Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Spectrum Occupancy Prediction Using Deep Learning Algorithms

By: Addisu Melkie Tafere

A Thesis Submitted to the Department of Electrical and Computer Engineering for the Partial Fulfillment of the Degree of Master of Science in Computer Engineering

Addis Ababa, Ethiopia

July 2024

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Spectrum Occupancy Prediction Using Deep Learning Algorithms

By: Addisu Melkie Tafere

A Thesis Submitted to the Department of Electrical and Computer Engineering for the Partial Fulfillment of the Degree of Master of Science in Computer Engineering

Advisor: Dr. Eng. Getachew Alemu

Addis Ababa, Ethiopia

July 2024

Board of Examiners

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

By: Addisu Melkie Tafere

This is to certify that the thesis prepared by Addisu Melkie, titled Spectrum Occupancy Prediction Using Deep Learning Algorithms and Submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering compiles with the regulations of the University and meets the accepted standards to originality and quality.

Approved by the Board of Examining Committee:

	Name	Signature
Dean, School of Electrical and Computer Engineering:	Dr. Bisrat Derebssa	____________
Advisor:	Dr.Eng. Getachew Alemu	_____________
Internal Examiner:	_______________________	______________
External Examiner:	_______________________	_____________

Addis Ababa, Ethiopia

July 2024

Declaration

I hereby declare that this MSc thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: Addisu Melkie

Signature:__________________

Date:__________________

This MSc thesis has been submitted for examination with my approval as a university advisor.

Dr.-Eng. Getachew Alemu

Signature:__________________

Date:__________________

DEDICATION

I would like to dedicate this thesis to my beloved family, especially to Yegna Enat and Yegna Abat, for their unreserved support and encouragement throughout my life.

Acknowledgments

First, I would like to thank Almighty God and his mother, Saint Virgin Mary with the herald angel San Gabriel, for helping me reach this milestone after so many ups and downs.

Very special gratitude goes out to all down to my advisor Dr.-Eng Getachew Alemu for the encouragement and support by giving me good comments and suggestions. I will not forget to remember his encouragement and their timely support and guidance until the completion of the study.

I would also like to thank Dr. Bisrat Derebssa, Dr. Fistum Assaminew, and their colleagues for their valuable support and all the difficulties they solved.

I am also thanking Dr. Ephrem T Bekel, Dr. Beneyam B Haile, and all Zeregaw Internet and Cloud Services Provider staff for their help in the data collection by devoting their time from helping in the equipment configuration and the material support.

I am also so grateful to thank my friends who were with me during the regional data collection time, especially Aron Woledu, Seifu Girma, Rediet Million, and Samson Takle for their great collaborations and encouragement.

I would like to express my heartfelt gratitude to my friends who were with me throughout my study. In particular, I want to extend a special thanks to Ashenafi Workie, who stood by me during many challenging times.

Finally, I would like to thank the Ethiopia Communication Authority and Ethio Telecom for their valuable support during the data collection time.

Thank you for all your encouragement, God will repay you for all the good things you have done for me.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
BP	Backpropagation
Bi-LSTM	Bidirectional Long Short-Term Memory
CDMA	Code Division Multiple Access
CFD	Cyclostationary Feature Detection
Conv-LSTM	Convolutional Long Short-Term Memory
COR	Channel Occupancy Rate
CR	Cognitive Radio
CRN	Cognitive Radio Network
CRIoT	Cognitive Radio Internet of Things
CRU	Cognitive Radio User
CSS	Cooperative Spectrum Sensing
DC	Duty Cycle
DL	Deep Learning
DNN	Deep Neural Network
ECA	Ethiopian Communication Authority
ED	Energy Detection
FCC	Federal Communication Commission
FSA	Fixed Spectrum Allocation
GA	Genetic Algorithm
GPU	General-Purpose Graphical Processing Units
GRU	Gated Recurrent Unit
GSM	Global System for Mobile Communication
HMM	Hidden Markov Model
HNN	Hybrid Neural Network
IMT	International Mobile Technology
IOT	Internet of Things
ISM	Industrial Scientific and Medical
KM	Killo Meter

LTE	Long Term Evolution
LMR	Land Mobile Radio
LSTM	Long Short-Term Memory
MAC	Media Access Control
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multilayer Perceptron
MIMO	Multiple Input Multiple Output
MAP	Mean Square Error
PSD	Power Spectral Density
PU	Primary User
RBF	Radial Basis Function
RF	Radio Frequency
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
QoS	Quality of Service
SDR	Software Defined Radio
SMS	Short Message Service
SNR	Signal-to-Noise Ratio
SS	Spectrum Sensors
SU	Spectrum Utilization
SUs	Secondary Users
TCI	Telecommunication Intelligence
TV	Television
UMTS	Universal Mobile Telecommunications Service
USA	United States of America
WLAN	Wireless Local Area Network

List of Acronyms

2G	Second Generation
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
ADAM	Adaptive Moment Estimation
dB	Decibel
ReLU	Rectified Linear Unit
mm	Millimeter
Wi-Fi	Wireless Fidelity

Abstract

The fixed spectrum allocation (FSA) policy causes a waste of valuable and limited natural resources because a significant portion of the spectrum allocated to users is unused. With the exponential growth of wireless devices and the continuous development of new technologies demanding more bandwidth, there is a significant spectrum shortage under current policies. Dynamic spectrum access (DSA) implemented in a cognitive radio network (CRN) is an emerging solution to meet the growing demand for spectrum that promises to improve spectrum utilization that enables secondary users (SUs) to utilize unused spectrum allocated to primary users (PUs). CRNs have capabilities for empowerment to spectrum sensing, decision-making, sharing, and mobility. Spectrum sharing gets spectrum usage patterns from spectrum occupancy prediction to determine the channel states as "idle" or "busy". This study has addressed all the limitations of the previous studies by implementing a comprehensive approach that encompasses reliable spectrum sensing, potential candidate spectrum band identification, long-term adaptive prediction modeling, and quantification of improvements achieved in the prediction model. The Long-Short Term Memory (LSTM) Deep Learning (DL) model was proposed as a solution for this study to address the challenge of capturing temporal dynamics in sequential inputs. The LSTM model leverages a gating mechanism to regulate information flow within the network, allowing it to learn and model long-term temporal dependencies effectively. The dataset used for this study was obtained from a real-world spectrum measurement by employing the Cyclostationary Feature Detection (CFD) approaches in the GSM900 mobile network uplink band, spanning a frequency range of 902.5 to 915 MHz over five consecutive days. The dataset comprises a total of 225,000 data points. The five-day spectrum measurement data analysis yields an average spectrum utilization of 20.47%. The proposed model has predicted the spectrum occupancy state for 5 hours ahead in the future with an accuracy of 99.45% improved the spectrum utilization from 20.47% to 98.28% and reduced the sensing energy to 29.39% compared to real-time sensing.

Keywords: Spectrum, Spectrum Occupancy, Dynamic Spectrum Access, Deep Learning, and Cognitive Radio

Chapter One

1. Introduction

1.1. Background of the Study

The Radio Frequency (RF) spectrum stands as a limited and valuable natural resource used for various wireless communication systems, encompassing voice radio, digital terrestrial television (DTT), mobile telephony, and mobile broadband (MBB), all of which use the spectrum to facilitate the transmission and reception of data[1]. The RF spectrum spans a wide range of electromagnetic waves that demonstrate a direct relationship with wavelength, each spectrum band has distinct advantages and disadvantages. Lower frequencies, for instance, can propagate over longer distances and exhibit superior penetration through building walls. This characteristic makes them well-suited for applications such as broadcasting in expansive geographic areas. On the other hand, higher frequencies offer advantages in microelectronic devices like cell phones. Their shorter wavelengths enable the use of proportionally smaller antennas, allowing these devices to transmit larger volumes of data. Table 1.1 provides an overview of the diverse bands within the RF spectrum and their corresponding applications [2] .

Table 1.1 RF Spectrum bands and their applications

Band	Frequency	Wavelength	Propagation	Application
VLF-Very Low Frequency	3-30 kHz	100-10 km	Ground	Long-range radio navigation
LF-Low Frequency	30-300 kHz	10-1 km	Ground	Radio beacons and navigational locators
MLF-Medium Frequency	300-3000 kHz	1000-100 m	Sky	AM radio
HF-High Frequency	3-30 MHz	100-10 m	Sky	Citizens band (CB), ship/aircraft
VHF-Very High Frequency	30-300 MHz	10-1 m	Sky and line-of-sight	VHF TV, FM radio
UHF-Ultra High Frequency	300-3000 MHz	100-10 cm	Sky and line-of-sight	UHF TV, cellular phones, paging, satellite
SHF-Super High Frequency	3-30 GHz	10-1 cm	Line-of-sight	Satellite
EHF-Extremely High Frequency	30-300 GHz	10-1 cm	Line-of-sight	Radar, satellite

As wireless technologies continue to advance, the effective management and allocation of the RF spectrum remain essential for sustaining the growth and reliability of these systems. The spectrum used for wireless communication systems is a limited resource that cannot be simultaneously utilized for different services. This is because the simultaneous use of the same spectrum channel by different services can lead to interference. Therefore, careful spectrum channel management is crucial to prevent interference. In recent years, the demand for spectrum has increased dramatically due to the exponential growth of wireless devices and the continuous development of new bandwidth-hungry technologies. This has resulted in a spectrum scarcity, further increasing its commercial value. Spectrum stakeholders must, therefore, develop adaptable strategies to manage and use the spectrum efficiently, meeting both current and future spectrum standards [1][2][3]. The current fixed spectrum allocation (FSA) policy is not addressing the growing demand for spectrum due to its rigid command-and-control approach, which assigns channels to a single user regardless of actual usage. This inefficient allocation can lead to the wastage of spectrum resources and a decrease in the quality of service. However, studies have shown that significant portions of the spectrum assigned to licensed users are unused, indicating the need for a more dynamic and responsive spectrum allocation policy [4] [5] [6].

Different communication technologies have used various techniques to overcome spectrum utilization challenges. Such as higher frequencies using millimeter waves(mmWave) and massive multiple input multiple outputs (MIMO) in fifth-generation (5G), network densification, and MIMO in fourth-generation (4G). However, all of these techniques have their drawbacks. For example, mmWave and densification can only cover short ranges, and MIMO-based techniques require more complex and power-hungry transmitters [7]. To meet the growing demand for the spectrum and to change the conventional course of spectrum allocation, researchers are exploring new solutions. Dynamic spectrum access (DSA) implemented in cognitive radio networks (CRNs) has now emerged as a solution to improve spectrum utilization and reduce spectrum waste by allowing secondary users (SUs) to share unused spectrum with primary users (PUs)[8] . Because the CRNs are comprised of two types of users i.e., PUs and SUs, where PUs have a higher priority than SUs in accessing the channels. The SUs logically divides the channels allocated to the PUs into slots. Within each slot, the SUs has to sense the PU channel slot and accordingly access the slot when idle. The idle slots are called spectrum holes or white spaces[4].

Spectrum sharing requires knowledge of spectrum usage patterns, which can be obtained through spectrum sensing. However, real-time spectrum sensing is considered unreliable due to its high energy and time consumption. Spectrum occupancy state prediction, which infers the future states of the spectrum channel, proactively forecasts these states and estimates the effective bandwidth in the next slot. This allows SUs to adjust their data rates in advance for improved spectrum sensing. Consequently, SUs can conserve energy and time by avoiding the busy portions of the spectrum and focusing on idle portions during sensing [4][8]. This demonstrates that spectrum occupancy state prediction is a key enabler for shared spectrum access in the DSA model, as it allows SUs to identify and access idle spectrum channels before they become busy [8][9][10]. SUs in CRNs search for idle spectrum channels to use temporarily. They are equipped with the cognitive ability to effectively implement the CR system, which performs the following cycle of functions: Sensing: to observe and sample spectral channels, Decision: to allocate suitable spectral holes, Sharing: to contend access with other secondary users, and Mobility: to evacuate the spectral hole when a PU is present[10].

1.2. Motivation of the Study

Spectrum is a limited resource essential for a wide range of private and national activities, faces escalated demand driven by the rapid expansion of novel services and innovative technologies. The escalating demand introduces challenges compelling stakeholders to proactively adjust the formulation of new spectrum allocation and access policies. In response to these dynamic trends, spectrum prediction has emerged as a hot research area, offering the potential to optimize spectrum utilization. Predicting how the spectrum will be utilized in the future allows stakeholders to anticipate and plan for evolving demands. In particular, SUs can access spectrum channels that are not currently accessed by PUs. This proactive approach not only enhances spectrum utilization but also mitigates interference between primary and secondary users. Therefore, spectrum prediction can empower regulatory bodies, service providers, and technology developers to make informed decisions regarding spectrum management. The rigidity of the command-and-control spectrum allocation policy was considered the root cause of the problems of spectrum scarcity and underutilization.

Motivated by the imperative to overcome these challenges in managing spectrum resources, this study focuses on the exploration of spectrum occupancy prediction. This study holds promise as it offers a potential solution to the problems posed by traditional allocation policies. Specifically, it empowers CRs to operate in underutilized spectrum bands, dynamically adapting to changing conditions. This study has developed an analytical model to evaluate the spectrum utilization of the candidate spectrum band and the spectrum utilization improvement obtained from spectrum sharing. The significance of this study lies in mitigating the underutilization of spectrum that can significantly degrade the quality of service in wireless communication. Policymakers stand to be motivated by the prospect of spectrum sharing, recognizing it as a viable and innovative solution for future CR deployments. Spectrum sharing promises to drive technological innovation and holds the potential to maximize economic benefits and improve overall connectivity [5] [6].

1.3. Statement of the Problem

The statements of the problems in this study could be considered as four issues. The first is from the spectrum sensing methods used to identify the state of the PU, whereas the second one is the long-term spectrum occupancy state prediction i.e., the degree of predictability concerned with the increased length of future predictions, and identifying the potential candidate spectrum band for implementing a CR and quantifying the improvements achieved in the spectrum occupancy state prediction model.

There is a parametric sensing approach that works based on some prior information about the states of PU activity. In many real-world applications, the lack of prior information about the PU activity has led to a preference for nonparametric sensing methods, with Energy Detection (ED) being a common choice due to its low computational complexity and ease of implementation [11]. However, the wireless environment introduces challenges such as fading and hidden node problems, resulting in an exponential decay of field strength over transmission in ED. This drawback complicates the effective selection of threshold values, particularly when the signal-noise ratio (SNR) is very low, making ED inefficient and susceptible to interference when used to implement CRs [12][13][14].

In contrast, the parametric sensing approach known as Cyclostationary Feature Detection (CFD), offers advantages over non-parametric sensing approaches because it is not affected by noise. The CFD sensing approach leverages the spectral correlation of cyclostationary signals, a property absents in noise. This characteristic enables the CFD sensing approach to operate effectively in regions of low SNR and provides better performance even when prior information about the PUs signal is known. Cyclostationary feature detection exhibits superior performance and demonstrates robustness to noise uncertainties in low SNR environments [14] [15].

Traditional statistical models and machine learning algorithms assume spectrum occupancy states as stationary processes, implying that they remain constant over time and are suitable for short-term predictions [6] [16]. Artificial Neural Networks (ANNs) are less effective for modeling temporal data due to the absence of memory elements. Recurrent Neural Networks (RNNs) have been employed for such tasks; however, they face challenges, such as the vanishing gradient problem, hindering their ability to capture long-term dependencies [8] [11]. Long Short-Term Memory (LSTM) neural networks have been introduced to address the vanishing problem. LSTMs overcome the vanishing gradient problem by incorporating memory cells, allowing them to retain information over extended periods. This feature is particularly advantageous for modeling temporal data, such as spectrum occupancy [7] [8] [11] [16]. Long-term spectrum occupancy state prediction plays a crucial role in anticipating the channel idle period duration, which reduces channel switching costs and enhances channel selection in CRNs. Previous studies have overlooked quantifying the improvements achieved in spectrum occupancy state prediction. Accurately quantifying the improvements is essential for a comprehensive assessment of the effectiveness of the models in spectrum occupancy state prediction.

1.4. Objective of the Study

1.4.1. General Objective

The general objective of this study is to design a model for spectrum occupancy state prediction using deep learning algorithms for a cognitive radio network.

1.4.2. Specific Objectives

To achieve the general objective, this study has carried out the following specific objectives:

- To select relevant attributes for building the spectrum occupancy state prediction model.
- To build the spectrum occupancy state model with the proposed deep learning algorithm.
- To evaluate the performance of the model.
- To quantify the improvements achieved in the spectrum occupancy state prediction model.

