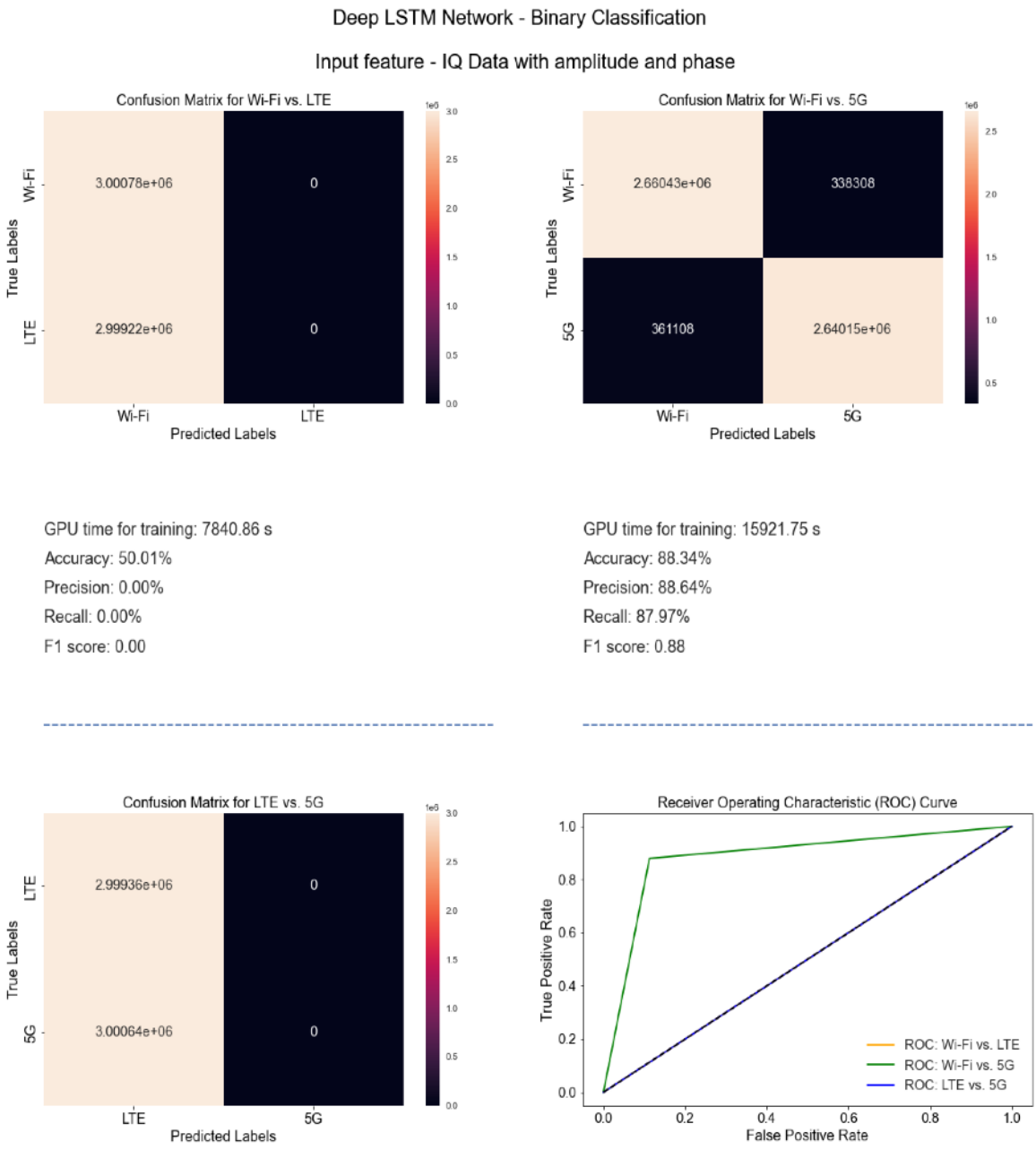

Figure 4.7: LSTM binary classification performance with I/Q, amplitude and phase input features

It can be observed from the above figures that the LSTM model was successfully able to classify between the signals in all the three scenarios when using the proposed input features. An F1 score of 0.98 for Wi-Fi vs. LTE and Wi-Fi vs. 5G classification and a score of 0.88 for LTE vs. 5G classification was achieved. The computational time needed for training the models using the proposed input features was under 45 seconds per scenario, whereas the model needed as high as 3237 seconds (54 minutes) per scenario for training when using the alternate input features. However, when using the IQ data with amplitude and phase features, it failed to classify the signals in the Wi-Fi vs. LTE and LTE vs. 5G scenario, and achieved an F1 score of 0.88 when classifying between Wi-Fi and 5G, and needed 15,921 seconds (265 minutes) to train.

In all of the cases above, classification performance was greater when the proposed input features were used, and the various NN architectures performed similarly, with DFNN being the best of the three models.

The results of the binary classifications for Wi-Fi vs. LTE performance are summarized in Table 4.1.

Table 4.1: Wi-Fi vs. LTE

Models	**Novel Input Features**					**I/Q with amplitude and phase features**				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**
DFNN	97.80%	97.06%	98.59%	0.98	0.98	77.81%	84.30%	68.36%	0.75	0.78
CNN	97.38%	96.30%	98.55%	0.97	0.97	76.37%	81.91%	67.69%	0.74	0.76
LSTM	97.51%	97.87%	97.14%	0.98	0.98	50.01%	0%	0%	0	0.5

The results of the binary classifications for Wi-Fi vs. 5G performance are summarized in Table 4.2.

Table 4.2: Wi-Fi vs. 5G

Models	Novel Input Features					I/Q with amplitude and phase features				
	Accuracy	Precision	Recall	F1 Score	AUC	Accuracy	Precision	Recall	F1 Score	AUC
DFNN	97.69%	97.33%	98.07%	0.98	0.98	88.31%	89.48%	86.81%	0.88	0.88
CNN	98.13%	97.95%	98.31%	0.98	0.98	88.12%	88.18%	88.04%	0.88	0.88
LSTM	97.73%	98.04%	97.40%	0.98	0.98	88.34%	88.64%	87.97%	0.88	0.88

The results of the binary classifications for LTE vs. 5G performance are summarized in Table 4.3.

Table 4.3: LTE vs. 5G

Models	Novel Input Features					I/Q with amplitude and phase features				
	Accuracy	Precision	Recall	F1 Score	AUC	Accuracy	Precision	Recall	F1 Score	AUC
DFNN	89.05%	87.72%	90.85%	0.89	0.89	76.0%	77.91%	72.52%	0.75	0.76
CNN	87.43%	85.08%	90.78%	0.88	0.87	75.0%	76.85%	71.57%	0.74	0.75
LSTM	87.29%	82.36%	94.34%	0.88	0.87	49.99%	0%	0%	0	0.5

4.2 Multi-class Classification

In this category, the IQ data was classified between Wi-Fi, LTE, and 5G, to evaluate the classification efficiency of the NN models, described in Section 3.9.2, in a scenario where a spectrum is shared between three widely used wireless access technologies. The performance were evaluated based on the metrics described in Section 3.8.

4.2.1 Deep Feedforward Network Performance

The performance of the DFNN model for multi-class classification with the two input feature approaches are shown in Figures 4.8 and 4.9.

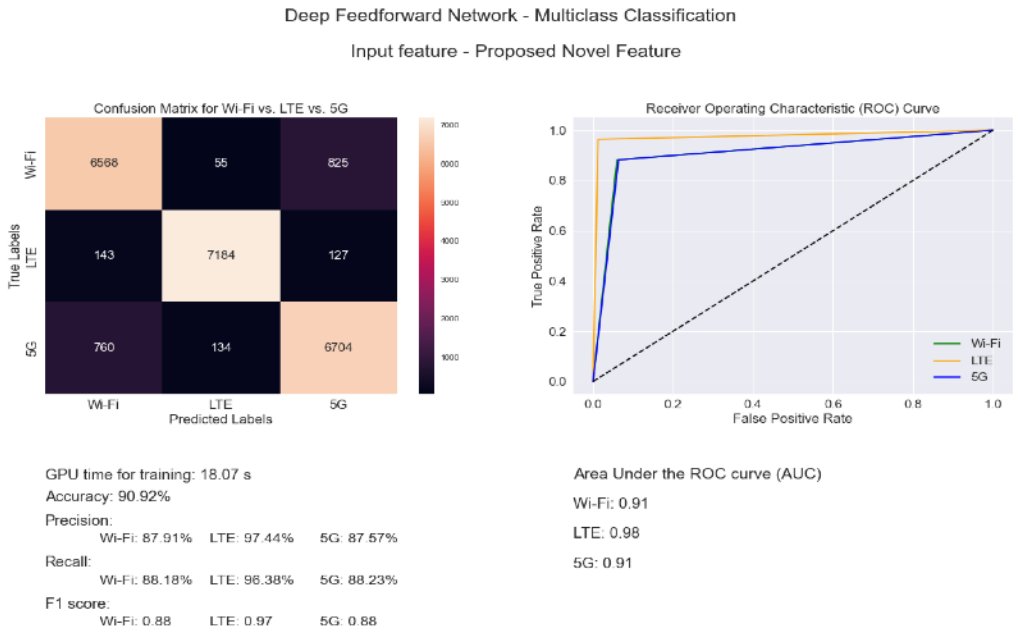

Figure 4.8: DFNN multi-class classification performance with the proposed novel input features