1.5. Scope and Limitations of the Study

1.5.1. Scope of the Study

The scope of this study is to predict the spectrum occupancy state of the GSM900 MHz mobile network uplink band ranging from 902.5 to 915 MHz being used in Addis Abeba City.

1.5.2. Limitations of the Study

A CRN spectrum occupancy prediction involves the predictions of the occupancy state, radio environment, and transmission rate. This study is limited to the spectrum occupancy state prediction only.

1.6. Significance of the Study

This study will provide significant benefits for industry professionals and researchers, including:

- Improves national policymakers' and regulatory organs to perform spectrum management techniques based on real spectrum utilization levels.
- Motivates policymakers to adjust existing policies to promote the deployment of new services in underutilized bands.
- Informs legal and policy decisions by predicting future spectrum occupancy state, which can lead to more efficient spectrum usage and promote spectrum sharing.

1.7. Contribution of the Study

This study makes the following contributions to the field of cognitive radio:

1. **Defining the PU channel characterization in a new mode:** This study proposed a new time-domain approach called CFD to characterize primary user states. This spectrum sensing approach has a more accurate probability of channel detection than the ED, which can reduce the interference between the PUs and SUs.
2. **Long-term spectrum occupancy prediction:** This study developed a long-term spectrum occupancy prediction more than in the literature that can predict the spectrum occupancy state of how long it will be busy and idle. This allows the SUs to improve spectrum access, reduce channel-switching costs, and increase the CRN throughput.
3. **Improvements achieved in the spectrum occupancy prediction:** This study has quantified the improvements achieved in the spectrum occupancy state prediction.

1.8. Organization of the Study

The study is organized as follows in seven chapters:

Introduction (**Chapter One**) describes the background and motivation of the study, statement of the problem, objective of the study, objective of the study, scope, limitations, significance, and contributions of the study.

Theoretical Background and Related Works (**Chapter Two**) gives an overview of what is a CR, the main characteristics of a CR, spectrum sensing and access techniques, applications of a CR, challenges in spectrum sensing, and a review of previous works of spectrum occupancy prediction that have been done using DL models like ANN, RNN, GRU, LSTM, and HNN.

Research Methodology (**Chapter Three**) provides a brief background of methodology that can fit the proposed system architecture's design, implementation, and evaluation and parallelly discusses data selection, collection, and analysis in more detail the methods, procedures, and tools used to analyze the study.

Proposed Approach (**Chapter Four**) presents the proposed spectrum occupancy state prediction and a brief description of the sequence-to-sequence LSTM deep neural network architecture used to perform the spectrum occupancy state prediction.

Implementation Details (**Chapter Five**) discusses the details of the dataset used for this study, implementation details like the setup of each experiment, the result obtained, and the evaluation methods applied to measure the performance and the improvements achieved in the proposed model.

Conclusion and Future Work (**Chapter Six**) concludes the study result and presents the future work of the study.

Chapter Two

2. Theoretical Background and Related Works

2.1. Cognitive Radio

FSA policy leads to inefficient spectrum utilization as licensed users may not fully occupy all allocated spectrum channels (or frequency), resulting in artificial scarcity. In contrast to the FSA policy, where spectrum channels are statically assigned, CR has emerged as a solution to enable opportunistic spectrum access for SUs in instances where spectrum channels are not used by PUs. This approach enhances spectrum utilization, accommodating the growing demand for wireless connectivity that enables more devices to be connected [12] [13]. The spectrum decision made in CRNs can be characterized based on a comprehensive understanding of the spectrum occupancy states as depicted in Figure 2.1[5].

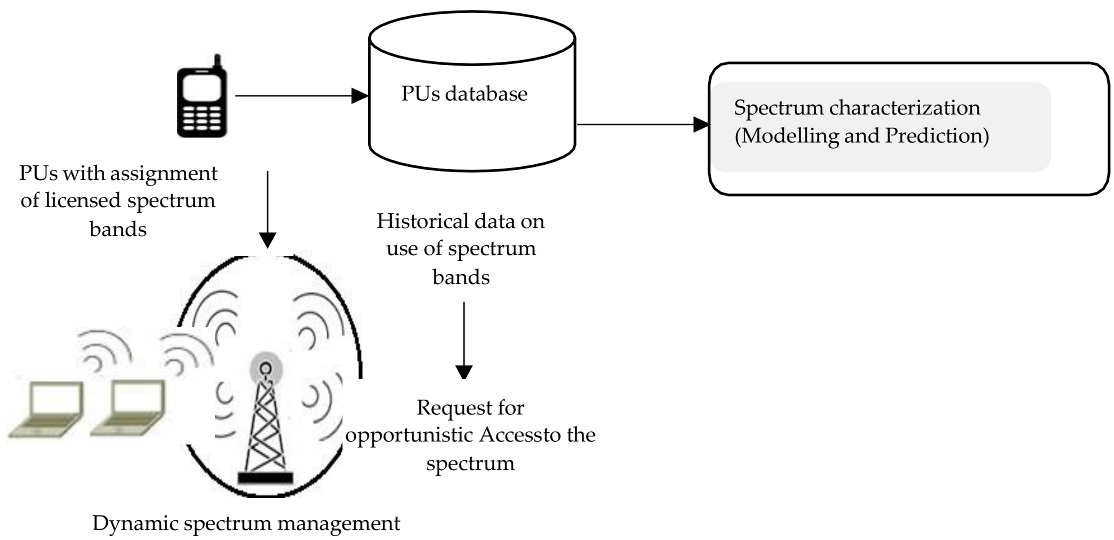


Figure 2.1 Contextualizing the characterization of spectral decision stage in CRN

CRs have distinct characteristics that distinguish them from traditional radio systems and Software-Defined Radios (SDRs). These distinctive characteristics are:

1. **Cognitive Capability:** CRs are equipped with the ability to sense the spectrum, allowing them to gain insights into their environment. Through spectrum sensing, CR systems can detect the occupancy and usage patterns of the spectrum bands. This cognitive capability enables CRs to make informed decisions regarding spectrum utilization. For instance, they can identify idle channels, and predict how long a channel is likely to remain idle. By dynamically adapting to the spectral environment, they enhance spectrum efficiency and optimize communication performance [17].
2. **Reconfigurability:** Unlike traditional radios that operate on fixed parameters, CR systems can dynamically adjust their operating parameters. This adaptability includes the ability to change frequency, modulation schemes, and transmission power levels without requiring hardware modifications. This dynamic reconfigurability empowers CRs to respond to changes in the radio frequency environment, mitigating interference issues and optimizing communication quality [14] [15] [17].

CRNs are intelligent networks capable of autonomously learning and dynamically adapting to optimize spectrum, this allows them to inherit the adaptability of the DL models that excel in learning complex patterns from data to make informed decisions. The integration of the DL models into CRNs with the capacity to analyze vast amounts of data and enhance the awareness of CRNs about their operating environment holds significant potential in spectrum-sharing models. The DL models can adapt and learn from the changing spectrum state conditions, allowing CRNs to dynamically optimize their communication parameters, spectrum channels, and transmission scenarios. Therefore, the synergy presents a relationship where DL's learning capabilities empower CRNs with enhanced situational awareness for intelligent decision-making [7][18].

2.1.1. Spectrum Sensing Techniques

In CRNs, spectrum sensing techniques can be broadly categorized into two main types: non-cooperative sensing and cooperative sensing [14] [15].

1. **Non-cooperative Spectrum Sensing techniques**

Non-cooperative spectrum sensing is a technique in which the CR independently decides the presence or absence of the PU signal in any given specified spectrum band.

It does not require any coordination with other CRs. The non-cooperative sensing techniques are further divided into ED, CFD, and matched filter detection (MFD) [15].

A. **Energy Detection**: is the simplest and most common sensing technique that works by comparing the energy of the received signal to a predetermined threshold. If the energy is above the threshold, then a PU is assumed to be present. ED is a low-complexity technique that does not require any knowledge of the PU signal. However, it is not very reliable in low SNR conditions, as it cannot distinguish between noise and the PU signal [13][14][15].
B. **Cyclostationary Feature Detection**: it works by exploiting the periodicity of the PU signals that measure statistical parameters which are the time function. It is a more sophisticated sensing technique that is more reliable in low SNR conditions because it is robust to noise uncertainties. CFD is a more complex technique, due to the requirement of prior knowledge about the PU signal [12] [13][14][15].
C. **Matched Filter Detection**: is the most complex sensing technique. It works by correlating the received signal with a known template of the PU signal. MFD is the most reliable sensing technique, but it requires knowledge of the PU signal for example, the type of modulation used, the central frequency, and the bandwidth. Any bad information on the signal will lead to the degradation of the detection performance [12] [13][14].

CFD is better than other techniques because it has a priori information about the signal and performs well for a low value of SNR. However, ED performs lower than others because it does not know a priori information about the signal, and the value of SNR significantly impacts the performance [13].

2. Cooperative Spectrum Sensing

Transmission channels are inherently susceptible to challenges such as fading and hidden node problems. Fading refers to the attenuation of signal strength as it traverses the transmission channels, a phenomenon induced by environmental factors like multipath propagation. On the other hand, the hidden node problem arises when an SU is unable to detect the presence of a PU due to physical separation[12], [13] [13] [14]. To address these issues, Cooperative Spectrum Sensing (CSS) techniques have been introduced. CSS involves multiple SUs collaboratively sensing the spectrum to collectively determine the state of a PU.

This cooperative approach enhances the reliability of spectrum sensing by combining information gathered from multiple SUs. CSS methods can be categorized into three main types: centralized, distributed, and cluster-based [12] [14] [15].

A. **Centralized CSS:** In centralized CSS, a central node collects the sensing information from all the SUs and decides the state of a PU. This is the simplest and most efficient way to implement CSS, but it requires a central node that is always available [14][15].
B. **Distributed CSS:** In distributed CSS, the SUs make their own decisions about the state of a PU, but they share their sensing information. This can improve the reliability of sensing, but it requires more communication between the SUs [14][15].
C. **Cluster-based CSS:** In cluster-based CSS, the SUs is organized into clusters, and each cluster has a cluster head. The cluster heads make decisions about the state of a PU based on the sensing information from the SUs in their cluster. This can improve the reliability of sensing and reduce the communication overhead, but it requires more coordination between the SUs [15].

CSS is a technique for overcoming the challenges of spectrum sensing in wireless environments. However, it still faces challenges, such as computational complexity and efficient information sharing.

2.1.2. Spectrum Access Techniques

There are three main spectrum access techniques used in CRNs: underlay, overlay, and interweave.

1. **Interweave:** In interweave, the SUs opportunistically access the channel when it is not being used by the PUs. This means that the SUs must sense the channel to determine whether it is available. If the channel is available, the SUs can transmit their signals. However, if the channel is being used by a PU, the SUs must stop transmitting and wait until the PU is finished [19][20].

2. **Underlay**: In underlay, the SUs transmit their signals in the same channel as the PUs. However, the SUs must ensure that their transmissions do not cause harmful interference. This means that the SUs must transmit at a lower power level than the PUs [19].

3. **Overlay**: In the overlay, the SUs transmit their signals in the same channel as the PUs, but they also cooperate with the PUs. This means that the SUs may help relay the PUs' signals or provide other types of assistance [19][20].

The choice of spectrum access technique depends on factors like the governing regulations of spectrum use, the capabilities of the SUs, and the requirements of the PUs. The interweave spectrum access technique is based on the prediction of the spectrum state by the PUs. The underlay and overlay spectrum access techniques are more complex than the interweave spectrum access technique. This is because they require the SUs to have more information about the PUs, such as the PUs' transmission power and the channel state information [19][20].

2.1.3. Applications of Cognitive Radio

CR has the potential to revolutionize a wide range of industries, including Military, Healthcare, Home appliances, Real-time surveillance, Vehicular networks, Addressing connectivity problems, Content distribution networks, Smart cities, Campus-wide network coverage, Disaster relief, Emergency networks, Weather forecasting and Traffic control [17].

2.1.4. Challenges in Spectrum Sensing

Spectrum sensing plays a pivotal role in CR systems that enable the identification of spectrum channel states, which SUs can then opportunistically access. Nevertheless, spectrum sensing encounters various challenges stemming from factors such as hardware constraints, hidden PUs problems, spread spectrum PUs, security, and sensing duration in the radio frequency environment.

1. **Hardware Constraints:** CRs need to be able to sense multiple frequency bands to identify unused spectrum. This requires high-performance hardware, which can be expensive[21].
2. **Hidden PUs Problem:** Factors like fading and shadowing cause these problems. This can be addressed by using cooperative sensing techniques in which several cognitive radios cooperate to enhance the overall sensing ability [14][15][21].
3. **Spread Spectrum PUs:** Spread spectrum PUs transmit their signals over a wide frequency range. This makes it difficult for CRs to detect them, as they may not be able to receive enough signal power to make a reliable detection. It can be detected with prior awareness of synchronization pulses and hopping patterns [15][21].

4. **Security Issues:** A harmful user can change its air interface to mimic the PU. It gives some wrong ideas about spectrum sensing. Public key encryption methods are used to avoid this scenario [15].
5. **Sensing Duration and Frequency**: The sensing duration and frequency are the parameters that determine how often and for how long a CR senses the spectrum. These parameters need to be carefully chosen to ensure that the CR can reliably detect PUs without wasting too much time or energy [15].

2.2. Related Works

Spectrum occupancy prediction is a critical problem in wireless communication technology. Many studies have been conducted on spectrum occupancy prediction using different DL models. This study reviews those studies that guided this study, focusing on the five selected models of ANN, RNN, GRU, LSTM, and HNN.

2.2.1. Spectrum Occupancy Prediction Using ANN

B. G. Najashi et al. [22] proposed spectrum occupancy prediction using a neural network model that aims to improve the reliability and accuracy of spectrum channel state prediction. The main attraction of employing neural networks for spectrum channel state prediction in cognitive radios is their ability to autonomously learn and adapt using historical data gathered by the cognitive radio system, eliminating the need for complete system redesigns. First, the reliability of the model was improved by using data obtained from cooperative spectrum occupancy measurements which involved two spectrum analyzers that combine measurements from the two nodes instead of the traditional method that involved a single device, because data captured in a spectrum occupancy measurement done using a single device can be viewed as unreliable, especially in harsh channel conditions. Cooperative spectrum sensing minimizes the effects of fast-fading and hidden terminal problems, which can mislead SUs and cause interference with the PUs. The collaboration among multiple spectrum sensors reduces the probability of false alarms and improves the overall reliability of the spectrum sensing process. Secondly, the prediction accuracy of the model was improved by employing a genetic algorithm for optimizing the weight selection of the model.

The data used for prediction was generated from five different services in a cooperative spectrum occupancy measurement conducted in Abuja, Nigeria. The data obtained from the measurements are combined using logical OR operators. The data used for prediction was generated from services of 800 broadcasting, GSM900 uplink and downlink, and 3G 1865 uplink and downlink bands. The process resulted in a total of 2,700 data sets. Each data set corresponds to a single slot having a duration of 16 seconds, which is the time required for the spectrum analyzer to perform one complete sweep. The accuracy of the model prediction was as follows: GSM 900 uplink: 96.77%, GSM 900 downlink: 99.85%, 3G uplink: 98.25%, Broadcasting: 99.93%, and 3G downlink: 99.75%. The difference in prediction accuracy between the uplink and downlink bands arises because base stations continuously transmit information to users in the downlink band, leading to more stable and predictable patterns. In contrast, the uplink band is used more sparsely, exhibiting less predictable patterns due to the randomness of occupancy.

D. **D. Das et al.** [23]proposed the design of a simple and faster spectrum prediction model known as a Functional Link Artificial Neural Network (FLANN) to forecast the future spectrum usage profile and to obtain utilization statistics for the industrial, scientific, and medical bands of 2.4–2.5 GHz. The study conducted measurements indoors at the Swearingen Engineering Center, University of South Carolina over five working days from Monday to Friday to present the occupancy statistics and gather realistic data for assessing the performance of the proposed model. Models such as MLP, RNN, and RBF are good for learning in CRNs, but they have high computational cost and complexity due to the hidden layer. However, FLANN is a novel single-layer model that eliminates the need for a hidden layer and is proposed to replace MLP due to its faster convergence rate and lower complexity. It operates on the entire input pattern using a set of linearly dependent functional expansions. Spectrum occupancy was estimated and predicted by employing different ANN models including MLP, RNN, Chebyshev FLANN, and Trigonometric FLANN. It is observed that the absence of a hidden layer in FLANN makes it more efficient than other ANN models in predicting the occupancy in less computational time and with less complexity, which achieved the best prediction accuracy of 99%. This suggests that FLANN is a promising approach for an effective spectrum occupancy prediction to designing intelligent learning systems for CR applications.

2.2.2. Spectrum Occupancy Prediction Using RNN

R. Fan et al. [24]proposed a multi-channel state prediction method based on the RNN model to address the scarcity of spectrum resources and enhance spectrum utilization efficiency in CR. First, they generated synthetic data to address the issue of how to model the primary user behavior in CR considering each PU has M channels and B Hz bandwidth, which are logically mutually independent ones.

Second, the historical information of the PU channel state behavior in CR by representing a binary sequence has been used to perform spectrum occupancy state prediction using the RNN. The RNN takes the multiple channel occupancy state information with a time step and uses the previous L time slots to predict the channel state at the L+1 time slot. Then, the predicted channel state at the L+1 time slot and the preceding historical information was used to predict the channel state at the L+2 time slot, and so on until the required length of prediction was achieved. The proposed multi-state predictor provides better performance when the time step increases. However, the improvement does not continue to increase when the time step reaches a certain length. Because the historical information strongly correlates with the channel state at the next moment for certain time steps only. Therefore, when the prediction length increases, the prediction accuracy will decrease due to information fading in learning the long-term dependence of RNN, leading to a problem known as vanishing gradient. The vanishing gradient problem is encountered in training RNNs when the gradient of the loss function for the parameters of the RNN tends to zero as the time step increases. This makes it difficult for the RNN to learn long-term dependencies. To reveal the proposed model advantage, it has been compared with other existing spectrum occupancy state prediction models, numerical results illustrate this method has better performance than other existing models and can provide sufficient information to the secondary users to utilize spectrum holes.

Learning long-term dependence is the biggest challenge in RNN. Gated RNNs (GRUs) that have memory blocks to control the flow of information within the network and were developed to address this challenge, allow to selectively remember and forget information over long periods.

GRU has the same structure as the RNNs, but it has improved in its hidden layer to maintain long-term memory achieved through memory blocks to regulate the flow of information into and out of the hidden state. The GRU has a simple and faster structure consisting of only two gates, namely the reset and update gates. The reset gate determines how much of the previous hidden state information should be forgotten, while the update gate determines how much of the new input information should be used to update the hidden state [25].

2.2.3. Spectrum Occupancy Prediction Using GRU

L. Xing et al. [26] proposed a GRU-based deep learning model for learning complex patterns in spectrum usage and adopted a time series prediction for multiple channel states in CR networks. This implementation of intelligent spectrum sensing and dynamic spectrum access enables the prediction of spectrum usage, facilitating compliance with spectrum usage regulations and improving overall spectrum utilization. They have investigated the use of channel state values from historical time slots to predict the channel state values of future time slots. The study examined the impact of input and output sequence lengths on prediction accuracy and discussed the performance differences between multi-user joint prediction and single-user independent prediction. After the channel state data is segmented into slices each slice represents channel states over a specific time interval. Each channel state sequence is then input into the model to obtain the corresponding output of the predicted future channel states. First, they evaluated the input sequence length effect on the model's performance. More accurate predictions were achieved within longer input sequence lengths containing more historical information. However, the prediction accuracy improvement is limited to a certain length because the model has a finite length of data dependence. Next, they compared the GRU-based model prediction accuracy with an MLP model. The GRU-based model achieved better prediction accuracy than the MLP model, especially when the input sequence length was too large. Second, they evaluated the output sequence length effect on the model's performance. The longer the output sequence length, the more accurate the prediction is for the immediate future. However, the prediction accuracy decreases for longer output sequence lengths. Lastly, they evaluated the joint prediction of multiple channels by taking four channels as one regarding them as independent and uncorrelated. The comparison with the independent prediction shows that it has more accuracy than predicting each channel independently because it can consider the dependencies between the channels.

2.2.4. Spectrum Occupancy Prediction Using LSTM

M. A. Aygul et al. [27] proposed an LSTM model to predict the next spectrum occupancy state because identifying spectrum opportunities is crucial for efficient spectrum utilization in CR systems. The aim is to improve spectrum occupancy prediction accuracy while reducing computational complexity. Spectrum occupancy states correlated over time, and spectrum prediction can be effectively treated as a time-series process.

This study demonstrates the advantage of leveraging both the correlation over frequency and the correlation over time to enhance the accuracy of spectrum occupancy predictions. By exploiting these correlations, this study investigates a more realistic scenario using a 2D LSTM model. This model incorporates previous time and frequency spectrum measurements to predict future spectrum occupancy states. The model was trained on data obtained from real-world spectrum measurements collected over one hour in a city surrounded by commercial and residential areas. The measurements were taken in the frequency range of 832-862 MHz, which is used by most telecom operators in Turkey for the uplink band. The model predicts spectrum occupancy by using known binary values to forecast the corresponding spectrum occupancy state at the next time instant. Extensive experimental results demonstrated that the proposed approach outperformed classical prediction methods in terms of accuracy, computational complexity and the efficiency of time-based prediction using DL models. The proposed model was compared to classical prediction models in terms of precision, recall, and F1-score. The model was also easy to train and converge faster than the other models. Therefore, the proposed LSTM model is a promising approach to improve the accuracy and computational complexity of the spectrum occupancy prediction.

A. Shenfield et al. [7] proposed an LSTM-based DL model to predict the availability of the spectrum the next time to reuse the underutilized spectrum frequencies, which is a promising approach to overcoming spectrum limitations. This allows the SUs to select more effectively the channels with the highest probability of being free the next time. The proposed model was evaluated using both synthetic and real data sets. The results show that the proposed model outperforms the Markov chain-based model in predicting the availability of a channel for multiple future steps. For example, the probability of success in selecting an idle channel reaches 90% for two-step ahead predictions and 80% for five-step ahead predictions using a synthetic data set.

This indicates that the proposed model algorithm can also minimize the channel switching cost by accurately selecting a free channel for more than one-time slot ahead. The proposed model, tested on real datasets, achieves a selection accuracy of 80% for two-step-ahead predictions and 70% for five-step-ahead predictions. This represents, on average, a 5% improvement in the probability of success compared to other existing methods in the literature.

L.Yu et al. [28] built an LSTM model to predict future spectrum availability at arbitrary locations by leveraging the intrinsic spectral-temporal correlations. This approach solves the problem of spectrum availability in cognitive aerospace communications, which is more effective and efficient than a model based on temporal correlations only. The model sets a relatively large number of nodes to better characterize the dynamic states of spectrum occupancy and more effectively capture the features of the input data. The model performance was evaluated using real-time spectrum occupancy data gathered from the cities of New York and Vienna. The data covered the frequency range of 3MHz to 5.4MHz, divided into 26 channels. The channel's availability was predicted by setting the look-back window length, which is the number of past observations used to make a prediction, from 15 to 90 time slots. The results showed that the model achieved an accuracy of 94.06% when the look-back window length was 60. However, when the look-back window length was set to 75, the model accuracy improved to 98.14%. This shows that the model can learn long-term temporal correlations in the spectrum data, which helps to improve the predictions' accuracy. The model was also compared to an ANN model. The results show that the LSTM model outperforms the ANN model in terms of accuracy. This is because the LSTM model can capture the long-term dependencies in the data.

L.Yu et al. [29] an LSTM-based model with multiple layers was constructed and trained through supervised learning to predict frequency hopping sequences. This model is designed to resist interference by accurately predicting the frequencies used by adversaries. By correctly predicting these interference frequencies, the model enables effective avoidance, thereby enhancing the robustness of communication systems against interference. The output of the LSTM network is transformed into 0 or 1 which indicates the idle state or occupied compared to the threshold. The simulation data used in this experiment consisted of frequency hopping patterns with a length of 160-time slots across ten channels. The model achieved a prediction accuracy of 90%.

They have compared the influence of the LSTM network depths and widths on the prediction accuracy performance with a Back Propagation (BP) model. The prediction accuracy of the BP model increases when the network depth increases, while the prediction accuracy of the LSTM model increases when the network width increases. The experiment results show that the LSTM model has better predictive performance than the BP model. The shallow and wide LSTM neural network model captures the characteristics of the data set better than the deep and narrow LSTM neural network model.

However, the prediction performance of the BP network did not change significantly under similar situations. When there is something wrong with spectrum sensing, channel states with error and correct channel states both can be used as learning labels. The simulation results indicate that prediction accuracy for the latter one is about 4% better than the previous one.

B. S. Shawel et al. [30] proposed a Convolutional Long Short-Term Memory (ConvLSTM) model to enhance long-term temporal prediction by learning the joint spatial-spectral-temporal dependencies observed in spectrum usage. This approach seeks to alleviate the inefficient use of the radio spectrum through DSA with CR enabling opportunistic users that share spectrums without causing interference when these spectrum channels are not being used by their PUs. The data used to train the ConvLSTM model was collected from a real-time environment measurement in the UHF bands of 450-520 MHz to evaluate the prediction accuracy of the proposed network for increasing future time steps and different spectrum channels. Five sparsely distributed sensors were used to collect the data and the signal power level was sent to a central entity for decision-making. The ConvLSTM model predicted the spectrum occupancy for the next 3 hours. The prediction's accuracy was assessed utilizing the root mean square error (RMSE), yielding a value of 5.012%. During prediction, an increase in future time steps leads to a higher accumulated error from previous predictions, resulting in an increased RMSE. However, the RMSE values have remained below 5.012, the benchmark set by the combined sample standard deviation. This demonstrates that the predictions effectively capture spectral dependencies and long-term temporal spectrum usage patterns, even though they may not be as accurate for short-term changes.

Spectrum sensing and prediction from a single SU perspective is unreliable under harsh environmental conditions. This is because the SU does not have a clear view of the spectrum channel state and is affected by problems such as hidden terminals, shadowing fading, and multipath fading [8]. To address these challenges, cooperative spectrum sensing has been proposed. In cooperative spectrum sensing, multiple SUs collaborates to sense the spectrum and make a more reliable decision about the spectrum channel state. This approach can improve the detection of the channels and reduce the probability of interference. However, cooperative spectrum sensing faces challenges such as the need for more hardware-intensive devices for sensing and the time required to combine the data collected from multiple devices [15].

2.2.5. Spectrum Occupancy Prediction Using HNN

S. S. Shirgan et al. [31] proposed a Hybrid Neural Network (HNN) model that combines Radial Basis Function (RBF) and Multilayer Perceptron (MLP) techniques to enhance prediction performance by proper learning for the deployment of CR, intending to achieve energy-efficient and time-efficient spectrum analysis. The measurement results of spectrum occupancy utilization analysis provide information only about the current occupancy status, not future usage. In the context of spectrum sharing and mobility in CR, offering cognitive users a predefined channel list in advance would be a more effective solution to mitigate interference with existing PUs. Therefore, CR performance can be enhanced by adopting predictive methods to analyze spectrum occupancy and determine the future occupancy status of PUs. First, the spectrum occupancy analysis was performed using the statistical method of data collected from a seven-day measurement campaign. The results show that the overall occupancy was 6.32% in the TV band, 45.24% in the GSM900 MHz, 36.91% in the GSM1800, and 13.09% in the UMTS2100. Second, the HNN model predicts the spectrum occupancy at weekends and weekdays, the result shows the HNN prediction outperformed the MLP and RBF model's prediction accuracy by combining the strength of both MLP and RBF. Even though MLP is simple to understand and implement, it requires more hidden layers for better approximation, this makes it difficult to train and slows down the learning process. RBF has a single hidden layer with a Gaussian activation function, which makes it faster to train than MLP. However, RBF is more complex to understand and interpret.

The model's performance was evaluated using the RMSE of the four bands' predictions: 1.5%, 4.5%, 0.5%, and 0.7%, respectively. They also tested the performance of the HNN model for seven days. The mean expectation error was a minimum of 0.036. The model was found to perform poorly on Sundays than on weekdays due to the unpredictable behaviour of the spectrum on holidays. The performance of the model examined for popular bands at weekends and weekdays was found to be accurate compared to MLP, and RBF. This proves that by combining MLP and RBF, the HNN can achieve the best of both worlds. It is simple to understand and implement as MLP, while also being as fast to train as RBF.

2.2.6. Summary of Related Works

Table 2.1 Summary of Related Works

Authors	Methodology	Objective	Gaps
B. G. Najashi et al. [22]	ANN	To improve the reliability and accuracy of spectrum occupancy prediction.	Short-term prediction. Cooperative SS.
R. Fan et al. [23]	RNN	To predict the future channel state multiple steps ahead.	Synthetic data. does not solve the vanishing gradient problem.
M. A. Aygul et al. [27]	LSTM	To improve the spectrum occupancy prediction accuracy and computational complexity.	Short-term prediction.
B. S. Shawel et al. [30]	ConvLSTM	Long-term prediction.	Computationally complex and more hardware intensive.
S. S. Shirgan et al. [31]	HNN	To achieve better spectrum prediction performance.	Short-term prediction.

The table above summarizes some studies that predict spectrum occupancy. These studies have achieved promising results, but they have limitations like performing short-term predictions that do not consider the effect of channel state on future time slots, using unreliable spectrum sensing methods, and not identifying potential candidate spectrum bands. This study addresses these limitations by developing a reliable spectrum sensing method that can correctly detect the state of the PU, identify the potential candidate spectrum bands based on a proper analysis of spectrum utilization, and develop a long-term spectrum occupancy prediction model.

Chapter Three

3. Research Methodology

3.1. Overview

This chapter elaborates on data selection, collection, pre-processing, details about data analysis, and the tools used to conduct this study. First, the source of data and the collection methods applied were explained. Next, it explains the data pre-processing activities like feature selection that have been used to select essential features for building the model, cleaning the data, and transforming the data into a format suitable for the model building where feature engineering and data analysis were performed. Finally, the tools used to analyze the data and to conduct the study, and the performance evaluation metrics used in this study have been presented.

3.2. Data Collection

The data used for this study was collected from a real-world spectrum measurement in Addis Abeba, Ethiopia using the TCI spectrum monitoring system. The real-time measurement was conducted in the GSM900 MHz mobile network uplink spectrum band ranging from 902.5 to 915 MHz for five consecutive days from January 28th to February 1st, 2021. The area of Bole was selected for spectrum measurement due to its status as a commercial hub, which is expected to have a higher spectrum demand than other areas [31]. The resolution bandwidth of each spectrum channel was 100 kHz with 4 minutes resolution time. This resulted in a total of 450,000 measurement data points for five days. The GSM900 MHz uplink band presented to be a promising potential candidate for the deployment of a CR due to its underutilization from the sparse use of its users communicating on the network [22] [32] [31].

3.2.1. Description of Data Features

The data features contain spectrum measurements that include Channel, Frequency, Maximum occupancy (%), Average occupancy (%), Maximum field strength, and Average field strength. These attributes are time series data attributes that can be used for the implementation of the spectrum occupancy state prediction.

The following table 3.1 is a description of all the spectrum measurement data features or attributes.

Table 3.1 Description of spectrum measurement data attributes

Attribute	Type	Description
Channel	Numeric	Channel number
Frequency	Numeric	Frequency of a channel
Max Occupancy	Numeric	Percentage of maximum occupied time
Average Occupancy	Numeric	Percentage of average occupied time
Max power gained	Numeric	Values of maximum power obtained
Average power gained	Numeric	Values of average power obtained

3.3. Data Pre-processing

Data pre-processing is all about cleaning the data and removing the noise, selecting the features to reduce the data dimension, and transforming it into a format appropriate for the proposed model. So, before it has been applied to the proposed model, the raw data has passed through several transformation steps that provide a significant data quality to improve the model performance. In this study, after the spectrum measurement data has been obtained from the spectrum monitoring system, it has been arranged in a proper format. Data pre-processing was started with feature selection parallelly used for data analysis, and after that data cleaning was performed. Finally, data transformation has been applied.

3.3.1. Feature Selection

All the data features obtained from the spectrum measurement are not necessary to predict the spectrum occupancy for the proposed model. The TCI spectrum monitoring system presents two types of attributes, some have valuable information about spectrum occupancy while others don’t. For this study, feature selection was performed by selecting only the features that have relevant information for the spectrum occupancy state prediction model, and ignoring features that don’t have valuable information [33]. The data collected from the TCI spectrum monitoring system have six features which are Channels, Frequency, Maximum occupancy (%), Average occupancy (%), Maximum field strength, and Average field strength.