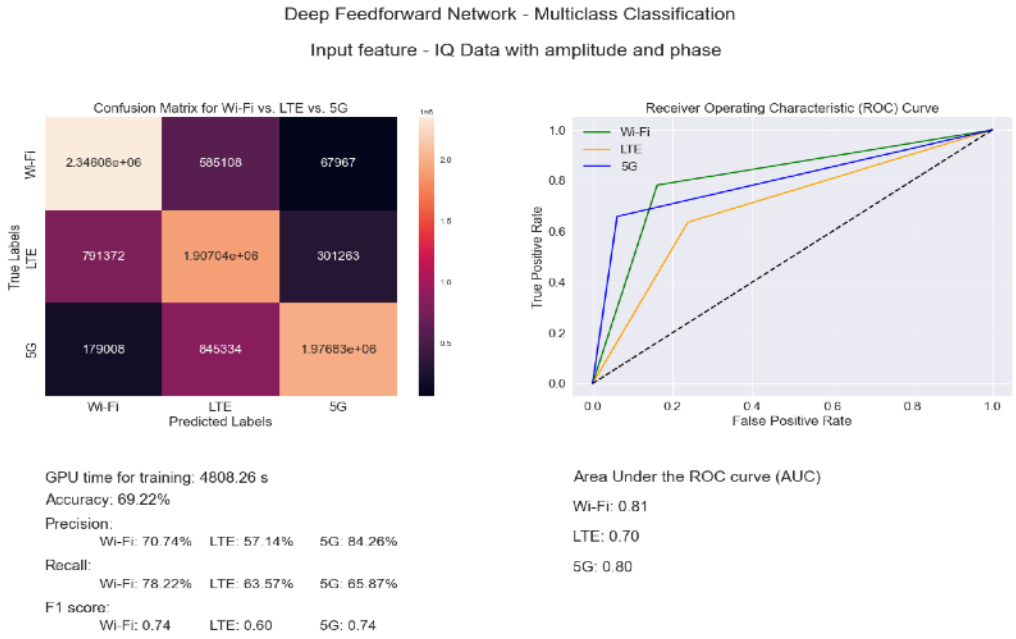

Figure 4.9: DFNN multi-class classification performance with I/Q, amplitude and phase input features

It can be observed that the DFNN model with the proposed input features significantly outperforms the model with I/Q, amplitude and phase features. F1 scores of 0.88, 0.97, 0.88 were obtained for Wi-Fi, LTE, 5G, respectively using the novel features, whereas scores of 0.74, 0.60, 0.74 were obtained for Wi-Fi, LTE, 5G, respectively when using the alternate features. The computational time required for training the model using the proposed input features was under 20 seconds, whereas the model needed as high as 4808 seconds (80 minutes) for training when using the alternate input features.

4.2.2 Deep Convolutional Network Performance

The performance of the CNN model for multi-class classification with the two input feature approaches are shown in Figures 4.10 and 4.11.

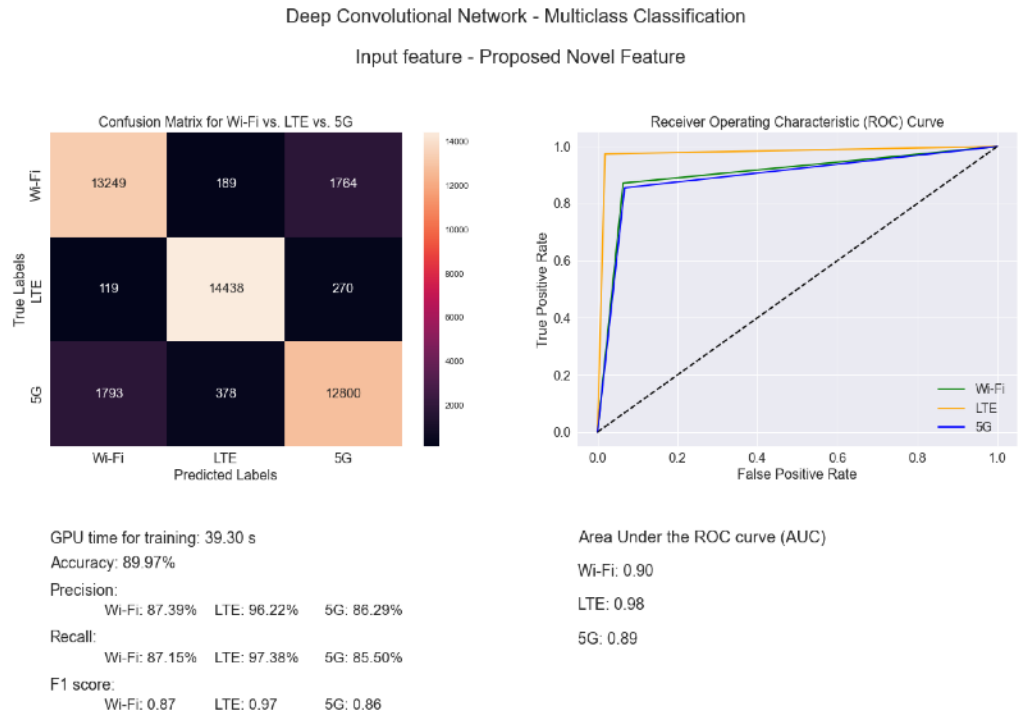

Figure 4.10: CNN multi-class classification performance with the proposed novel input features

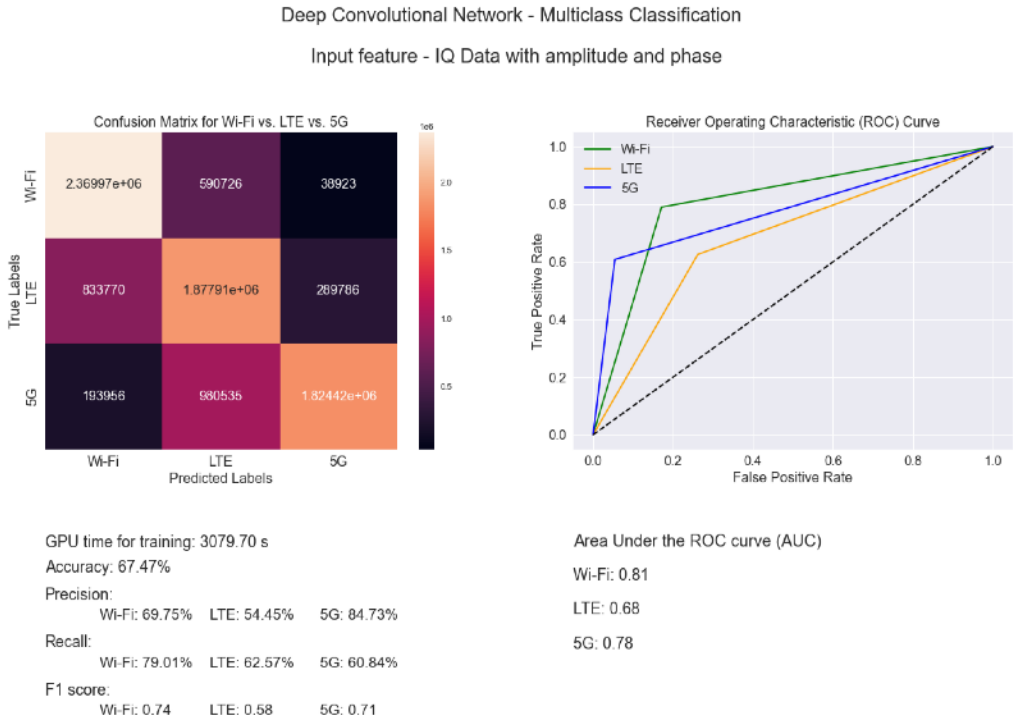

Figure 4.11: CNN multi-class classification performance with I/Q, amplitude and phase input features

It can be observed that the CNN model with the proposed input features significantly outperforms the model with I/Q, amplitude and phase features. F1 scores of 0.87, 0.97, 0.86 were obtained for Wi-Fi, LTE, 5G, respectively using the novel features, whereas scores of 0.74, 0.60, 0.74 were obtained for Wi-Fi, LTE, 5G, respectively when using the alternate features. The computational time required for training the model using the proposed input features was under 40 seconds, whereas the model needed as high as 3079 seconds (51 minutes) for training when using the alternate input features.

4.2.3 Deep LSTM Network Performance

The performance of the LSTM model for multi-class classification with the two input feature approaches are shown in Figures 4.12 and 4.13.

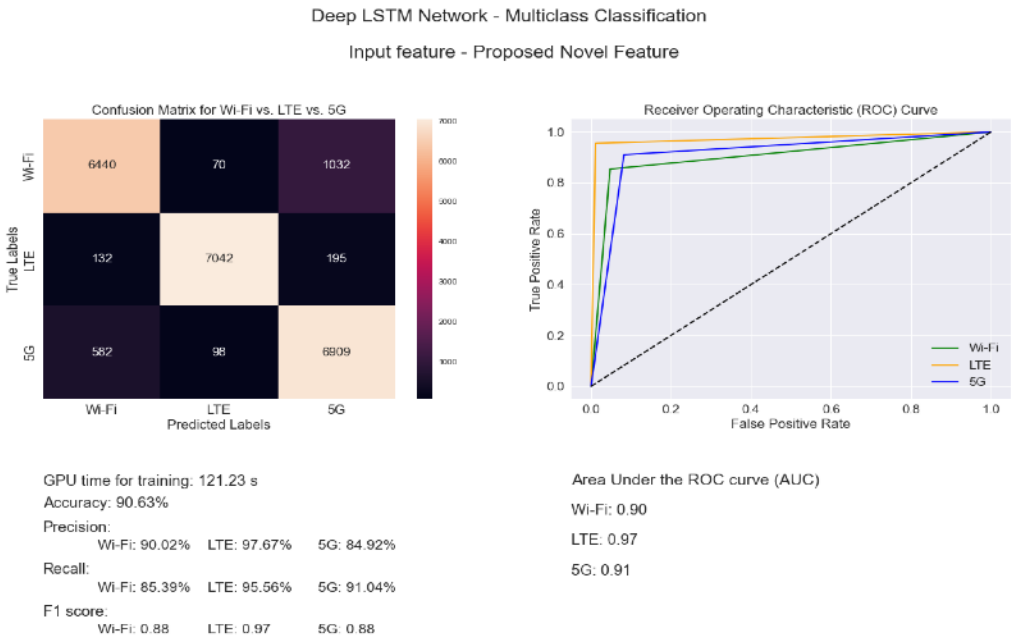

Figure 4.12: LSTM multi-class classification performance with the proposed novel input features