For modeling spectrum occupancy prediction, three features were selected based on filter-based feature selection [34]: Frequency, Average Occupancy (%), and Average Field Strength. These features were chosen because they more effectively describe spectrum occupancy. In contrast, the maximum values of occupancy and field strength represent only the peak values observed at a single instance during the measurement period, which do not accurately reflect the overall state of spectrum occupancy.

3.3.2. Data Cleaning

Cleaned data was used to improve the performance of the prediction model. Data quality problems arise from mistaken data entry, missing values, redundant, and invalid data. Data cleaning was performed by removing redundant data using appropriate techniques to remove inconsistencies that helped to improve the performance of the proposed model.

3.3.3. Feature Engineering

Data found in the real world can be messy. Feature engineering transforms such data into a common understandable format. It ensures that all numerical values are on the same scale to avoid computational error, making the data easy to interpret and understand. It can also improve the model performance by incorporating domain knowledge, such as classifying the raw numeric values obtained from the spectrum measurement into two states: i.e., idle or busy [35]. In this study, depending on the spectrum measurement numerical value, if the duty cycle is lower than 51% in the CFD-based SS, the channel can be considered “idle” because mostly the spectrum channel was not in use.

3.4. Setup the Data

After all data pre-processing tasks have been completed, the collected dataset is divided into training and validation sets with a ratio of 80%, and 20%, respectively. The data setup has 180,000 data points selected for training, and the other 45,000 data points were used for validating the prediction algorithms. The validating points were used to validate the performance of the prediction algorithm.

3.5. Design and Implementation Tools

Many types of tools were used to conduct this study, starting from data collection and pre-processing to the design and implementation of the long-term spectrum occupancy prediction. The following design and implementation tools, which can be hardware or software, were used during the study for the design, implementation, and reporting of the study document:

3.5.1. Hardware Tools

The following table shows the hardware tool used in this study along with its specific function throughout the study.

Table 3.2 Hardware tools used for implementation of the study

S.no	Device name	Used in the Study
1	TCI Server	The TCI spectrum monitoring system was used to collect the spectrum occupancy data.
2	Hard Disk	Used as storage for large datasets.
3	GPU	To increase the computation and to fasten the training.
4	RAM	To accelerate the training process cooperatively with GPU.

3.5.2. Software Tools

Software tools and libraries were used in the study to write the code, debug the program, visualize results, collect data, and report the study document.

Scorpio Client: is an application software that provides an interface to the measurement server (Scorpio Server). The Scorpio client provides all functions necessary to arrange the measurement parameters, to download the data from the server, and to select the appropriate data features from the data.

Anaconda: is an application used to install Python programming language with its all modules. It provides a navigator application to view different settings like Jupyter Notebook, spider, vs-code, and modules installed in the environment.

Python: Programming language used to perform the implementation of the study.

Jupyter Notebook: An interactive web-based application that helps to configure, load Python API, and to write Python code.

Keras: API with easy extensibility to work on CPU and GPU, that supports modularity, and works with Python as a high-level framework.

TensorFlow: a machine learning library used to build a model and deploy it in client environments. It supports ML, DL, and other types of flexible numerical computation.

Microsoft Office Packages: Tools used to write and organize the study like Word to write the document, Excel to make the CSV dataset, Visio to draw diagrams and write equations, and PowerPoint to make the presentation slides.

Mendeley Desktop: is a powerful reference manager tool that serves as an academic, social network for referencing similar works.

Bib Guru: Excellent reference and citation generator tool that can quickly add sources paper and make citations in IEEE and other styles.

3.6. Model Performance Evaluation Metrics

A model can be evaluated using different metrics to quantify its performance on a test set. These metrics measure how the model fits the data, works on new data, and makes no discrimination among the model results. Some metrics used for the prediction of time-series data include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) [36] [37].

3.6.1. Mean Squared Error

It is the average of the square of the difference between the predicted values and actual values. It has the same units as the actual and predicted values squared and is always positive.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y'_t - y_t)^2 \qquad 3.1$$

Where y'_t is the predicted value, y_t is the actual value, and n is the total number of values in the test set. It is clear from the equation that MSE is more penalizing for larger errors or outliers.

3.6.2. Root Mean Square Error

It is the square root of the mean square error. It is also always positive and is in the range of the data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y'_t - y_t)^2} \quad 3.2$$

Where $y't$ is the predicted value, yt is the actual value, and n is the total number of values in the test set. It is in the power of unity and is more interpretable than MSE. RMSE is also more penalizing for larger errors.

3.6.3. Mean Absolute Error

It is the average absolute difference between predicted and actual values. It has the same units as the predicted and actual value and is always positive.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y'^t - y^t| \quad 3.3$$

Where $y't$ is the predicted value, yt is the actual value, and n is the total number of values in the test set.

3.6.4. Mean Absolute Percentage Error

It is the average absolute difference percentage between predicted values and actual values, divided by the actual value.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y'^t - y^t}{y^t}\right| * 100\% \quad 3.4$$

Where $y't$ is the predicted value, yt is the actual value, and n is the total number of values in the test set.

3.7. Confusion Matrix

A confusion matrix is a table that shows how well a model is performing on a given dataset. It compares the actual values of the data to the predicted values.

1. **True positive (TP)**: This is the condition when both the actual and the predicted values are true. This is known as the detection probability in a CRN.
2. **True negatives (TN)**: This is the condition when both the actual and the predicted values are false. This is known as the identification probability in a CRN.
3. **False positives (FP)**: The condition when the actual value of the data is false whereas the predicted value is true. This is known as the false alarm probability in a CRN.
4. **False negatives (FN)**: The condition when the actual value of the data is true whereas the predicted value is false. This is known as the miss-detection probability in a CRN.

The confusion matrix can be used to calculate several performance metrics, such as accuracy, precision, specificity, recall, and F1 score. These metrics can be used to compare the performance of different models and to identify areas where the model can be improved.

1. **Accuracy:** Accuracy is the percentage of predictions that are correct. It is calculated by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad 3.5$$

2. **Precision:** Precision is the percentage of predicted positives that are actually positive. It is calculated by dividing the number of true positives by the number of true positives plus the number of false positives**.**

$$Precision = \frac{TP}{TP+FP} \qquad 3.6$$

3. **Recall:** Recall is the true positive rate used to measure the fraction of positive values that are correctly predicted.

$$Recall = \frac{TP}{TP+FN} \qquad 3.7$$

4. **Specificity:** Specificity is the true negative rate used to measure the fraction of negative values that are correctly predicted.

$$Specificity = \frac{TN}{TN+FP} \qquad 3.8$$

5. **F1_score: The** F1-score is a weighted average of precision and recall or the harmonic mean between recall and precision, used to measure the model's overall accuracy.

$$\text{F1\_score} = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad 3.9$$

Chapter Four

4. Proposed Approach

4.1. Overview

This chapter discusses the details of the proposed approach and the model architectures that include the block diagram, flow charts, and the algorithms used to solve the study problem. Moreover, a brief description of the sequence-to-sequence LSTM deep learning neural network architecture used in this study to predict spectrum occupancy state from the frequency spectrum and temporal perspectives used was explored.

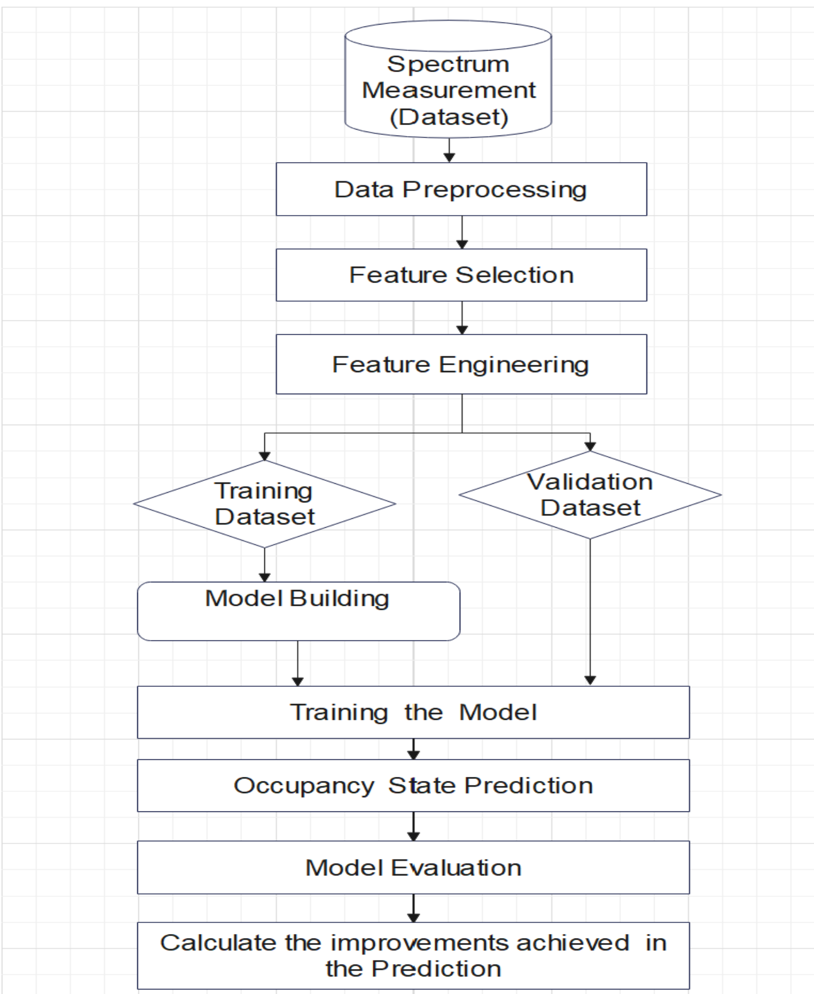


Figure 4.1 The Proposed Deep Learning Approach

4.2. System Model

This study aims to predict the spectrum occupancy state used to model a CRN. The heterogeneous spectrum occupancy state model was used to implement a CRN for spectrum sharing in a DSA model [7]. The spectrum band has been divided into k contiguous frequencies or channels. The channel states represent the spectrum channel state at time t and are denoted by a vector matrix. Each element in the matrix represents the corresponding channel that is occupied, or idle which is ready to be used by the opportunistic SU [28]. A heterogeneous spectrum occupancy state model has multiple PUs and SUs, which are centrally controlled by a database that identifies the spectrum occupancy states based on prediction[4]. This study has proposed a long-term spectrum occupancy state prediction model in a stationary location by exploiting the spectral and temporal correlation of the data. The occupancy of a channel is characterized by the presence of a PU signal, while the presence of a spectrum hole characterizes the vacancy of a channel. These cases are formally stated as hypotheses (H0) and (H1).

$H_0: y[t] = w[t]$ when there is no PU

$H_1: y[t] = h[t]x[t] + w[t]$ when PU's the signal is present 4.1

where x[t] denotes the PU signal, w[t] is white noise and y[t] is the received signal at t^{th} time instant. H_0, the null hypothesis indicates the noise samples while H_1, the alternate hypothesis indicates the presence of PU signal along with noise t^{th} instant [6] [12] [17].

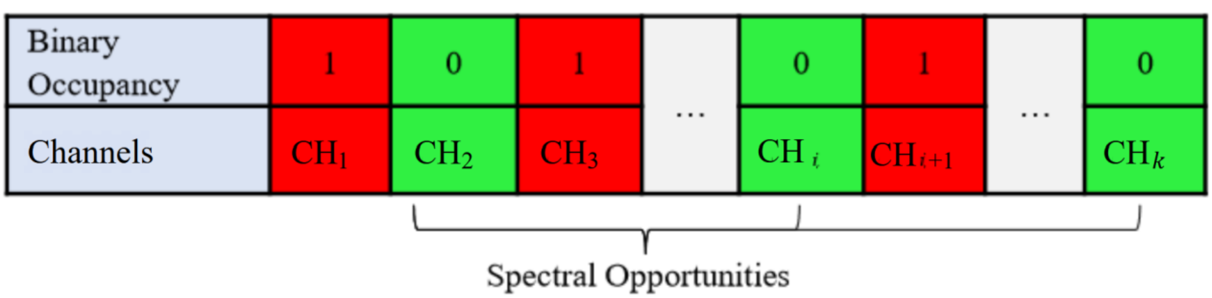


Figure 4.2 Spectrum channels occupancy state modeling

For the sequentially obtained time-series spectrum measurement data $X_1, X_2, X_3, X_4, \ldots$, the long-term spectrum channel state prediction using the deep learning model can be done within the sequence-to-sequence neural network architecture in the LSTM deep learning model is defined as $(X_{t-n}, \ldots, X_{t-2}, X_{t-1}, X_t)$ to $(X_{t+1}, X_{t+2}, X_{t+3}, \ldots, X_{t+m})$ where *n* and *m* represent the historical observations and the future instants in time, respectively [30].

4.2.1. Time-Series Model Analysis

Time series data is composed of sequence data points measured over regular intervals. Time series models are designed to comprehend the patterns and trends inherent in the data that can feat the natural temporal ordering. This shows that observations closer in time are more similar than further apart observations because values in a time series data at a given time were derived from past values. Time series analysis is a methodology used to study time series data that identifies relationships and makes predictions of future values[38]. Time series data analysis begins by identifying whether the data is stationary or non-stationary. A stationary time series has consistent statistical properties, including mean, variance, and autocorrelation. This stability implies that the underlying data-generating process is predictable and stable. On the contrary, non-stationary time series data displays varying statistical properties over time, often influenced by trends, seasonality, or other random fluctuations. Time series prediction is a technique that estimates future values based on historical data. In spectrum occupancy state prediction, the objective is to forecast the state of the spectrum occupancy state at the next time point [38][39].

Time series data should be selected carefully, considering the different variations that can occur at different timescales. For example, a day can have four seasons (morning, afternoon, evening, and night), while a week can have only two (weekday and weekend). The spectrum occupancy data can vary significantly depending on the time of day, day of the week, and peak or trough times. Therefore, representative data must be chosen for the specific period and timescale of interest [32].

4.3. Proposed Deep LSTM Architecture

This study proposed a long-term spectrum channel state prediction using a deep-learning LSTM model. The method uses a sequence-to-sequence neural network architecture based on the LSTM to predict the spectrum channel state based on preceding channel states.

The proposed model can accurately predict the spectrum channel state for the next time slot and several time slots ahead. The proposed long-term spectrum channel state prediction model is then implemented to facilitate a DSA. This model enables users to access spectrum channels that are not used by PUs, enhancing overall spectrum utilization efficiency.

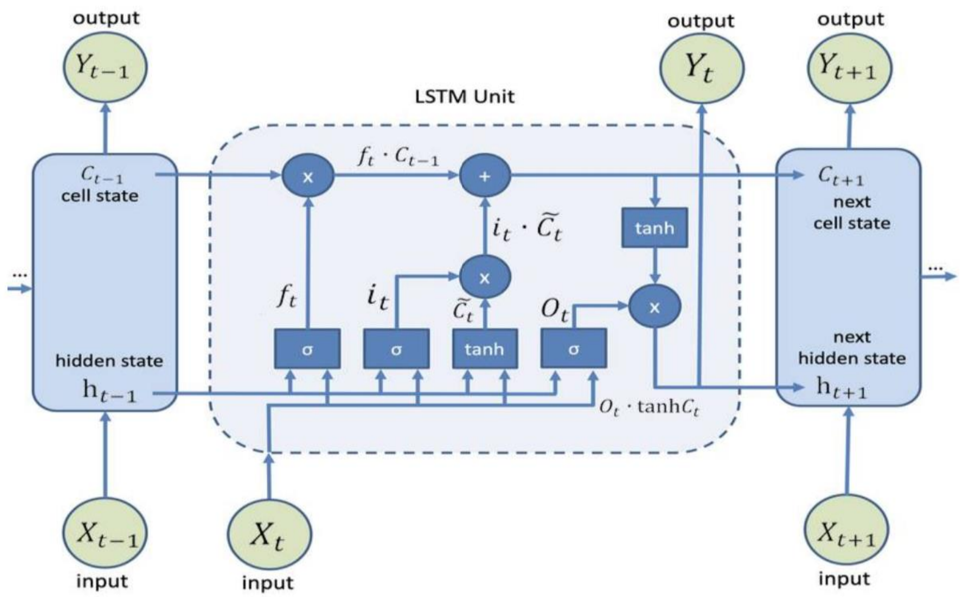


Figure 4.3 Basic Architecture of the LSTM Model

Conventional RNNs can model the temporal dynamic behavior of sequential inputs, but they have difficulty learning long-range dependencies. This is because backpropagating errors through long sequences can cause the vanishing gradient problem. LSTM networks solve this problem by using a gating mechanism to control the information flow in the network. The gating mechanism consists of three gates: the input gate, the forget gate, and the output gate. And the cell memory that is used to store information over long periods [7] [8].

1) The **input gate** which controls how much new information is added to the LSTM's memory state.

2) The **forget gate** which controls how much information is removed from the memory state.

3) The **output gate** which controls how much information from the memory state is output by the LSTM cell.

The gates are mathematically formulated as:

$$i_t = \sigma\left(W_{ix}X_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{fx}X_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \varphi\left(W_{cx}X_t + W_{ch}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{ox}X_t + W_{oh}h_{t-1} + W_{oc}c_t + b_0\right)$$
$$h_t = o_t \circ \varphi\left(c_t\right) \quad 4.2$$

Where i, f, o, and c denote the input gate, forget gate, output gate, and cell state, respectively. All these gates are on the same dimension as the hidden vector h is assumed to be of N ×1 dimension. σ is a sigmoid function, and φ is a nonlinear function that maps the input to [-1, 1]. W_{ic}, W_{fc}, and W_{oc} are the peephole connection matrices that connect cell state to their respective gates. Whereas, W_{ix}, W_{fx}, Wox, and W_{cx} are the weight matrices that connect the input vector X_t and input gate, forget gate, output gate, and cell state, respectively. Because the gates and the input vector X_t have the dimensions of N ×1 and M × 1 respectively. The dimensions of matrices can have W_{ih}, W_{ic}, W_{fh}, W_{ch}, W_{oh}, and W_{oc} are all the same, which is N × N, and the dimensions of the matrices W_{ix}, W_{fx}, W_{cx}, and W_{ox} are N × M.

4.3.1. Proposed Deep Learning LSTM Basic Architecture

In the model presented in [27], illustrated in Figure 4.3, an LSTM neural network architecture was employed for long-term spectrum occupancy prediction. The model design focuses on a specific set of frequencies, with known binary values provided for spectrum occupancy state prediction within the model. The input and output data for the model are constructed using a sliding window that traverses both the time and frequency axis's. This process generates a 2D matrix from the spectrum measurement data, where each element corresponds to a specific point in time and frequency, along with its associated binary value. This matrix serves as the input dataset used to train the model. During the validation stage, the proposed model demonstrates its operational capability in real-time.

Spectrum measurements for time and frequency lags were utilized to predict the corresponding spectrum occupancy state at the next instant. This prediction is achieved by inputting the binary values, arranged as a grid, to the model, which then generates the binary values for the subsequent instant [27].

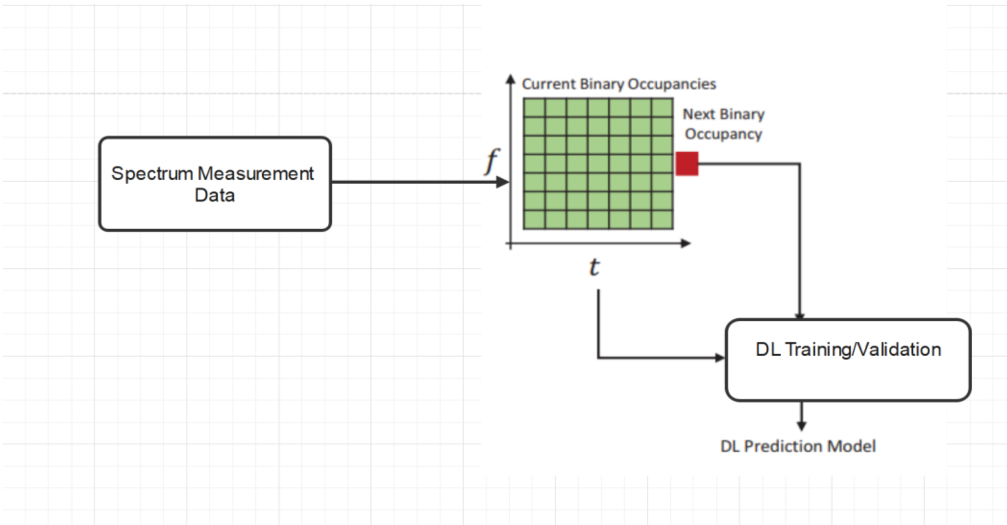


Figure 4.4 Learning and predicting long-term spectrum occupancy in LSTM

In this study, the CRN model was developed to enhance spectrum resource utilization through long-term spectrum occupancy prediction. The model is conceptually likened to a smart parking management system that has two types of customers, illustrating its capability to optimize the use of spectrum resources akin to the efficient utilization of parking spaces. Much like a smart parking system identifies and directs vehicles to available parking spaces, the CRN can identify both available and occupied spectrum channels. It then guides SUs to optimal spectrum channels, providing information about their availability and duration. This approach aims to optimize spectrum utilization in a manner analogous to how a smart parking management system optimizes parking space usage. Therefore, the two technologies have similarities in their operation, emphasizing resource optimization, dynamic adaptability, and the pursuit of key objectives.

4.3.2. Proposed Deep LSTM Architecture Components

The sequence-to-sequence LSTM model used for spectrum occupancy state prediction consists of the following major components [28]. The main components of the LSTM units are:

1. The Main Framework

First the input goes through the LSTM layer to extract long-term dependencies from the input sequence, and the output goes into the full connected LSTM dense layer, where a dropout process is applied to avoid overfitting in training. Finally, the Sigmoid activation function is employed to obtain the final prediction result. The input X_t in the time slot t is a matrix of dimension $M \times T$, which means that the system predicts the spectrum occupancy state in the forthcoming time slot by exploiting the data in the most recent T time slots. In the input, each column represents the spectrum occupancy state in a time slot.

2. **LSTM Layer**

The input sequence undergoes processing through an LSTM layer, a RNN tailored for capturing and learning long-term dependencies within sequential data. This layer effectively analyses the input sequence, enabling the extraction of long-term patterns. The final output of the LSTM layer, representing the output of the last LSTM memory cell, is subsequently fed into a dense network for further processing.

3. Dense Network with Dropout

The dense layer is used to reduce the dimensionality and to produce the prediction final output, in the dense layer every node is connected to all nodes in the preceding layer**.** Specifically, it receives the output from the LSTM layer and learns to map it to predict spectrum occupancy in the subsequent time slot. To address overfitting, the model incorporates dropout, a regularization technique that helps to prevent overfitting. Dropout randomly drops out some of the neurons in the dense layer during training to prevent the model from overfitting and enhance the prediction performance.

4. Activation Function and Loss Function

In the computation of the final prediction output, the activation function chosen is Sigmoid, applied to the output of dense networks. Specifically, this activation function maps the output of the dense networks into a vector of elements ranging between 0 and 1.

Each element in the vector represents the probability of a channel being occupied, and the sum of all elements equals 1. To map the probabilities into binary series during training, an optimal threshold is determined. This threshold minimizes the disparity between actual data and the predicted values, which are either 0s or 1s. For learning and evaluation, a loss function is defined. In this system, cross-entropy is used as the loss function. This choice is motivated by the classification nature of the problem, where the goal is to categorize inputs into two labels. Cross-entropy has been widely adopted and demonstrated to be effective in classification tasks, making it a suitable choice for this work.

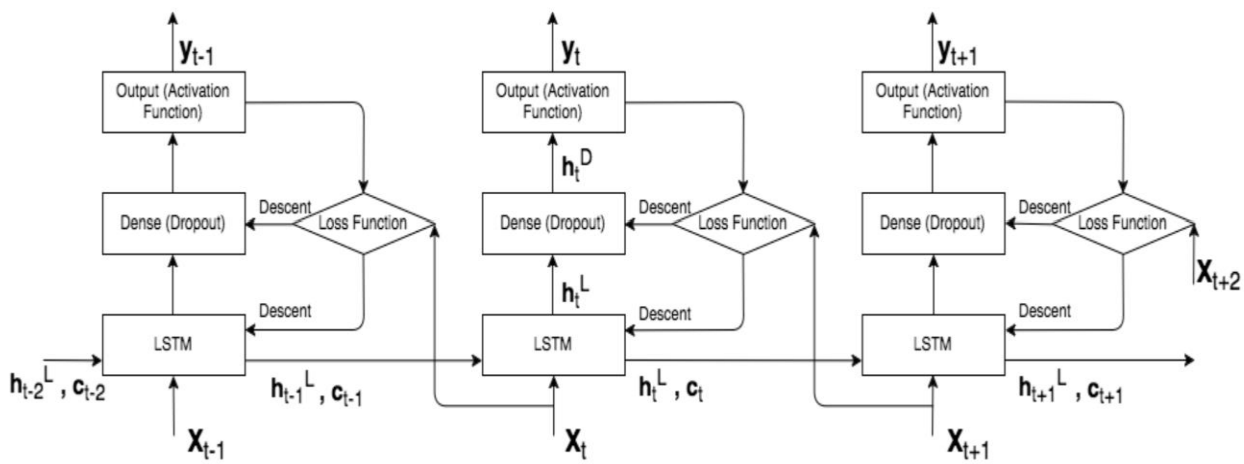


Figure 4.5 LSTM architecture with input X_t in time slot t

Chapter Five

5. Implementation Details, Result and Discussion

5.1. Introduction

This chapter details the data analysis and feature description, the experimental setup and configuration for each experiment, the evaluation methods applied to measure the model performance, the results obtained, and discussions about the improvements achieved in long-term spectrum occupancy prediction.

5.2. Data Set Description

The dataset used for this study was collected using the TCI spectrum monitoring system. The dataset contains spectrum occupancy measurements from January 28, 2021, to February 1, 2021. The dataset includes Channel, Frequency, Maximum occupancy (%), Average occupancy (%), Maximum field strength, and Average field strength. These attributes are time series data attributes that were used for the implementation of the spectrum occupancy state prediction.

Table 5.1 A sample data obtained from the spectrum measurement campaign

Channel	Frequency (MHz)	Occupancy (%)		Field Strength (dB)	
		Maximum	Average	Maximum	Average
1	902.6	0	0	-102	-107
2	902.7	0	0	-113	-115
3	902.8	52	52	-82	-88
4	902.9	52	52	-82	-88
39	906.2	75	60	-91	-95
40	906.3	75	60	-91	-95
41	906.4	0	0	-111	-113
42	906.5	0	0	-111	-113
60	908.3	100	100	-72	-73
61	908.4	100	100	-72	-73
62	908.5	100	100	-88	-90
63	908.6	100	100	-88	-90

5.2.1. Feature Description and Selection

All the features stated in Table 5.1 were not used to model the spectrum occupancy because they do not capture unique information and may negatively affect the generalization of the prediction model. To address this, a feature reduction technique was employed to eliminate redundancies, selecting only those features essential for describing the spectrum occupancy model. Feature selection aims to reduce the number of input variables, preventing an undue increase in model complexity and preserving its generalization capability. In this study, filter-based feature selection was implemented for feature reduction. Filter methods rely on statistical measures, such as information gain, to identify features that contribute the most information about the target variable. Notably, filter methods exclusively consider the association between each feature and the class label [34]. Following the feature reduction process, the selected spectrum occupancy state prediction features comprised frequency, average occupancy, and average field strength. These features were chosen based on their relevance and significance in the context of spectrum occupancy state modelling. The refined set of features is detailed in Table 5.2, providing a clear understanding of the variables considered in the predictive modelling of spectrum occupancy.

Table 5.2 The final features selected used to model the spectrum occupancy state prediction

Frequency	Average Occupancy (%)	Average Field Strength (dB)
902.6	0	-107
902.7	0	-115
902.8	52	-88
902.9	52	-88
906.2	60	-95
906.3	60	-95
906.4	0	-113
906.5	0	-113
908.3	100	-73
908.4	100	-73
908.5	100	-90
908.6	100	-90

These methods were applied to ascertain the average spectrum utilization within the designated spectrum band.

The methods used to characterize the spectrum occupancy model are the average field strength (or power spectral density) and average occupancy (or duty cycle). Average field strength indicates the intensity of the signal, providing insight into the strength of signals present within the analysed spectrum. On the other hand, average occupancy measures how often the signal is present or the frequency with which signals are detected, reflecting the temporal presence of signals within the specified spectrum band. In summary, the results presented in the table offer a comprehensive understanding of the spectrum occupancy state through signal strength (or average field strength) and temporal presence (or average occupancy) using the ED and CFD spectrums sensing methods respectively.

5.2.2. Data Analysis and Feature Engineering

A spectrum measurement campaign conducted in the Ethio-telecom GSM900 MHz 2G mobile network uplink band at four different regional cities in October 2021 under the project "**Digital Transformation for Ethiopia**" was used as supplementary data in this study shows the underutilization of the spectrum has an average utilization of 21.45% in Adama, 16.21% in Bahir Dar, 18.87% in Hawassa, and 33.52% in Jijiga.

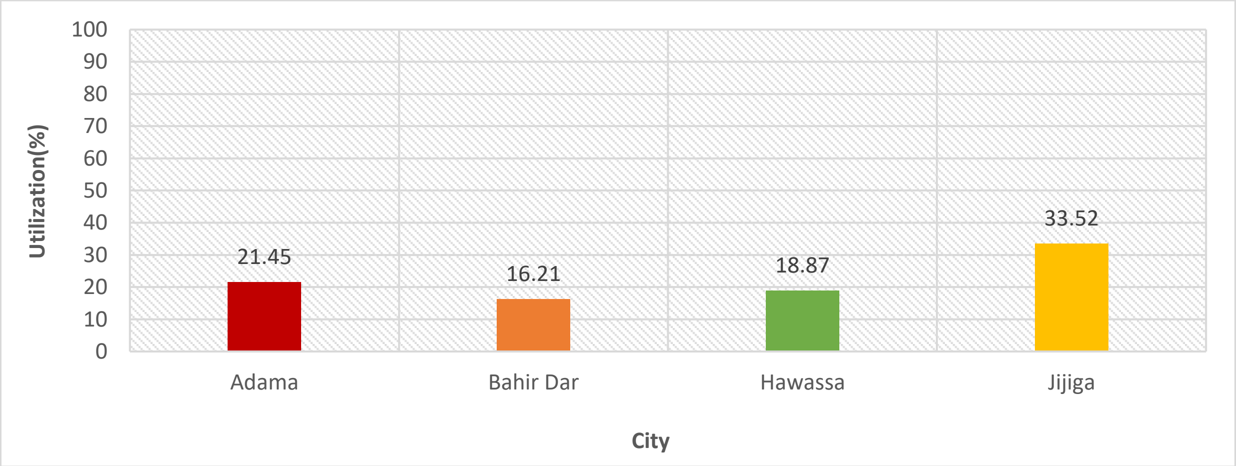


Figure 5.1 GSM900 MHz uplink band average spectrum utilization in four regional cities

The same spectrum measurement campaign was also conducted in Addis Abeba within four different mobile network uplink bands owned by Ethio-telecom.

The results showed the following average spectrum utilization values in the four spectrum bands: 14.72% in GSM 900MHz, 31.67% in UMTS 900MHz, 25.32% in LTE 1800MHz, and 6.75% in LTE 2600MHz, which was also underutilized.

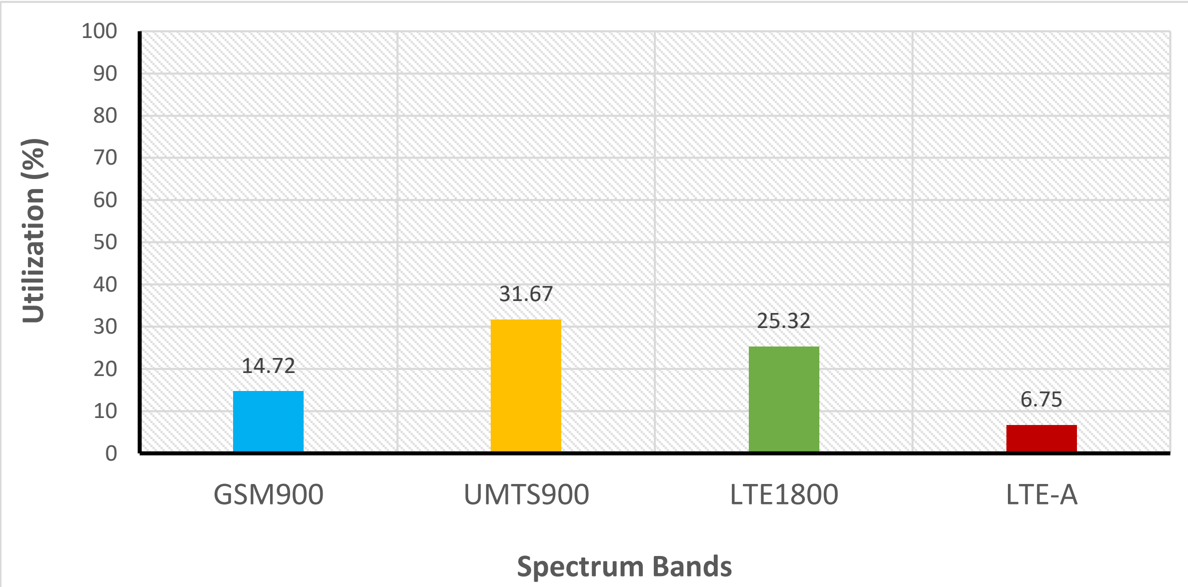


Figure 5.2 Addis Abeba City average spectrum utilization in four spectrum uplink bands

The result of the spectrum utilization analysis in the cities mentioned above indicates that the spectrum resource is underutilized. Even though all of the spectrum bands are now underutilized, subscribers would migrate from 2G to 3G, 4G, and 5G to get the latest technology features and services through the load-handling capacity of these new-generation mobile networks. The 900 MHz GSM 2G is the first mobile network in Ethiopia that offers only voice and short message services (SMS) [40]. Therefore, the 900 MHz 2G mobile network uplink band will become more underutilized because of the subscribers' migration, making it a potential candidate for implementing a CR.