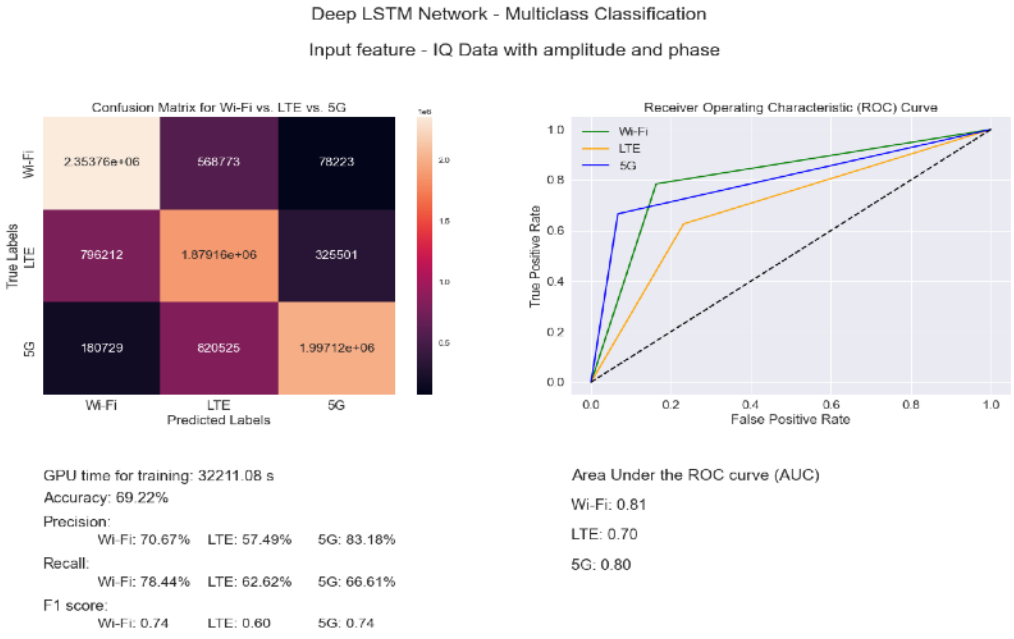

Figure 4.13: LSTM multi-class classification performance with I/Q, amplitude and phase input features

It can be observed that the LSTM model with the proposed input features significantly outperforms the model with I/Q, amplitude and phase features. F1 scores of 0.88, 0.97, 0.88 were obtained for Wi-Fi, LTE, 5G, respectively using the novel features, whereas scores of 0.74, 0.60, 0.74 were obtained for Wi-Fi, LTE, 5G, respectively when using the alternate features. The computational time required for training the model using the proposed input features was under 125 seconds, whereas the model needed as high as 32211 seconds (536 minutes) for training when using the alternate input features.

In all of the cases above, classification performance was greater when using the proposed input features, and the various NN models performed similarly, with DFNN being the best of the three models. Because the proposed input features significantly reduce training time, it is now feasible to train models on low-cost devices with limited computing capacity.

The results of the multi-class classifications are summarized in tables 4.4 and 4.5

Table 4.4: Wi-Fi vs. LTE vs. 5G using the novel input features

Models	**Novel Input Features**												
	Accuracy	**Precision**			**Recall**			**F1 Score**			**AUC**		
		Wi-Fi	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**
DFNN	90.92%	87.91%	97.44%	87.57%	88.18%	96.38%	88.23%	0.88	0.97	0.88	0.91	0.98	0.91
CNN	89.97%	87.39%	96.22%	86.29%	87.15%	97.38%	85.50%	0.87	0.97	0.86	0.90	0.98	0.89
LSTM	90.63%	90.02%	97.67%	84.92%	85.39%	95.56%	91.04%	0.88	0.97	0.88	0.90	0.97	0.91

Table 4.5: Wi-Fi vs. LTE vs. 5G using I/Q with amplitude and phase features

Models	**I/Q with amplitude and phase features**												
	Accuracy	**Precision**			**Recall**			**F1 Score**			**AUC**		
		Wi-Fi	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**	**Wi-Fi**	**LTE**	**5G**
DFNN	69.22%	70.74%	57.14%	84.26%	78.22%	63.57%	65.87%	0.74	0.60	0.74	0.81	0.70	0.80
CNN	67.47%	69.75%	54.45%	84.73%	79.01%	62.57%	60.84%	0.74	0.58	0.71	0.81	0.68	0.78
LSTM	69.22%	70.67%	57.49%	83.18%	78.44%	62.62%	66.61%	0.74	0.60	0.74	0.81	0.70	0.80

Chapter 5

Conclusions

The quality of the data collection is important for signal classification using IQ data. The comparisons revealed that the models with the proposed input features outperformed other input features significantly when classifying over-the-air Wi-Fi, LTE, and 5G signals.

Additionally, when the proposed input features were employed, the models were trained hundreds of times faster compared to previous methodologies. This allows the models to be trained on low-cost devices with limited computational power and supports model deployment in a dynamic RF environment. The models developed using the novel input features were highly consistent, as demonstrated by the high precision, recall, and F1 score values across all test scenarios.

Furthermore, it was demonstrated that the processing of the IQ data is critical for achieving high classification performance. Complex and advanced neural network architectures are not required to accurately classify IQ data; rather, a simple architecture such as the DFNN is sufficient when the suitable input features are provided.

5.1 Future Work

Binary classifications such as Wi-Fi vs. LTE and Wi-Fi vs. 5G achieved greater than 95% classification accuracy using the proposed feature engineering approach. However, classification of LTE vs. 5G achieved approximately 90% accuracy, which could be improved further.

Additional signal processing techniques can be developed to further improve classification performance, especially for multi-class classification. Future research can evaluate the effect of

various SNR levels on classification accuracy, especially for multi-class classification.

Another area of future research is determining whether the number of unique transmitters in a given area can be detected using IQ data and neural networks.

Bibliography

[1] **Cisco Annual Internet Report (2018-2023)**. Tech. rep. Accessed: 31 Jan 2022. `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf`: Cisco, 2020.

[2] Raza Umar et al. "Hybrid cooperative energy detection techniques in cognitive radio networks". In: **Handbook of Research on Software-Defined and Cognitive Radio Technologies for Dynamic Spectrum Management**. IGI Global, 2015, pp. 1–37.

[3] Fatima Salahdine and Hassan El Ghazi. "A real time spectrum scanning technique based on compressive sensing for cognitive radio networks". In: **2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)** (2017), pp. 506–511. ISSN: 153861104X.

[4] Joseph Mitola. "Cognitive radio architecture evolution". In: **Proceedings of the IEEE** 97.4 (2009), pp. 626–641. ISSN: 0018-9219.

[5] Notice of Proposed Rulemaking. **Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies**. Tech. rep. `https://docs.fcc.gov/public/attachments/FCC-03-322A1.pdf`: Federal Communications Commission, 2003.

[6] David Tse and Pramod Viswanath. **Fundamentals of wireless communication**. Cambridge University Press, 2005.

[7] Fadel F. Digham, M.-S. Alouini, and Marvin K. Simon. "On the energy detection of unknown signals over fading channels". In: **IEEE International Conference on Communications, 2003. ICC'03.** 5 (2003), pp. 3575–3579. ISSN: 0780378024.

[8] Y. Zeng, C. L. Koh, and Y.-C. Liang. "Maximum Eigenvalue Detection: Theory and Application". In: **2008 IEEE International Conference on Communications**. 2008, pp. 4160–4164. DOI: 10.1109/ICC.2008.781.

[9] Yonghong Zeng and Y.-C. Liang. "Eigenvalue-based spectrum sensing algorithms for cognitive radio". In: **IEEE transactions on communications** 57.6 (2009), pp. 1784–1793. ISSN: 0090-6778.

[10] Jarmo Lundén, Saleem A. Kassam, and Visa Koivunen. "Robust nonparametric cyclic correlation-based spectrum sensing for cognitive radio". In: **IEEE Transactions on Signal Processing** 58.1 (2009), pp. 38–52. ISSN: 1053-587X.

[11] Deborah Cohen and Yonina C Eldar. "Compressed cyclostationary detection for Cognitive Radio". In: **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. IEEE. 2017, pp. 3509–3513.

[12] Chenggang Jiang et al. "Statistical matched filter based robust spectrum sensing in noise uncertainty environment". In: **2012 IEEE 14th International Conference on Communication Technology**. IEEE. 2012, pp. 1209–1213.

[13] Xinzhi Zhang, Rong Chai, and Feifei Gao. "Matched filter based spectrum sensing and power level detection for cognitive radio network". In: **2014 IEEE global conference on signal and information processing (GlobalSIP)** (2014), pp. 1267–1270. ISSN: 1479970883.

[14] Hongjian Sun et al. "Wideband spectrum sensing for cognitive radio networks: a survey". In: **IEEE Wireless Communications** 20.2 (2013), pp. 74–81. ISSN: 1536-1284.

[15] Georgiana Weihe et al. **Passive and Active Spectrum Sharing (PASS)**. 2021. URL: https://scholar.colorado.edu/concern/datasets/q811kk83q (visited on 12/25/2021).

[16] **Spectrum Sharing**. NIST. Nov. 2019. URL: https://www.nist.gov/topics/advanced-communications/spectrum-sharing (visited on 12/25/2021).

[17] Wireless Innovation Forum. **Passive and Active Spectrum Sharing**. 2021. URL: https://winnf.memberclicks.net/assets/Proceedings/2021Virtual/Protection%20of%20Passive%20Services%20%20WInnComm%202021.pdf (visited on 12/25/2021).

[18] André Rosete. "Robust Spectrum Sensing and Signal Classification with Software-Defined Radios". English. PhD thesis. University of Colorado Boulder, 2020.

[19] **Facilitating Shared Use in the 3100-3550 MHz Band**. Tech. rep. https://docs.fcc.gov/public/attachments/FCC-20-138A1.pdf: Federal Communications Commission, Dec. 2015.

[20] Federal Communications Commission. **White Space**. URL: https://www.fcc.gov/general/white-space (visited on 01/31/2022).

[21] Adriana B. Flores et al. "IEEE 802.11 af: A standard for TV white space spectrum sharing". In: **IEEE Communications Magazine** 51.10 (2013), pp. 92–100.

[22] Carlos de M Cordeiro et al. "IEEE 802.22: An introduction to the first wireless standard based on cognitive radios." In: **J. Commun.** 1.1 (2006), pp. 38–47.