The CFD-based spectrum sensing method was employed in this study to characterize primary user (PU) channels and model spectrum occupancy prediction due to its superior performance in challenging SNR conditions. Over five days, spectrum measurements using CFD-based spectrum sensing achieved an average utilization rate of 20.47%.

The CFD-based spectrum sensing method was used in this study for defining PU channel characterization and modeling spectrum occupancy prediction due to its enhanced performance in challenging SNR conditions. In a five-day spectrum measurement campaign conducted using the CFD spectrum sensing method, the average spectrum utilization for the GSM 900 MHz uplink band was 20.47%.

The spectrum utilization analysis conducted in five days revealed distinct values for weekdays (Thursday, Friday, and Monday) and weekends (Saturday and Sunday). Specifically, the average spectrum utilization on weekdays was 19.13%, 19.03%, and 21.14%, respectively. In contrast, the average spectrum utilization on weekends exhibits a variation, with values of 17.3% on Saturday and a higher utilization rate of 25.76% on Sunday. This observed pattern suggests that spectrum usage tends to be lower on Saturdays and higher on Sundays. This difference can be attributed as the fact that Sunday is a holiday in Ethiopia.

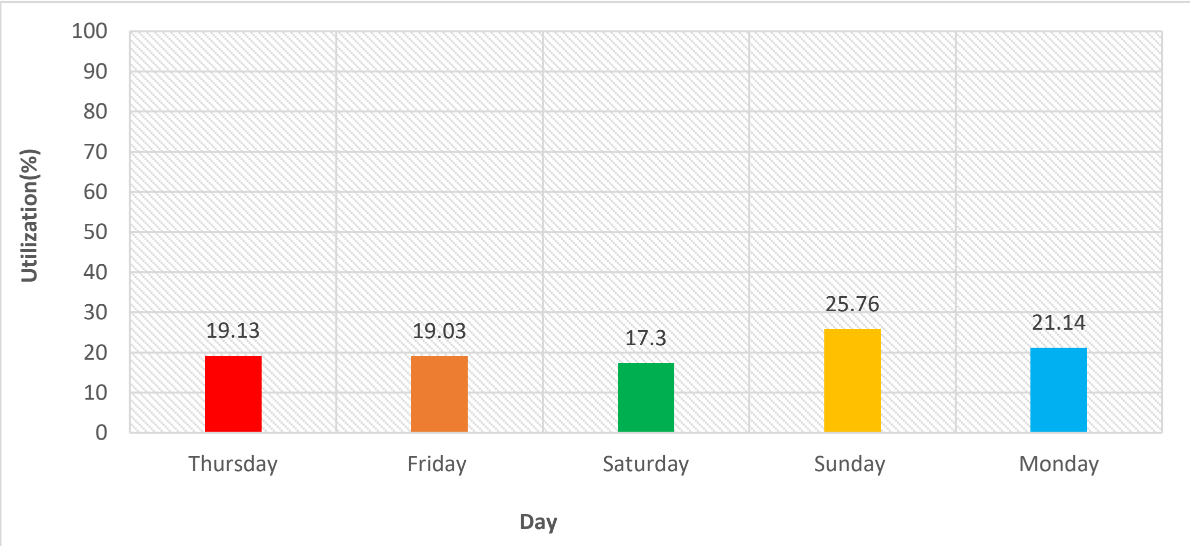


Figure 5.3 GSM900 spectrum utilization percentage in five different days

An in-depth analysis was performed on spectrum utilization by dividing the day into four equal six-hour periods to get detailed insights into the variation of spectrum usage at different times.

The result indicates a utilization rate of 8.27% during the night (00:00:00 - 06:00:00), 23.69% in the morning (06:00:00 - 12:00:00), 28.67% during the day (12:00:00 - 18:00:00), and 21.06% in the evening (18:00:00 - 00:00:00). The utilization has a visible difference over time, with the lowest utilization at night and the highest during the day.

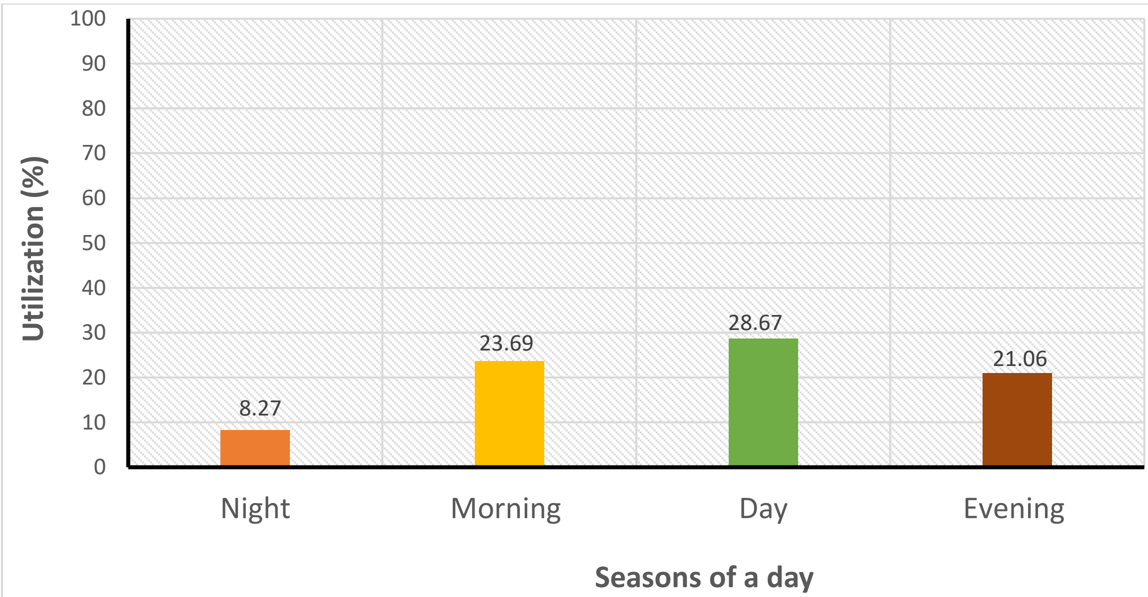


Figure 5.4 GSM900 spectrum utilization percentage in four different seasons of a day

The spectrum utilization can be further effectively represented through a spectrogram, as depicted in Figure 5.7. This visual depiction shows the frequency spectrum usage pattern over time. A spectrogram illustrates the relationship between duty cycle and frequency, with line brightness indicating the duty cycle for each corresponding frequency in the frame [41]. The presented figures in the spectrogram cover a comprehensive dataset collected over five days, considering the frequency range of 902.5-915 MHz. The x-axis denotes time, while the y-axis represents frequency in the spectrum occupancy plots. The depicted information reveals the spectrum utilization over the five days, indicating that the entire spectrum band experiences underutilization.

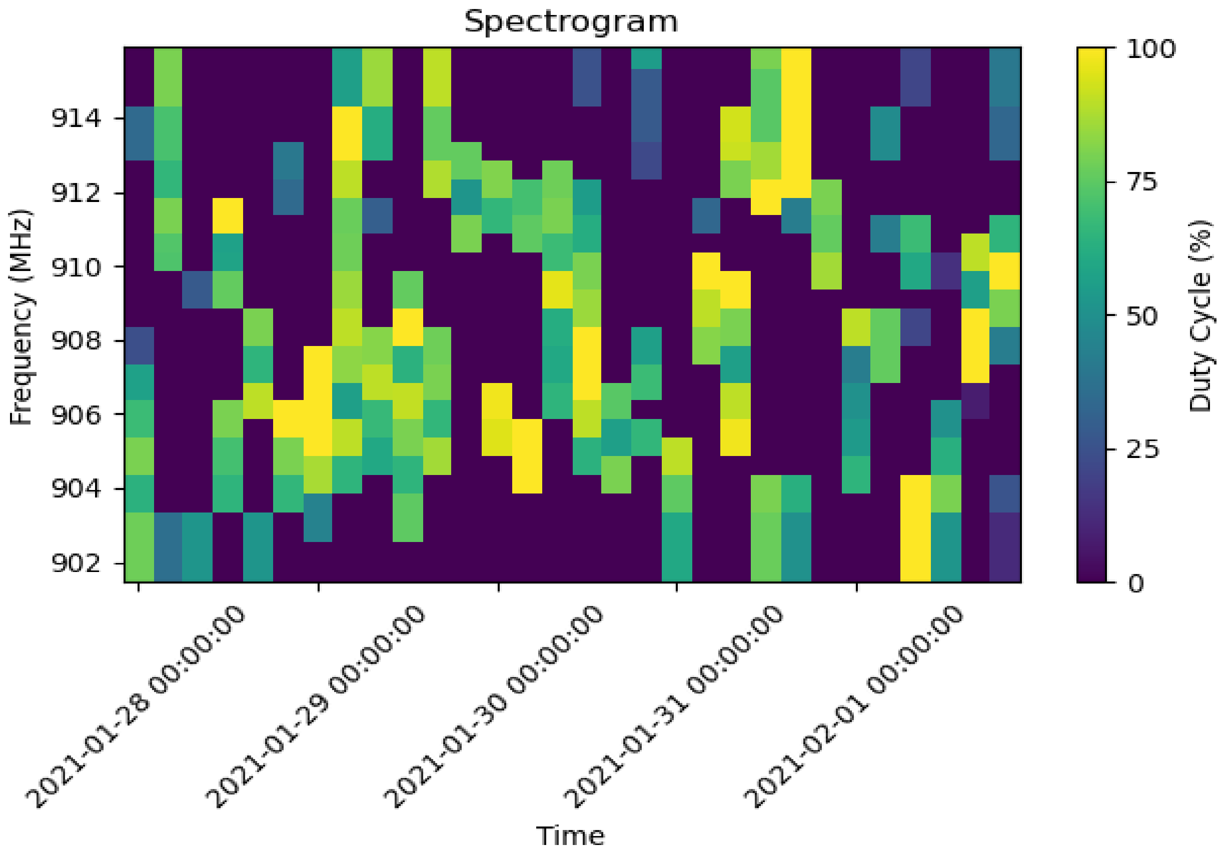


Figure 5.5 Spectrogram of the 902.5-915 MHz for the CFD based spectrum sensing

5.2.3. Preparing The Dataset

The original dataset had too many unnecessary features for the spectrum occupancy state prediction. Since then, irrelevant features that make the model complex and not necessary for the spectrum occupancy state prediction have been removed. The proposed model now utilizes a dataset generated from the CFD-based spectrum sensing, chosen due to its reliability-enhanced performance in challenging SNR environment conditions.

1	Date; Time	902.6	902.7	902.8	902.9	903	903.1			914.5	914.6	914.7	914.8	914.9	915
2	01/28/2021; 00:04:00	0	0	0	0	0	0			0	0	0	0	0	0
3	01/28/2021; 00:08:00	0	0	0	0	0	0			0	0	1	1	0	0
4	01/28/2021; 00:12:00	0	0	0	0	0	0			60	0	0	0	0	0
5	01/28/2021; 00:16:00	0	0	0	0	0	0			0	0	0	0	0	0
6	01/28/2021; 00:20:00	0	0	0	0	0	0			0	0	0	0	0	0
7	01/28/2021; 00:24:00	0	0	0	0	1	1			0	0	0	0	0	0

…

1796	02/01/2021; 23:40:00	0	0	0	0	0	0			0	0	0	0	0	0
1797	02/01/2021; 23:44:00	0	0	0	0	0	0			0	0	0	0	0	0
1798	02/01/2021; 23:48:00	0	0	0	0	0	0			0	0	0	0	0	0
1799	02/01/2021; 23:52:00	0	0	0	0	0	0			0	0	0	0	0	0
1800	02/01/2021; 23:56:00	0	0	1	1	0	0			0	0	0	0	0	0
1801	02/02/2021; 00:00:00	0	0	0	0	1	1			0	0	0	0	0	0

Figure 5.6 The dataset used for the spectrum occupancy state prediction model

5.2.4. Creating the Model

The Encoder-Decoder LSTM, a specialized type of RNN model, is designed to address sequence-to-sequence problems in spectrum occupancy state prediction. This model handles multiple time steps as both input and output data, making it suitable for many-to-many sequence prediction problems [41]. Consequently, the Encoder-Decoder LSTM model has proven to be highly effective for sequence-to-sequence prediction tasks, enabling multiple steps-ahead forecasting in spectrum occupancy state prediction.

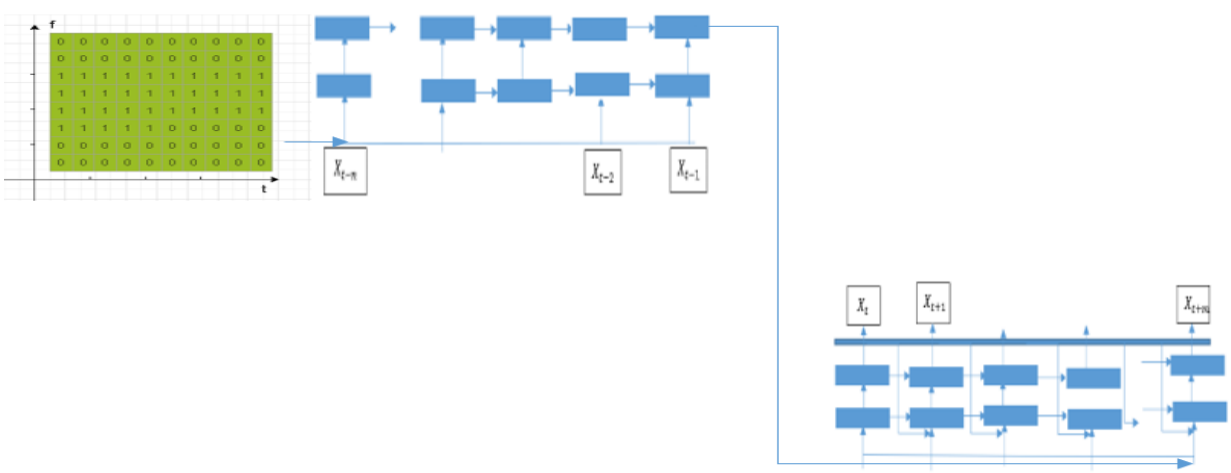


Figure 5.7 Learning and predicting the long-term spectrum occupancy in the proposed LSTM network with sequence-to-sequence architecture

5.3. Implementation Details

5.3.1. Working Environment and Tools

The following tools were used for design and implementation in this study.

Table 5.3 The hardware and software specifications used to conduct the experiments

Specification of Machine Used for Experiments	
Manufacturer	HP
System Model	HP ZBook Fury 17 G7 Mobile Workstation
Processor	Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz,
GPU	NVIDIA Quadro T1000 with Max-Q Design
SSD	Configured SSD is 1TB
Memory	16GB DDR4
Operating System	Microsoft Windows 11 Pro (64- bit)
Software	MATLAB R2021a and Python 3.10.9

5.3.2. Dataset

This study utilizes the data acquired through the CFD-based spectrum sensing method to define the characteristics of PU channels and develop a predictive model for the spectrum occupancy state. The dataset encompasses 225,000 data points, representing half of the five-day measurement data. Subsequently, this dataset is divided into training and validation sets, maintaining an 80% to 20% ratio. Specifically, 180,000 data points are earmarked for training purposes, while the remaining 45,000 are allocated for validating the predictive model. The selection of this split is driven by the available data for this study, aiming to provide a substantial number of training samples and validation data to ensure the effective generalization and robust evaluation of the trained model.

5.3.3. Experiment Setup with

The spectrum occupancy state prediction model was implemented using Python programming language and the Keras library, specifically using a DL algorithm. Training and validation experiments were conducted to ensure optimal model performance. These experiments involved a comprehensive exploration of hyperparameter values, specifically focusing on the learning rate, the number of hidden units, and the dropout rate.

The model designed for learning the long-term time-series data dependencies in spectrum occupancy state prediction was structured as a sequential model comprising seven layers. The model architecture comprises three LSTM layers with 128 units, followed by two dropout layers with a dropout rate of 0.1. Additionally, two dense layers with 128 and 64 units, were incorporated, along with a final output layer. The model was configured with an activation function of rectified linear unit (ReLU) for all hidden layers and sigmoid for the output layer, adaptive moment estimation (ADAM) as an optimizer, binary cross-entropy as a loss function, 0.001 learning rate, 128 batch size, and 400 Epochs. The model configuration was systematically evaluated based on key metrics like loss function and accuracy. Various combinations of hyperparameters were vigorously tested to minimize the loss percentage while enhancing the accuracy to identify the most effective model. Hyperparameter values were determined depending on the results of the iterative experiments. Therefore, the selection of the hyperparameters was guided by the dual goals of minimizing the loss function percentage and improving the accuracy simultaneously. The final spectrum occupancy state prediction model was chosen based on its consistent demonstration of the lowest loss and highest accuracy across the iterative experimentation phases. During the training phase, a dynamic scheduler plays a pivotal role in enhancing the model's performance. The scheduler continuously monitors the model's losses and dynamically adjusts hyperparameters to optimize prediction accuracy. The adaptive nature of the scheduler ensures that the learning rate is tuned automatically, reflecting a proactive approach to improving the model's accuracy.

Three distinct deep learning models, namely LSTM, Bi-LSTM, and ConvLSTM, were employed to compare the performance of spectrum occupancy state prediction, and three sets of experiments were conducted. The first set of experiments focused on short-term prediction, where the objective was to predict spectrum occupancy state one step ahead. In the second set of experiments, the focus shifted to long-term prediction. The models predicted the spectrum occupancy state for three and five hours ahead. This required the systems to forecast the spectrum occupancy patterns over extended time horizons, providing insights into the system's ability to capture and generalize temporal dependencies over a more extended period. These experiments were aimed to evaluate and compare the effectiveness of the models in different prediction scenarios.

5.4. Results of the Study

The models were evaluated using the performance evaluation metrics mentioned in 3.6 and 3.7. The results obtained in each model have been presented in the below sections from 5.4.1 to 5.4.3.

5.4.1. Spectrum Occupancy Prediction Using the LSTM Model

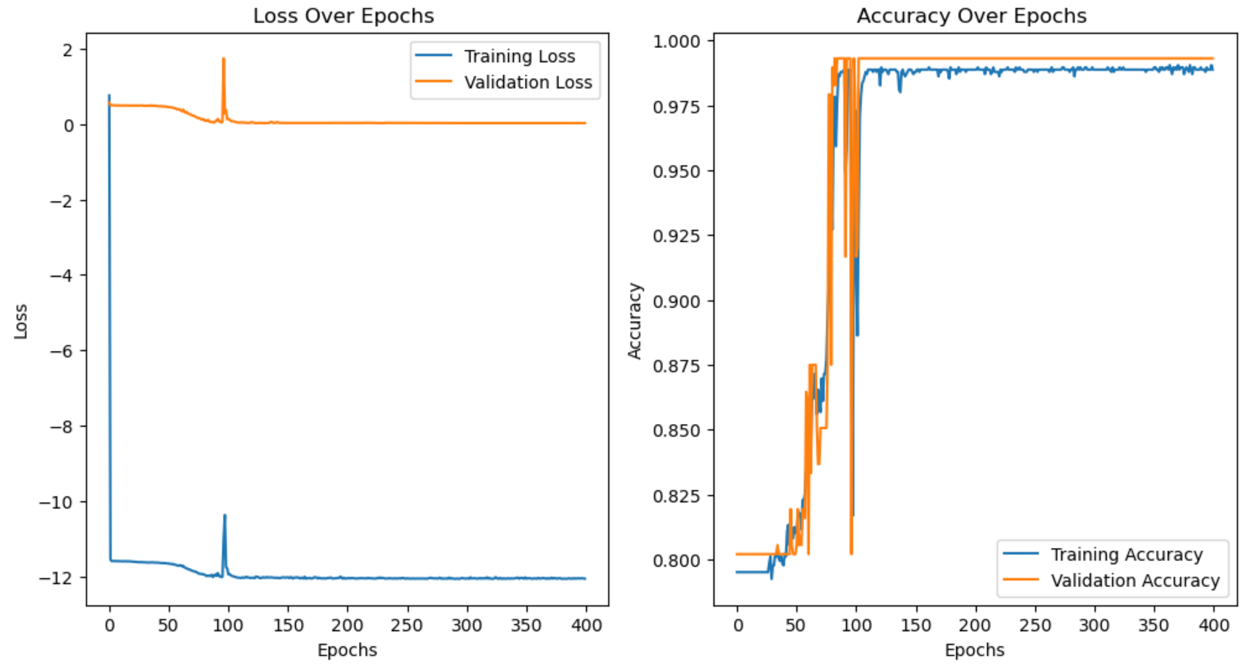


Figure 5.8 Training and validation, loss and accuracy of the LSTM model short-term prediction

Table 5.4 Performance results of the LSTM model for short-term prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

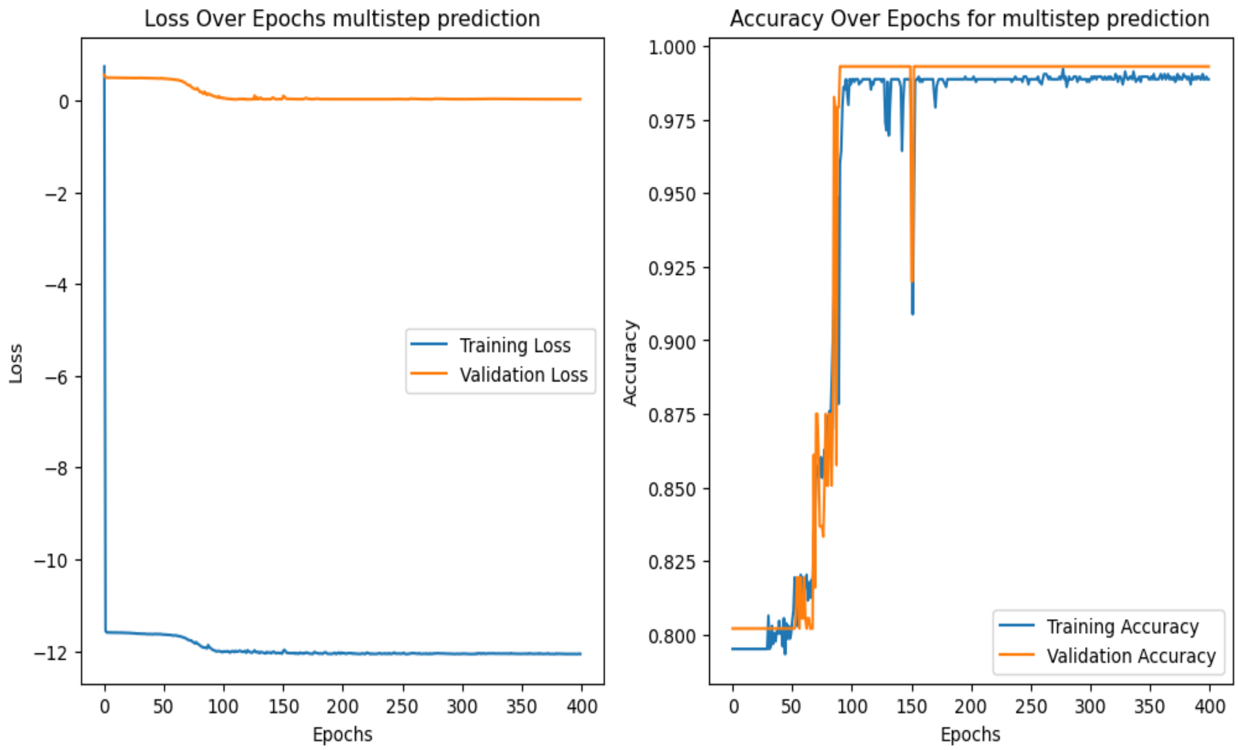


Figure 5.9 Training and validation, loss and accuracy of the LSTM model for 3-hours ahead prediction

Table 5.5 Performance results of the LSTM model for 3-hours ahead prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

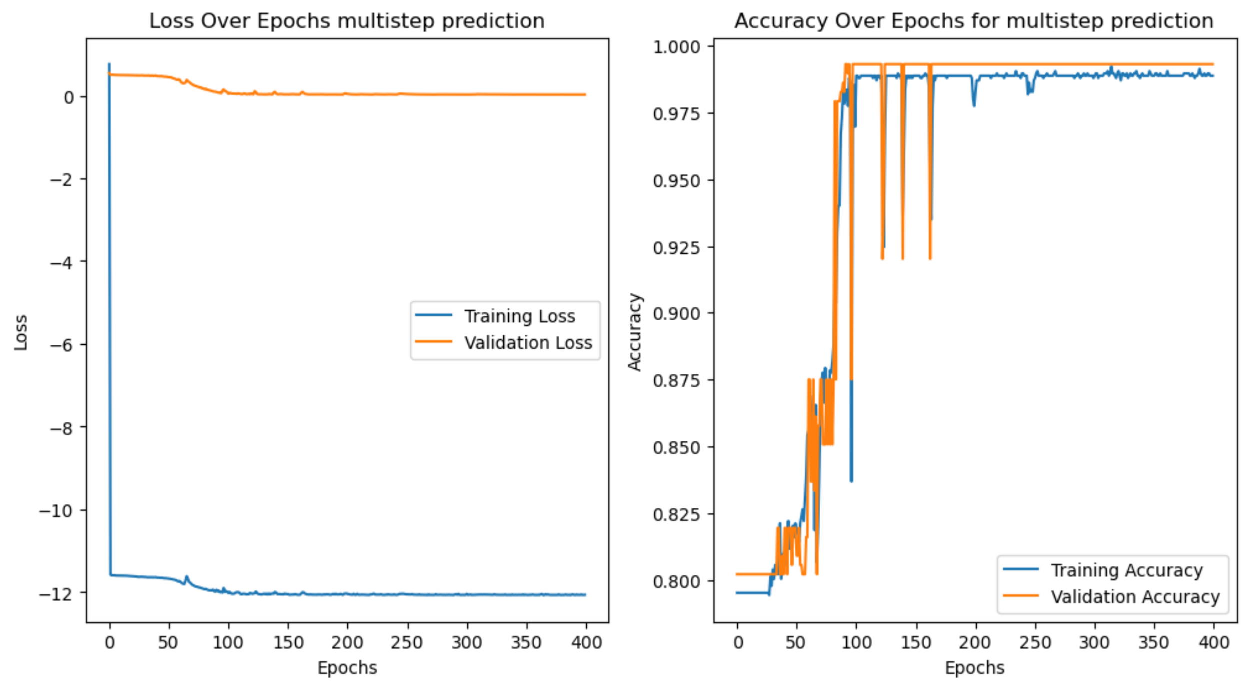


Figure 5.10 Training and validation, loss and accuracy of the LSTM model for 5-hours ahead prediction

Table 5.6 Performance results of the LSTM model for 5-hours ahead prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

The performance results presented in Tables 5.4 to 5.6 demonstrate that the spectrum occupancy state predictions obtained from the LSTM model consistently exhibit equal results across all evaluated metrics.

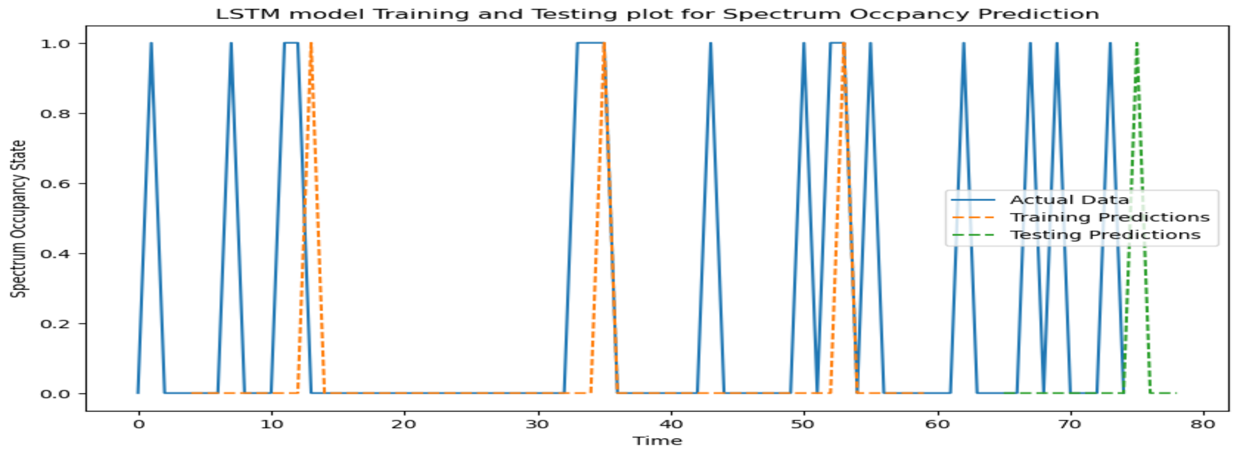


Figure 5.11 Spectrum occupancy state prediction in LSTM for the 904.1MHZ channel

Figure 5.11 compares the spectrum occupancy state actual data, training, and testing prediction performed for the 904.1MHZ channel. The LSTM network is trained to adapt new spectrum occupancy states, as shown in Figure 5.11, using the five-day spectrum measurement data for a one-channel 904.1MHz. In the graph, the brown dotted line representing the training data and the green dotted line representing the testing data closely match the actual observations depicted by the blue solid line. The test performance indicates an accuracy of 96%.