[23] **Unlicensed White Space Device Operations in the Television Bands**. Tech. rep. https://docs.fcc.gov/public/attachments/FCC-20-156A1_Rcd.pdf: Federal Communications Commission, Oct. 2020.

[24] Seshagiri Krishnamoorthy et al. "Characterization of the 2.4 GHz ISM band electromagnetic interference in a hospital environment". In: **Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)** 4 (2003), pp. 3245–3248. ISSN: 0780377893.

[25] Lorenzo Mucchi and Alessio Carpini. "Aggregate interference in ISM band: WBANs need cognitivity?" In: **2014 9th international conference on cognitive radio oriented wireless networks and communications (CROWNCOM)**. IEEE. 2014, pp. 247–253.

[26] Hongwei Huo et al. "Coexistence issues of 2.4 GHz sensor networks with other RF devices at home". In: **2009 Third International Conference on Sensor Technologies and Applications** (2009), pp. 200–205. ISSN: 0769536697.

[27] N. Azmi et al. "Interference issues and mitigation method in WSN 2.4 GHz ISM band: A survey". In: **2014 2nd International Conference on Electronic Design (ICED)** (2014), pp. 403–408. ISSN: 1479961035.

[28] Ad Kamerman and Nedim Erkocevic. "Microwave oven interference on wireless LANs operating in the 2.4 GHz ISM band". In: **Proceedings of 8th International Symposium on Personal, Indoor and Mobile Radio Communications-PIMRC'97** 3 (1997), pp. 1221–1227. ISSN: 0780338715.

[29] Chung-hao Chen, Pujitha Davuluri, and Dong-ho Han. "A novel measurement fixture for characterizing USB 3.0 radio frequency interference". In: **2013 IEEE International Symposium on Electromagnetic Compatibility** (2013), pp. 768–772. ISSN: 1479904104.

[30] Axel Sikora and Voicu F. Groza. "Coexistence of IEEE 802.15.4 with other Systems in the 2.4 GHz-ISM-Band". In: **2005 IEEE Instrumentationand Measurement Technology Conference Proceedings** 3 (2005), pp. 1786–1791. ISSN: 0780388798.

[31] **2021 Market Update**. Tech. rep. `https://www.bluetooth.com/wp-content/uploads/2021/01/2021-Bluetooth_Market_Update.pdf`: Bluetooth Special Interest Group, Jan. 2021.

[32] Moawiah Alhulayil and Miguel Lopez-Benitez. "LTE/Wi-Fi coexistence in unlicensed bands based on dynamic transmission opportunity". In: **2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)** (2020), pp. 1–6. ISSN: 1728151783.

[33] Joey Jackson. **CES 2015: Qualcomm takes on Wi-Fi with LTE-U**. Jan. 2015. URL: https://www.rcrwireless.com/20150106/network-infrastructure/ces-2015-qualcomm-demonstrates-lte-u-tag20 (visited on 12/20/2021).

[34] Hwan-Joon Kwon et al. "Licensed-assisted access to unlicensed spectrum in LTE release 13". In: **IEEE communications magazine** 55.2 (2016), pp. 201–207.

[35] Deepak R. Joshi, Dimitrie C. Popescu, and Octavia A. Dobre. "Adaptive spectrum sensing with noise variance estimation for dynamic cognitive radio systems". In: **2010 44th Annual Conference on Information Sciences and Systems (CISS)**. 2010, pp. 1–5. DOI: 10.1109/CISS.2010.5464913.

[36] Daniela Mercedes Martínez Plata and Ángel Gabriel Andrade Reátiga. "Evaluation of energy detection for spectrum sensing based on the dynamic selection of detection-threshold". In: **Procedia Engineering** 35 (2012), pp. 135–143. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2012.04.174. URL: https://www.sciencedirect.com/science/article/pii/S1877705812018097.

[37] Yonghong Zeng and Ying-Chang Liang. "Spectrum-sensing algorithms for cognitive radio based on statistical covariances". In: **IEEE transactions on Vehicular Technology** 58.4 (2008), pp. 1804–1815.

[38] Simon Haykin, David J Thomson, and Jeffrey H Reed. "Spectrum sensing for cognitive radio". In: **Proceedings of the IEEE** 97.5 (2009), pp. 849–877.

[39] F. F. Liedtke. "Computer simulation of an automatic classification procedure for digitally modulated communication signals with unknown parameters". In: **Signal Processing** 6.4 (1984), pp. 311–323. ISSN: 0165-1684. DOI: 10.1016/0165-1684(84)90063-X. URL: https://www.sciencedirect.com/science/article/pii/016516848490063X.

[40] J Aisbett. **Automatic modulation recognition using time domain parameters**. Tech. rep. ELECTRONICS RESEARCH LAB ADELAIDE (AUSTRALIA), 1986.

[41] Elsayed Elsayed Azzouz and Asoke K. Nandi. "Automatic identification of digital modulation types". In: **Signal Processing** 47.1 (1995), pp. 55–69. ISSN: 0165-1684.

[42] Chung-Yu Huan and Andreas Polydoros. "Likelihood methods for MPSK modulation classification". In: **IEEE Transactions on Communications** 43.2/3/4 (1995), pp. 1493–1504. ISSN: 0090-6778.

[43] Youness Arjoune and Naima Kaabouch. "On Spectrum Sensing, a Machine Learning Method for Cognitive Radio Systems". In: **2019 IEEE International Conference on Electro Information Technology (EIT)**. 2019, pp. 333–338. DOI: 10.1109/EIT.2019.8834099.

[44] A. Fehske, J. Gaeddert, and Jeffrey H. Reed. "A new approach to signal classification using spectral correlation and neural networks". In: **First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.** (2005), pp. 144–150. ISSN: 1424400139.

[45] Steven Hong et al. "Multi-user signal classification via spectral correlation". In: **2010 7th IEEE Consumer Communications and Networking Conference** (2010), pp. 1–5. ISSN: 1424451752.

[46] Karaputugala Madushan Thilina et al. "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks". In: **IEEE Journal on Selected Areas in Communications** 31.11 (2013), pp. 2209–2221. DOI: 10.1109/JSAC.2013.131120.

[47] Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. "Convolutional radio modulation recognition networks". In: **International conference on engineering applications of neural networks** (2016), pp. 213–226.

[48] Timothy James O'Shea, Tamoghna Roy, and T. Charles Clancy. "Over-the-air deep learning based radio signal classification". In: **IEEE Journal of Selected Topics in Signal Processing** 12.1 (2018), pp. 168–179. ISSN: 1932-4553.

[49] Daniel Czech, Amit Mishra, and Michael Inggs. "A CNN and LSTM-based approach to classifying transient radio frequency interference". In: **Astronomy and computing** 25 (2018), pp. 52–57. ISSN: 2213-1337.

[50] Rundong Li, Jianhao Hu, and Shuyuan Yang. "Deep Gated Recurrent Unit Convolution Network for Radio Signal Recognition". In: **2019 IEEE 19th International Conference on Communication Technology (ICCT)** (2019), pp. 159–163. ISSN: 1728105358.

[51] Jiabao Gao et al. "Deep Learning for Spectrum Sensing". In: **IEEE Wireless Communications Letters** 8.6 (2019), pp. 1727–1730. DOI: 10.1109/LWC.2019.2939314.

[52] Peixiang Cai, Yu Zhang, and Changyong Pan. "Coordination Graph-Based Deep Reinforcement Learning for Cooperative Spectrum Sensing Under Correlated Fading". In: **IEEE Wireless Communications Letters** 9.10 (2020), pp. 1778–1781. DOI: 10.1109/LWC.2020.3004687.

[53] Zhenjiang Shi et al. "Machine Learning-Enabled Cooperative Spectrum Sensing for Non-Orthogonal Multiple Access". In: **IEEE Transactions on Wireless Communications** 19.9 (2020), pp. 5692–5702. DOI: 10.1109/TWC.2020.2995594.

[54] Youheng Tan and Xiaojun Jing. "Cooperative Spectrum Sensing Based on Convolutional Neural Networks". In: **Applied Sciences** 11.10 (2021), p. 4440.

[55] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: **Neural computation** 1.4 (1989), pp. 541–551. ISSN: 0899-7667.

[56] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: **Advances in neural information processing systems** 25 (2012).

[57] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: **arXiv preprint arXiv:1409.1556** (2014).

[58] Shirin Aghabeiki et al. "Machine-learning-based spectrum sensing enhancement for software-defined radio applications". In: **2021 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)**. 2021, pp. 1–6. DOI: 10.1109/CCAAW50069.2021.9527294.

[59] D. Halverson and G. Wise. "Asymptotic memoryless discrete-time detection of phi-mixing signals in phi-mixing noise (Corresp.)" In: **IEEE Transactions on Information Theory** 30.2 (1984), pp. 415–417. DOI: 10.1109/TIT.1984.1056863.

[60] Zhibo Chen et al. "Deep STFT-CNN for Spectrum Sensing in Cognitive Radio". In: **IEEE Communications Letters** 25.3 (2021), pp. 864–868. DOI: 10.1109/LCOMM.2020.3037273.

[61] Simone Grimaldi, Aamir Mahmood, and Mikael Gidlund. "Real-time interference identification via supervised learning: Embedding coexistence awareness in IoT devices". In: **IEEE Access** 7 (2018), pp. 835–850. ISSN: 2169-3536.

[62] Malte Schmidt, Dimitri Block, and Uwe Meier. "Wireless interference identification with convolutional neural networks". In: **2017 IEEE 15th International Conference on Industrial Informatics (INDIN)** (2017), pp. 180–185. ISSN: 1538608375.

[63] Naim Bitar, Siraj Muhammad, and Hazem H. Refai. "Wireless technology identification using deep convolutional neural networks". In: **2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)** (2017), pp. 1–6. ISSN: 1538635313.

[64] Samer A. Rajab et al. "Energy detection and machine learning for the identification of wireless MAC technologies". In: **2015 international wireless communications and mobile computing conference (IWCMC)** (2015), pp. 1440–1446. ISSN: 147995344X.

[65] Vasilis Maglogiannis et al. "Enhancing the coexistence of LTE and Wi-Fi in unlicensed spectrum through convolutional neural networks". In: **IEEE Access** 7 (2019), pp. 28464–28477. ISSN: 2169-3536.

[66] Jaron Fontaine et al. "Towards low-complexity wireless technology classification across multiple environments". In: **Ad Hoc Networks** 91 (2019), p. 101881. ISSN: 1570-8705.

[67] S. Subray, S. Tschimben, and K. Gifford. "Towards Enhancing Spectrum Sensing: Signal Classification Using Autoencoders". In: **IEEE Access** 9 (2021), pp. 82288–82299. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3087113.

[68] **USRP B200mini-i**. URL: https://www.ettus.com/all-products/usrp-b200mini-i-2/ (visited on 12/10/2021).

[69] Fateme Talebi. **An Introduction to I/Q signals**. May 2020. URL: https://www.skyradar.com/blog/an-introduction-to-i/q-signals (visited on 12/10/2021).

[70] Alan Wolke. **What's Your IQ – About Quadrature Signals...** June 2015. URL: https://www.tek.com/en/blog/quadrature-iq-signals-explained (visited on 12/11/2021).

[71] Marc Lichtman. **PySDR: A Guide to SDR and DSP using Python**. URL: https://pysdr.org/content/sampling.html (visited on 12/12/2021).

[72] C.E. Shannon. "Communication in the Presence of Noise". In: **Proceedings of the IRE** 37.1 (1949), pp. 10–21. DOI: 10.1109/JRPROC.1949.232969.

[73] **Mathematical Basics of Band-Limited Samplingand Aliasing**. Tech. rep. https://pdfserv.maximintegrated.com/en/an/TUT3628.pdf: Maxim Integrated, Sept. 2005.

[74] **Fundamentals of Precision ADC Noise Analysis**. Tech. rep. https://www.ti.com/lit/eb/slyy192/slyy192.pdf?ts=1633541027864: Texas Instruments, Sept. 2020.

[75] Charles Boncelet. "Chapter 7 - Image Noise Models". In: **The Essential Guide to Image Processing**. Ed. by Al Bovik. Boston: Academic Press, 2009, pp. 143–167. ISBN: 978-0-12-374457-9. DOI: https://doi.org/10.1016/B978-0-12-374457-9.00007-X.

[76] V. Vapnik and A. Ya Lerner. "Recognition of patterns with help of generalized portraits". In: **Avtomat. i Telemekh** 24.6 (1963), pp. 774–780.

[77] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: **Machine learning** 20.3 (1995), pp. 273–297. ISSN: 0885-6125.

[78] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". In: **Psychological review** 65.6 (1958), p. 386.

[79] Jun Han and Claudio Moraga. "The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning". In: **Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation**. IWANN '96. Berlin, Heidelberg: Springer-Verlag, 1995, pp. 195–201. ISBN: 3540594973.

[80] Christopher M Bishop. **Pattern recognition and machine learning**. springer, 2006.

[81] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep sparse rectifier neural networks". In: **Proceedings of the fourteenth international conference on artificial intelligence and statistics** (2011), pp. 315–323.

[82] Fjodor Van Veen and Stefan Leijnen. **The Neural Network Zoo**. URL: `https://www.asimovinstitute.org/neural-network-zoo/` (visited on 07/17/2020).

[83] Aleksei Grigorevich. Ivakhnenko et al. **Cybernetic predicting devices**. English. New York: CCM Information Corp., 1965.

[84] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: **Nature** 323.6088 (1986), pp. 533–536. ISSN: 1476-4687.

[85] Haskell B Curry. "The method of steepest descent for non-linear minimization problems". In: **Quarterly of Applied Mathematics** 2.3 (1944), pp. 258–261.

[86] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: **Neural computation** 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667.

[87] Ettus Research. **USRP Hardware Driver and USRP Manual**. (Visited on 12/15/2021).

Appendix A

Data Collection Procedure

A.0.1 Determining optimal SDR gain value

One of the main sources of noise in an SDR is the Analog-to-Digital converter (ADC). The noise from the ADC can be categorized into thermal and quantization noise [74]. The USRP B200mini-i has a 12-bit ADC, however, all the IQ samples from the USRP are sent over the USB interface as complex 16-bit signed integers (16-bit for I and Q, respectively) [87], corresponding to a maximum available IQ data points of 2^{15} on either side of the zero-value as one bit is used for the sign. The SDR gain value was chosen to ensure a good dynamic range of the input signal and avoid clipping of potentially higher-powered signals and hence, the gain value selected should ensure that the captured signed integer was close to a 13-bit depth recording, i.e., 2-bits lower than the available 15-bits (1 bit is used by the sign), and the maximum IQ data point was approximately a 2^{13} signed integer value, i.e. 8192. The gain on the SDR was varied from the maximum value of 76 dB to the minimum value of 0 dB to identify the best value corresponding to 2^{13} positive signed integer value at both the SDRs and recording scenarios. Figure A.1 shows the variation in the maximum real and imaginary values of a recorded signal, as well as how many maximum values were captured on a 1e6 scale.

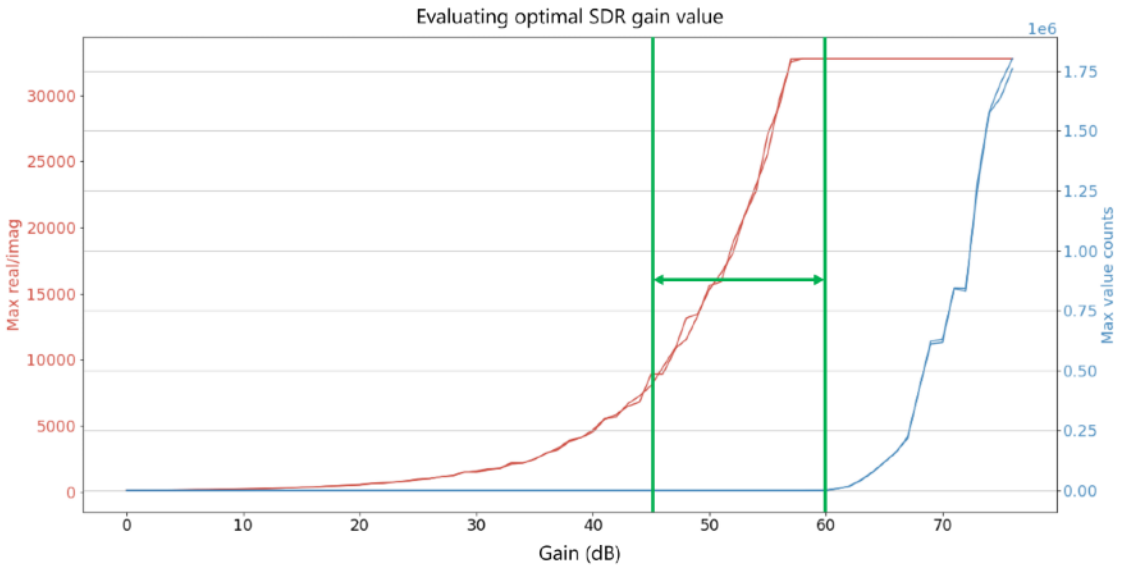

Figure A.1: Example of SDR gain selection for data capture

The ideal gain range is annotated as seen in Figure A.1. The number of maximum real and imaginary values of the captured signal reduced to 0, even though the maximum real and imaginary signed integer values were over 30,000 each, which corresponds to a bit-depth of 2^{15} for the received I/Q data, implying that only a few samples approached the maximum bit-depth. The gain value was selected as a value between the bit-depth of 2^{13} and the point at which the number of maximum real and imaginary values began to increase. Gain values outside the annotated range were excluded because they resulted in signal clipping or a reduction in the signal's dynamic range.

A.0.2 Wi-Fi signal data

A laptop was connected to an AP, and IEEE 802.11ax signals with Modulation and Coding Scheme (MCS) index of 7, which corresponds to 64-QAM modulation scheme with 5/6 coding, were captured as shown in Figure A.2. Table A.1 summarizes the hardware utilized to capture the data.

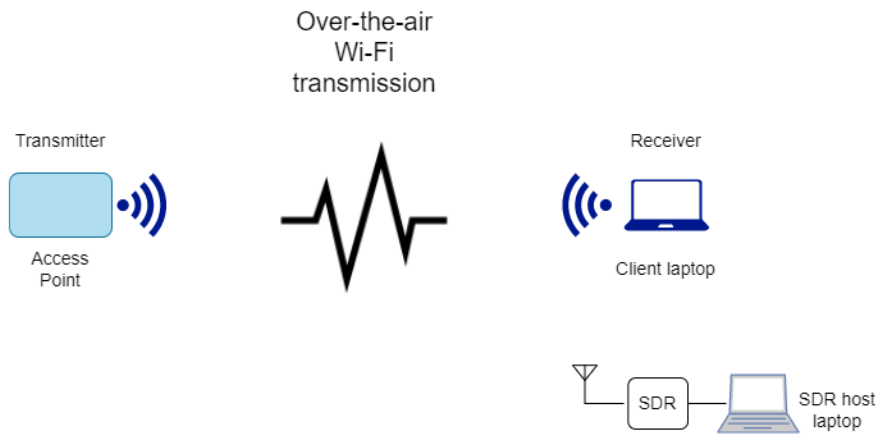

Figure A.2: Wi-Fi data capture overview

Table A.1: Hardware and software used for the Wi-Fi data collection

#	Name	Description
1	Extreme Networks AP650	Access Point
2	Macbook Air	Wi-Fi client
3	USRP B200mini-i	Software Defined Radio
4	VERT2450	Wi-Fi antenna with 3 dBi omni-directional gain
5	UHD Python API	Exposes UHD API to Python

The spectrogram and time-series plot of an IEEE 802.11ax signal are shown in Figure A.3 and Figure A.4, respectively.

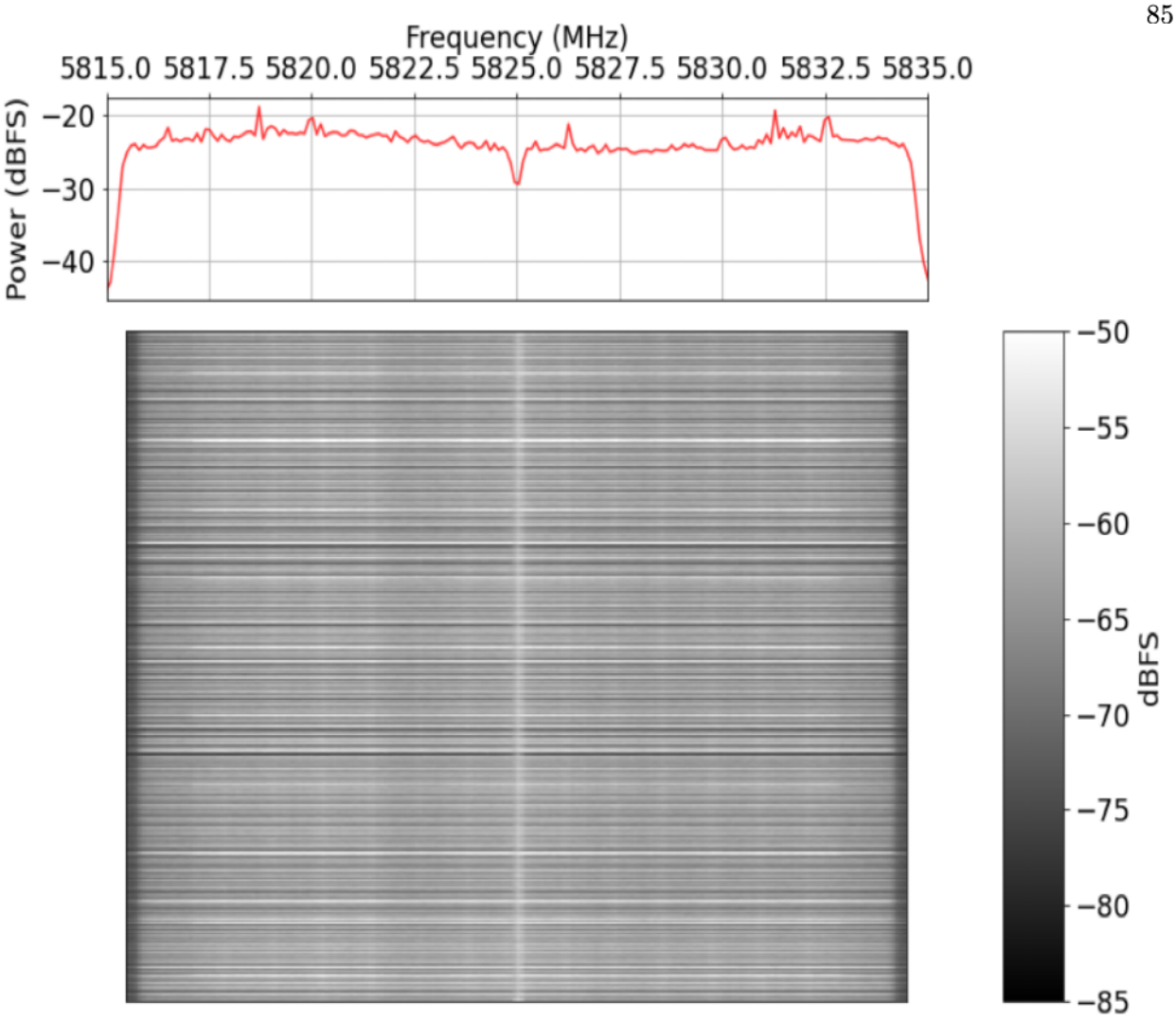

Figure A.3: Spectrogram of IEEE 802.11ax signal

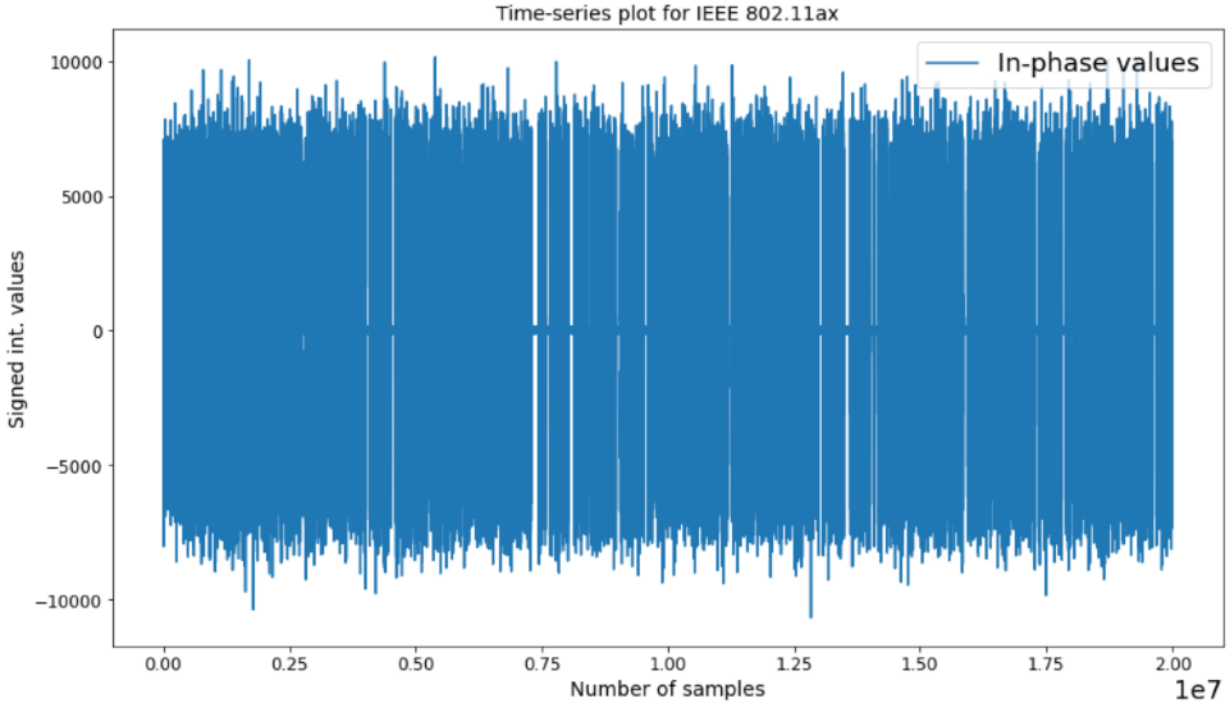

Figure A.4: Time-series plot of IEEE 802.11ax signal

The Table A.2 describes the parameters of the Wi-Fi signals that were captured, along with the SDR's gain and sampling rate used for the data collection.

Table A.2: Wi-Fi signal details

Parameter	**Value**
Frequency band	5815 - 5835 MHz
Protocol	IEEE 802.11ax
Transmission mode	TDD, Downlink
Modulation	MCS 7, 64 QAM
RSSI	-65 dBm
Bandwidth	20 MHz
SDR Sampling rate	20 MS/s
SDR Gain	50 dB

A.0.3 LTE signal data

A UE was connected to an LTE network and LTE data was captured such that the signal's SNR was greater than 20 dB, which corresponds to 64-QAM modulation scheme, was captured as shown in A.5. Table A.3 summarizes the hardware utilized to capture the data.

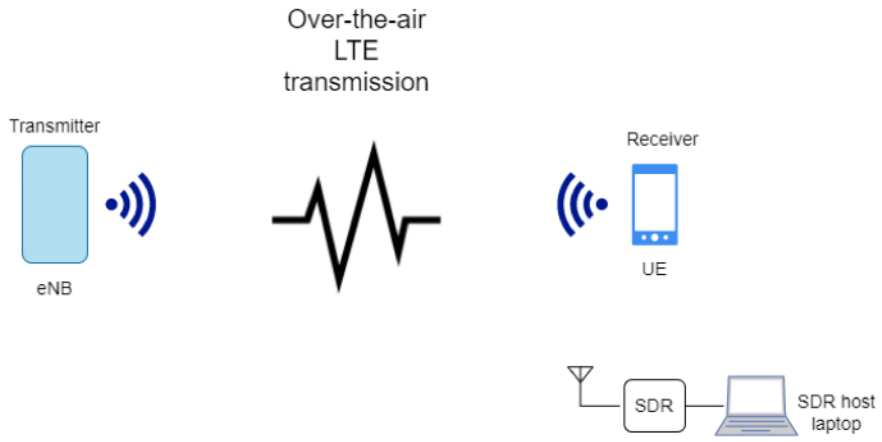

Figure A.5: LTE data capture overview

Table A.3: Hardware and software used for the LTE data collection

#	Name	Description
1	Nokia FWEA3 eNB	eNB
2	Samsung Galaxy S10e	LTE client
3	USRP B200mini-i	Software Defined Radio
4	ANT-LTE-WS-SMA	LTE dipole antenna with 5.9 dBi omni-directional gain
5	UHD Python API	Exposes UHD API to Python

The spectrogram and time-series plot of an LTE signal are shown in Figure A.6 and Figure A.7, respectively.

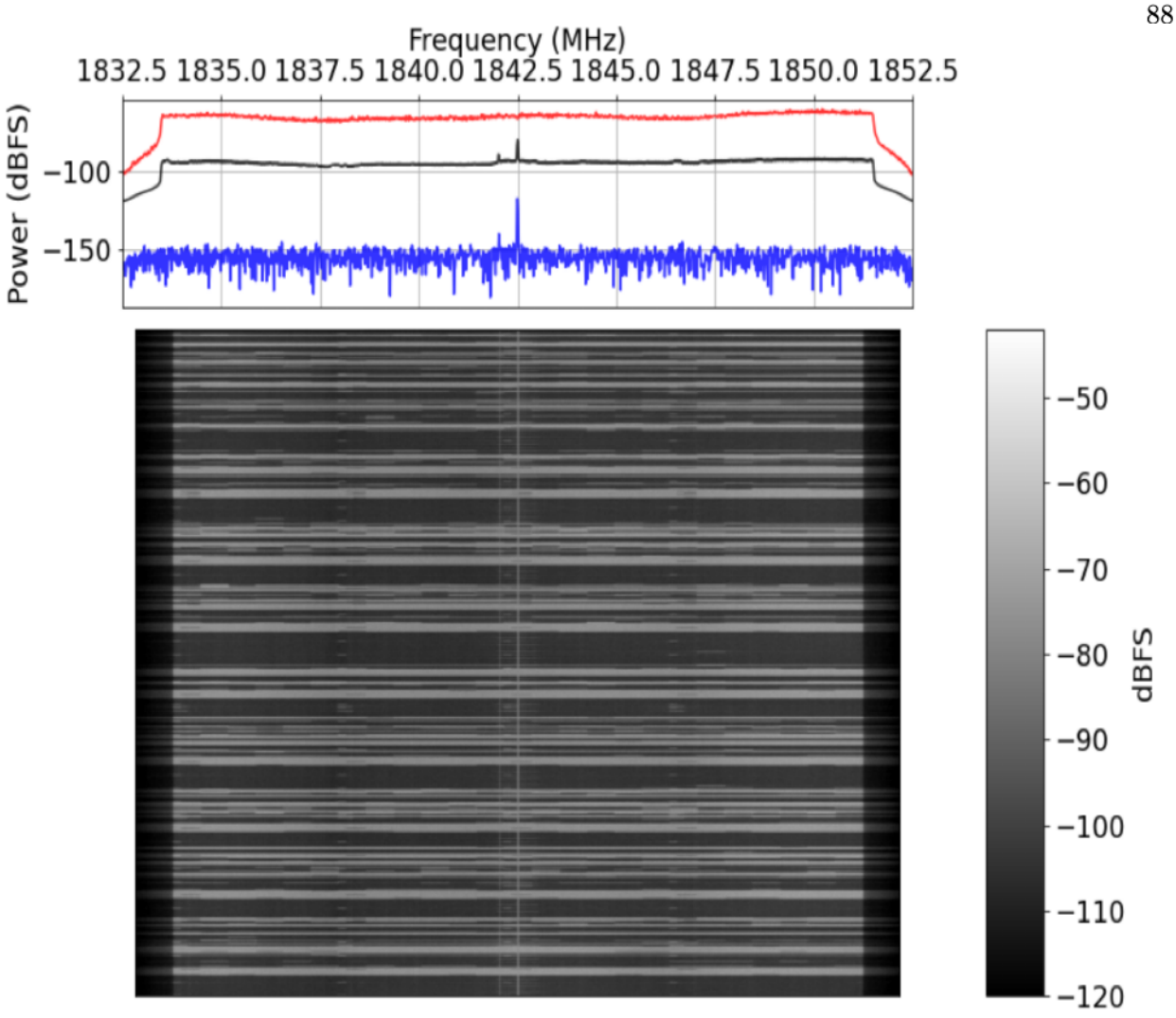

Figure A.6: Spectrogram of LTE signal

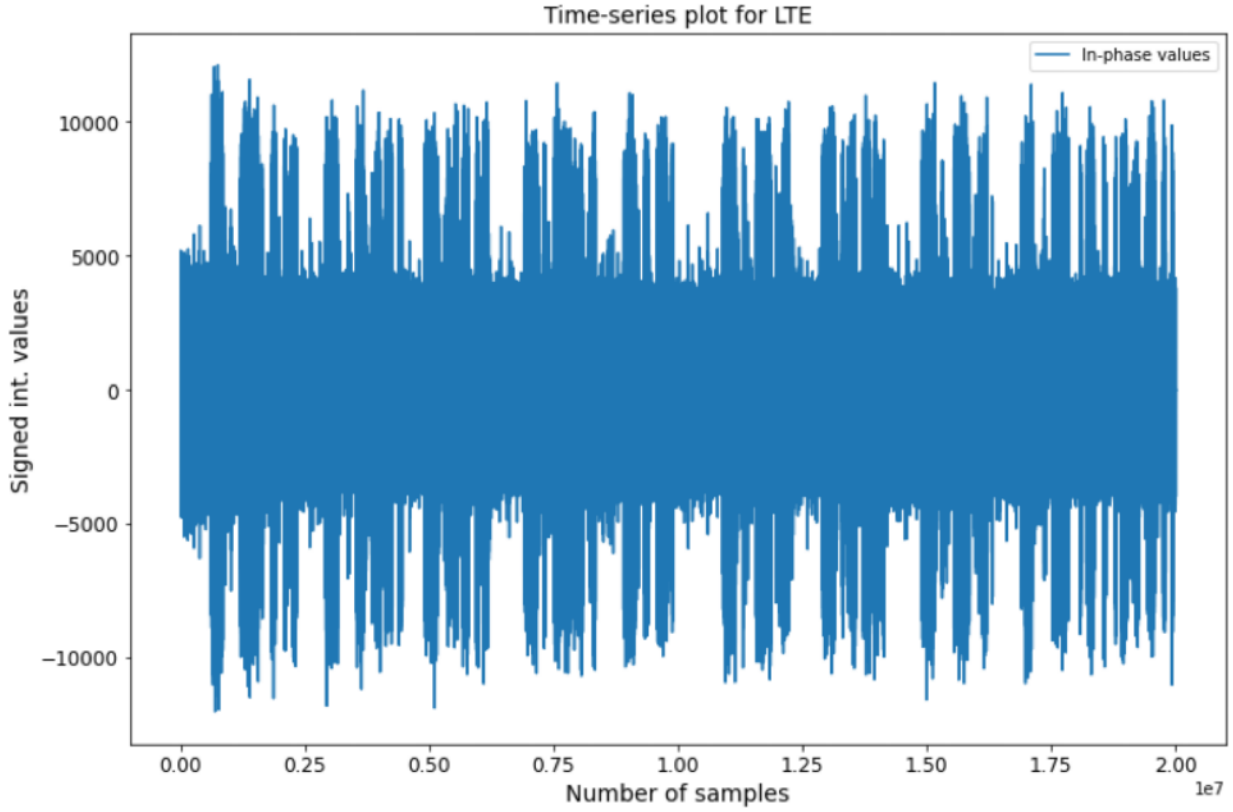

Figure A.7: Time-series plot of LTE signal

The Table A.4 describes the parameters of the LTE signal that was captured, along with the SDR's gain and sampling rate used for the data collection.

Table A.4: LTE signal details

Parameter	**Value**
Frequency band	1832.5 - 1852.5 MHz
Protocol	LTE
Transmission mode	FDD, Downlink
Modulation	64 QAM
SNR	30 dB
RSRP	-66 dBm
RSRQ	-9 dB
Bandwidth	20 MHz
SDR Sampling rate	20 MS/s
SDR Gain	12 dB

A.0.4 5G-NR signal data

The UE was connected to a 5G network that was deployed to the general public by telecommunications service provider, such as T-Mobile, and the 5G-NR signal data was collected over-the-air. The signal's SNR was approximately 18 dB. Table A.5 summarizes the hardware utilized to capture the data.

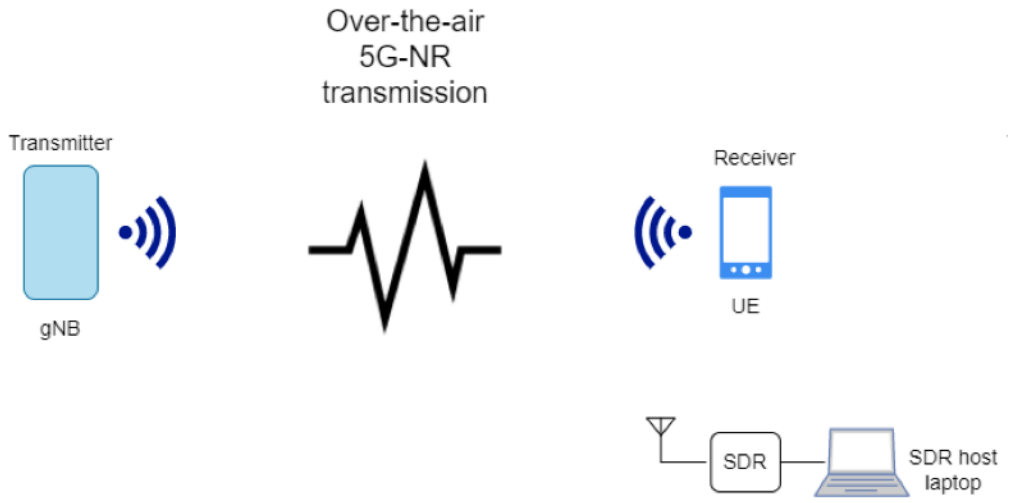

Figure A.8: 5G-NR data capture overview

Table A.5: Hardware and software used for the 5G-NR data collection

#	Name	Description
1	Google Pixel 5a	5G-NR client
2	USRP B200mini-i	Software Defined Radio
3	ANT-5GWWS1-SMA	5G low-band dipole antenna with 0.7 dBi omni-directional gain
4	UHD Python API	Exposes UHD API to Python

The spectrogram and time-series plot of a 5G-NR signal are shown in Figure A.9 and Figure A.10, respectively.

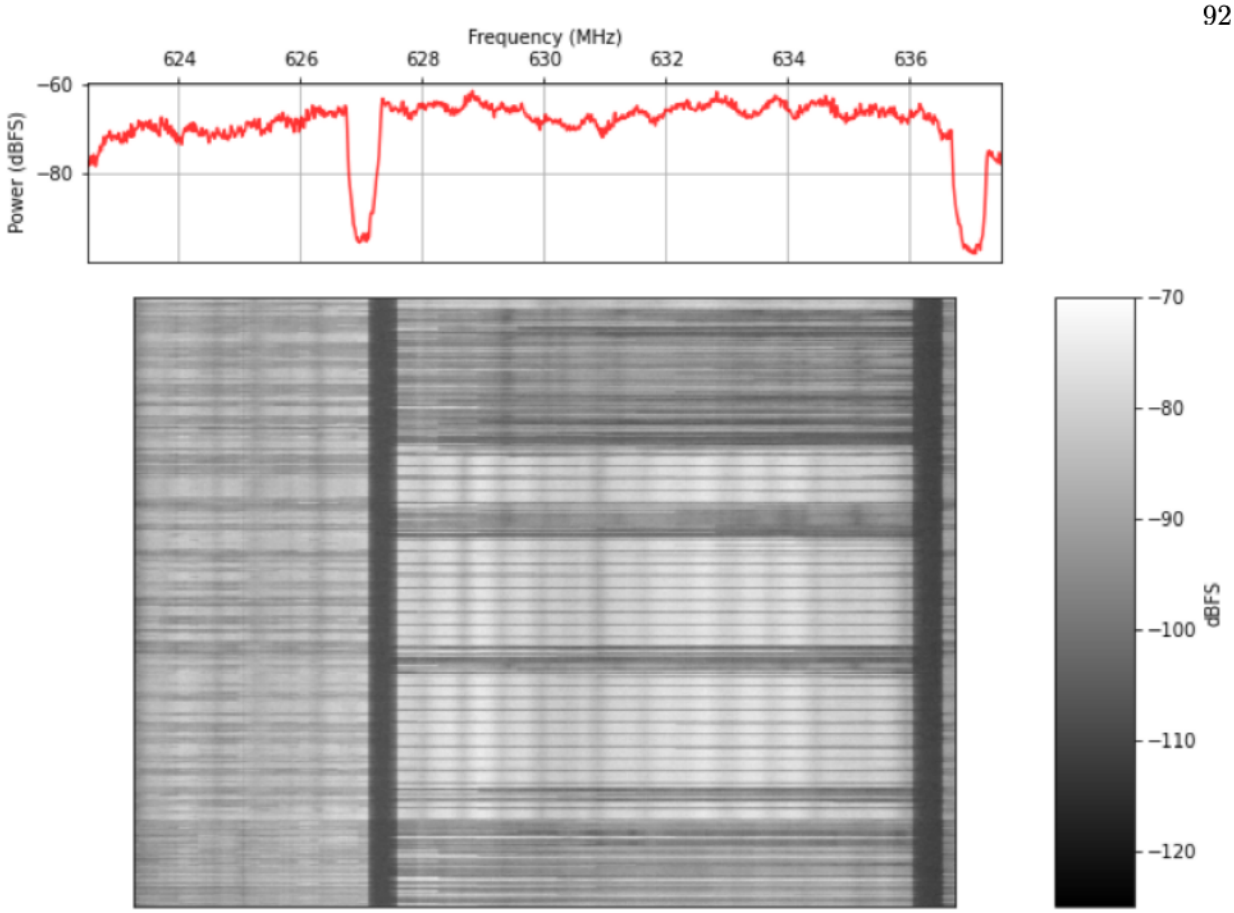

Figure A.9: Spectrogram of 5G-NR signal

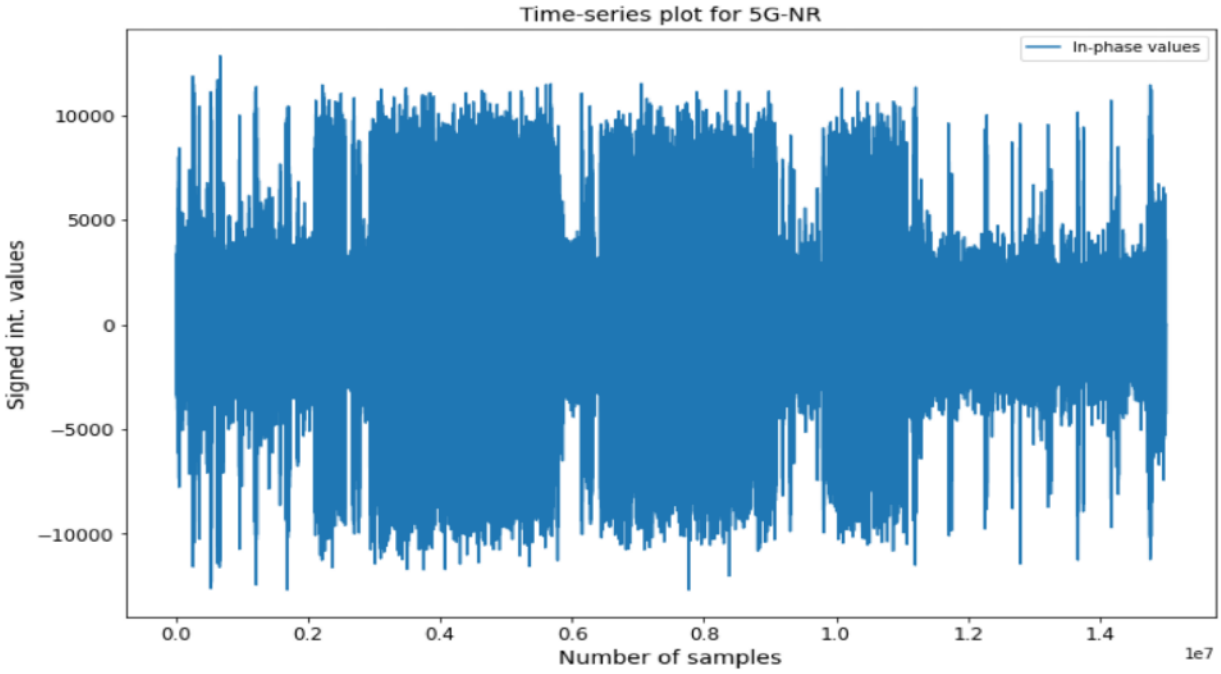

Figure A.10: Time-series plot of 5G-NR signal

The Table A.6 describes the parameters of the 5G-NR signal that was captured, along with the SDR's gain and sampling rate used for the data collection.

Table A.6: 5G-NR signal details

Parameter	**Value**
Frequency band	622 - 637 MHz
Protocol	5G-NR
Transmission mode	FDD, Downlink
Modulation	16 QAM, 64 QAM
SNR	18 dB
RSRP	-86 dBm
RSRQ	-12 dB
Bandwidth	15 MHz
SDR Sampling rate	15 MS/s
SDR Gain	45 dB

Appendix B

List of Abbreviations

Acronyms

3GPP 3rd Generation Partnership Project. 1, 7, 16, 20

5G 5th Generation Technology. 1

5G-NR 5G New Radio. 7, 8

ADC Analog-to-Digital Converter. 23, 27, 29

ANN Artificial Neural Network. 34

AP Access Point. 53, 83

API Application Programming Interface. 23

ATC Air Traffic Control. 4

AUC Area under the ROC Curve. 14, 46

BeS Beacon Signalling. 3

BS Base Station. 4

C2 Command-and-Control. 3

CBRS Citizens Broadband Radio Service. 4

CNN Convolutional Neural Network. 13–15, 20, 21, 38, 49, 57, 66, 67

COTS Commercial off-the-Shelf. 16, 22

CR Cognitive Radio. 5

CSMA/CA Carrier-Sense Multiple-Access with Collision Avoidance. 7

CSS Cooperative Spectrum Sensing. 3, 14

DAG-SVM Directed Acyclic Graph - Support Vector Machine. 14

DCR Direct-Conversion Receiver. 26

DFNN Deep Feedforward Neural Network. 21, 37, 49, 55, 63, 64, 66, 69, 70

DFS Dynamic Frequency Selection. 7

DFT Discrete Fourier Transform. 31

DRF Distributed Random Forest. 14

DSA Dynamic Spectrum Access. 2

ED Energy Detection. 7

ESC Environmental Sensing Capability. 4

FCC Federal Communications Commission. 2, 4, 7, 11

FFT Fast Fourier Transform. 10, 16, 31, 48

FN False Negative. 45

FP False Positive. 45

FPGA Field Programmable Gate Array. 23

FPR False Positive Rate. 45

GBM Gradient Boosting Machine. 14

GL-DB Geo-location Databases. 3

GPS Global Positioning System. 3, 5

GRU Gated Recurrent Unit. 13, 42

I/Q In-phase and Quadrature. 8, 27

IoT Internet of Things. 2, 6

ISM Industrial, Scientific, and Medical. 6, 7

KNN K-nearest neighbor. 13, 33

LAA License-Assisted Access. 7

LBT Listen Before Talk. 3

LO Local Oscillator. 27

LSTM Long Short-Term Memory. 13, 21, 42, 49, 60, 67, 69

LTE Long-Term Evolution. 1, 7

LTE-U LTE-Unlicensed. 7

MCS Modulation and Coding Scheme. 83

ML Machine Learning. 8, 9, 12, 13, 22, 31, 34

MLP Multilayer Perceptron. 37

NIST National Institute of Standards and Technology. 2

NN Neural Network. 8, 9, 21, 22, 49, 51, 54, 63, 64, 69

PASS Passive and Active Spectrum Sharing. 3

PCB Printed Circuit Board. 29

PU Primary Users. 4, 10

QAM Quadrature Amplitude Modulation. 8, 22, 49, 83, 87

QPSK Quadrature Phase Shift Keying. 8, 15

ReLU Rectified Linear Unit. 39

ResNet Residual Neural Network. 13

RF Radio Frequency. 8, 20, 24

RFIC Radio-Frequency Integrated Circuit. 23

RNN Recurrent Neural Network. 21, 40

ROC Receiver Operating Characteristic. 14, 45, 46

RSSI Received Signal Strength Indicator. 16, 17, 20

SAS Spectrum Access System. 4

SCD Spectral Correlation Density. 13

SDR Software-Defined Radio. 9, 18, 22, 23, 49

SNR Signal-to-Noise ratio. 8, 12, 16, 71, 87, 90

SS Spectrum Sensing. 3

STFT Short-Time Fourier Transform. 14

SU Secondary Users. 4, 5, 10

SVD Singular Value Decomposition. 10

SVM Support Vector Machine. 13, 14, 16, 31

TN True Negative. 45

TP True Positive. 45

TPR True Positive Rate. 45

TVWS Television White Spaces. 5

U-NII Unlicensed National Information Infrastructure. 7

U.S. United States. 2

UE User Equipment. 53, 87, 90

UHD USRP Hardware Driver. 23

UHF Ultra High Frequency. 5

USB Universal Serial Bus. 23

USRP Universal Software Radio Peripheral. 49

VHF Very High Frequency. 5

WLAN Wireless Local Area Network. 5, 20

WPAN Wireless Personal Area Network. 5