5.4.2. Spectrum Occupancy Prediction Using the Bi-LSTM Model

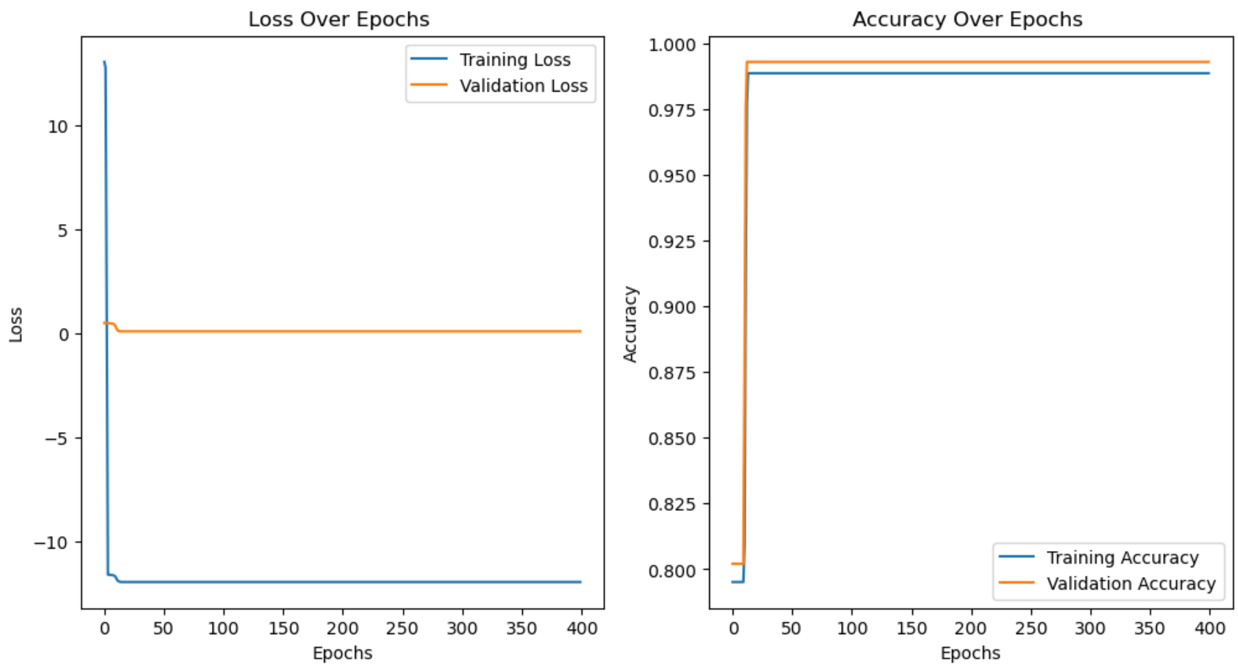


Figure 5.12 Training and validation, loss and accuracy of the Bi-LSTM model short-term prediction

Table 5.7 Performance results of the Bi-LSTM model for short-term prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

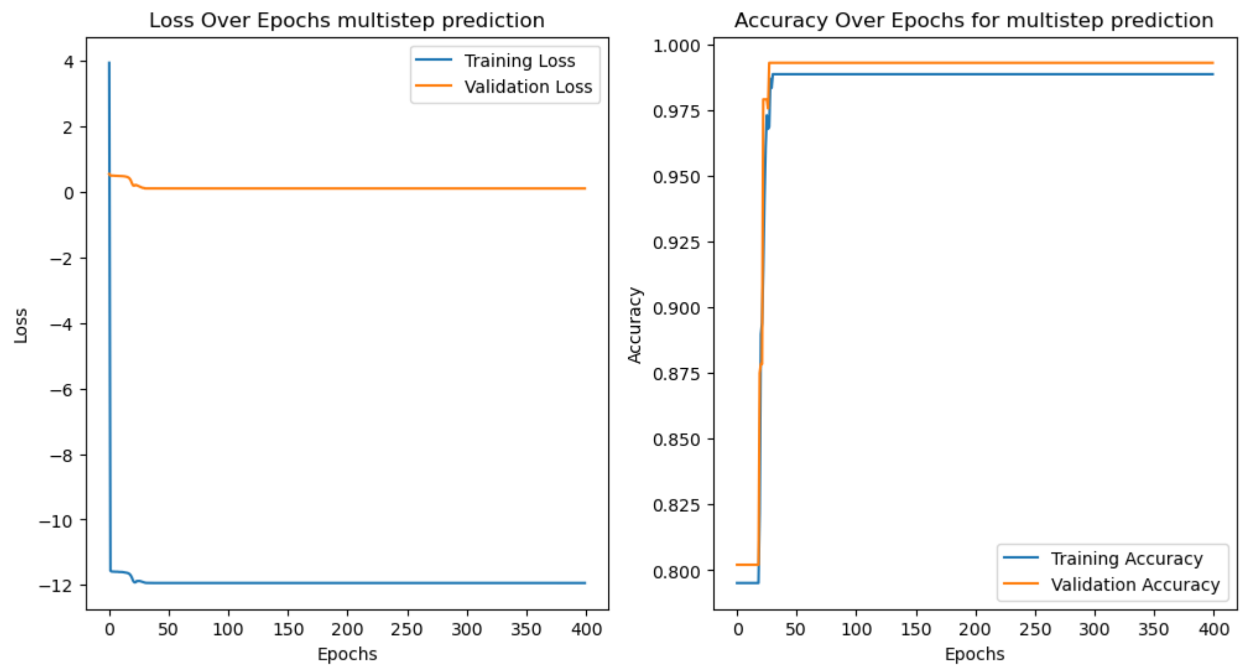


Figure 5.13 Training and validation, loss and accuracy of the Bi-LSTM model for 3-hours ahead prediction

Table 5.8 Performance results of the Bi-LSTM model for 3-hours ahead prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

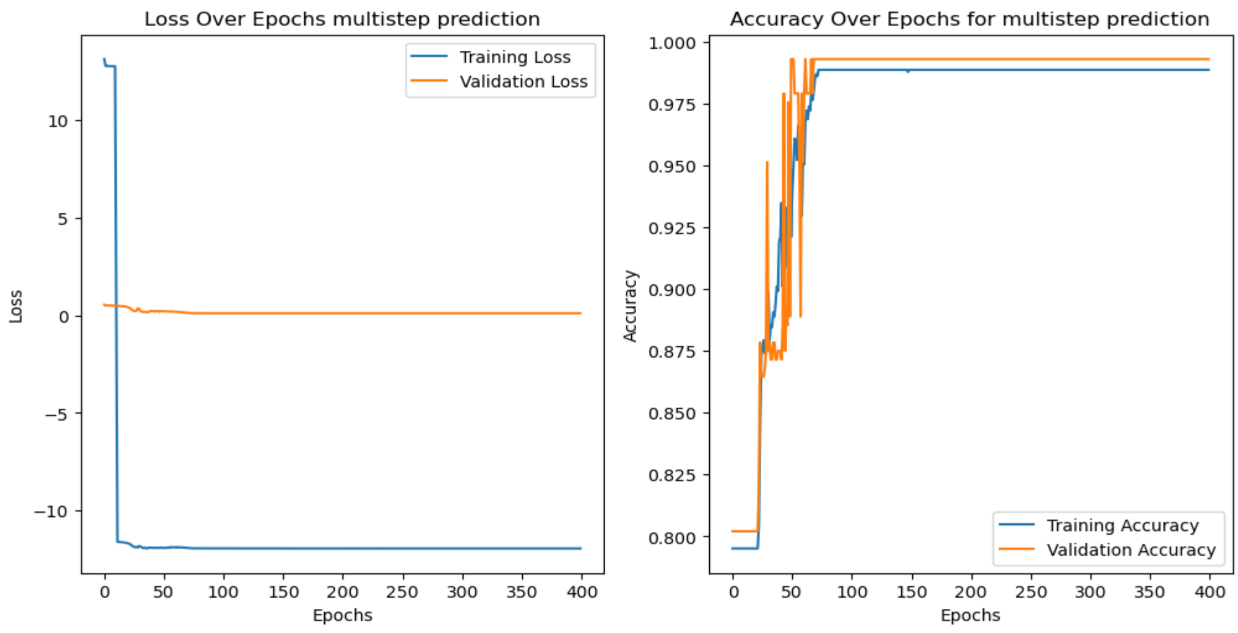


Figure 5.14 Training and validation, loss and accuracy of the Bi-LSTM model for 5-hours ahead prediction

Table 5.9 Performance results of the Bi-LSTM model for 5-hours ahead prediction

Metric	Value
Accuracy	0.99446
Precision	0.97619
Recall	1
F1-Score	0.987952
Specificity	0.992832
MSE	0.00554017
RMSE	0.0744323
MAE	0.00554017
MAPE	0.554017

The performance results presented in Tables 5.7 to 5.9 demonstrate that the spectrum occupancy state predictions obtained from the Bi-LSTM model consistently exhibit equal results across all evaluated metrics.

5.4.3. Spectrum Occupancy Prediction Using the ConvLSTM Model

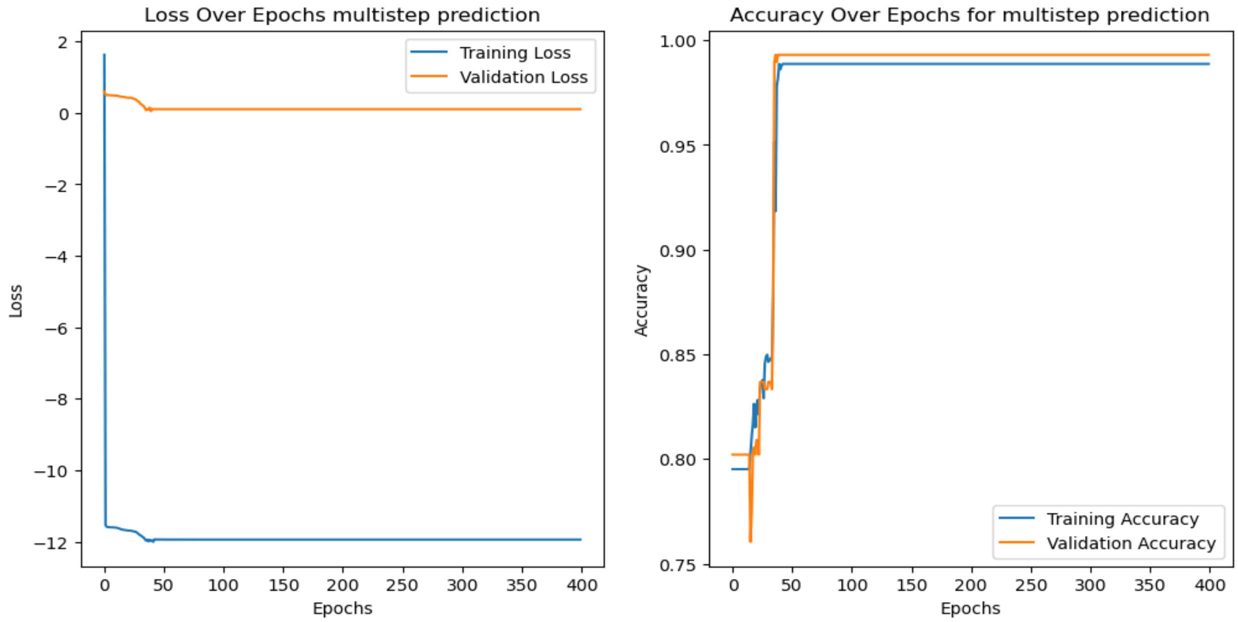


Figure 5.15 Training and validation, loss and accuracy of the ConvLSTM model for short-term prediction

Table 5.10 Performance results of the ConvLSTM model for short-term prediction

Metric	Value
Accuracy	0.99723
Precision	0.987952
Recall	1
F1-Score	0.993939
Specificity	0.996416
MSE	0.00277008
RMSE	0.0526316
MAE	0.00277008
MAPE	0.277008

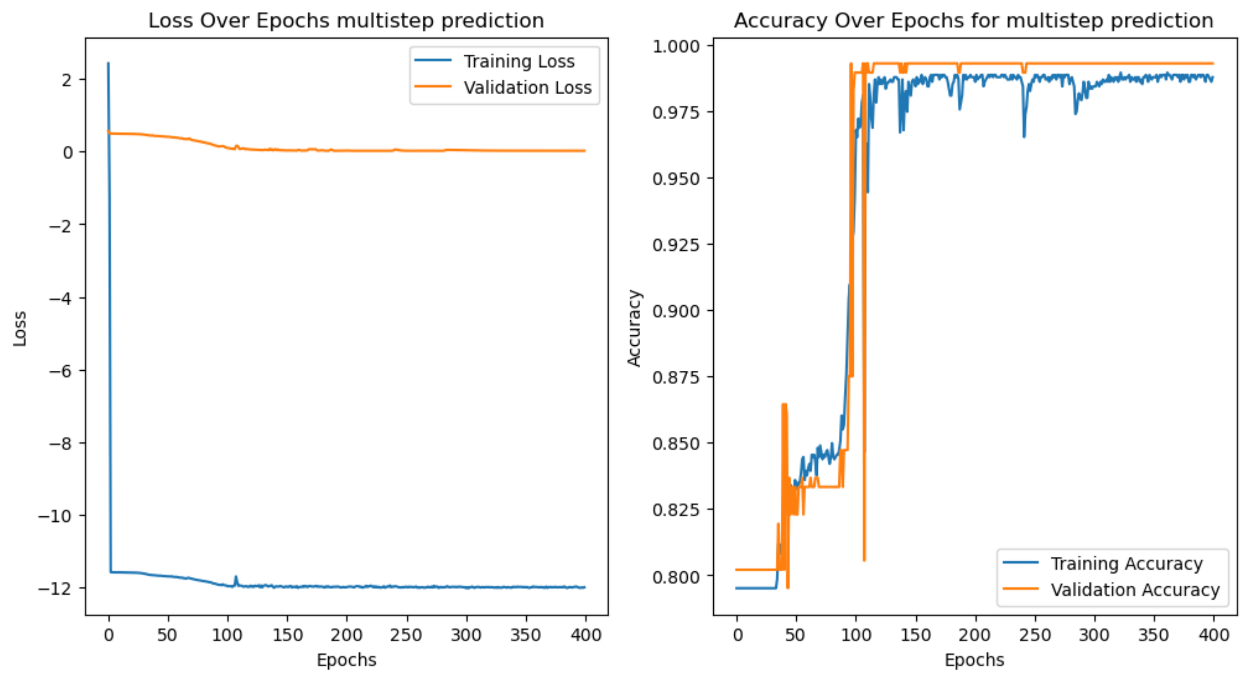


Figure 5.16 Training and validation, loss and accuracy of the ConvLSTM model 3-hours ahead prediction

Table 5.11 Performance results of the ConvLSTM model for 3-hours ahead prediction

Metric	Value
Accuracy	0.99723
Precision	0.987952
Recall	1
F1-Score	0.993939
Specificity	0.996416
MSE	0.00277008
RMSE	0.0526316
MAE	0.00277008
MAPE	0.277008

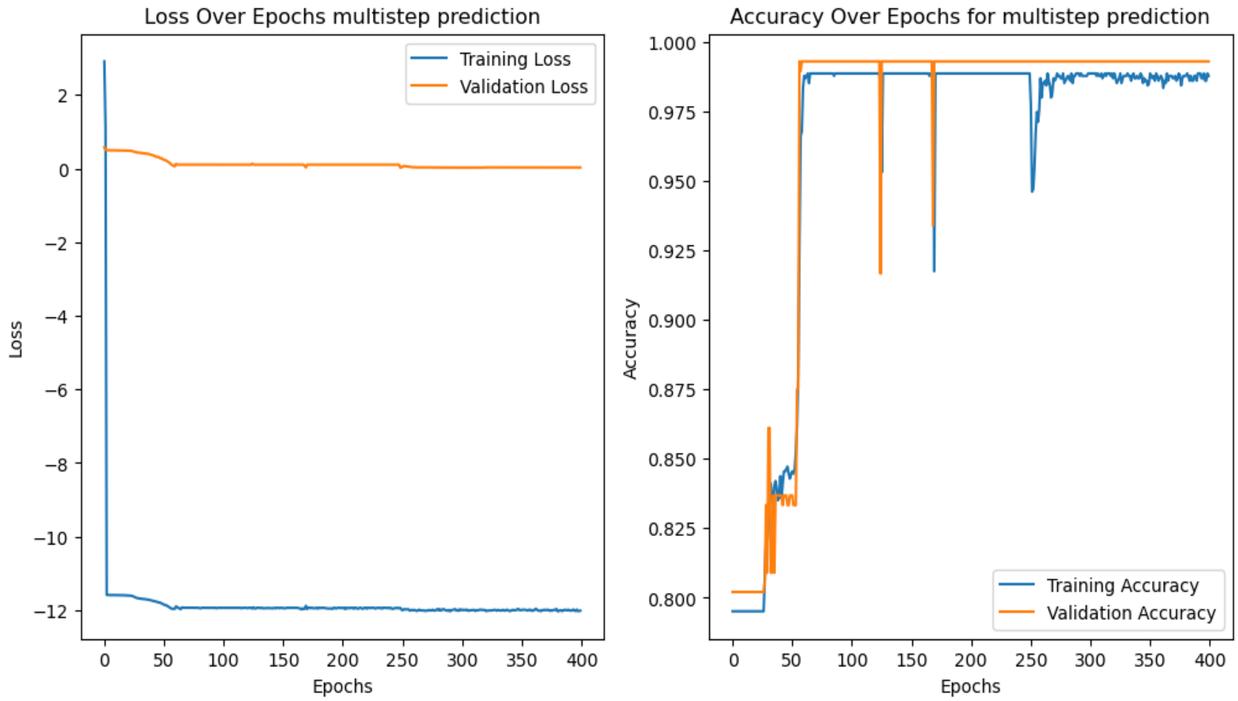


Figure 5.17 Training and validation, loss and accuracy of the ConvLSTM model for 5-hours ahead prediction

Table 5.12 Performance results of the ConvLSTM model for 5-hours ahead prediction

Metric	Value
Accuracy	0.99723
Precision	0.987952
Recall	1
F1-Score	0.993939
Specificity	0.996416
MSE	0.00277008
RMSE	0.0526316
MAE	0.00277008
MAPE	0.277008

The performance results presented in Tables 5.10 to 5.12 demonstrate that the spectrum occupancy state predictions obtained from the ConvLSTM model consistently exhibit equal results across all evaluated metrics.

5.5. Discussion

5.5.1. Quantifying Improvements

The long-term spectrum occupancy state prediction model used to implement a CRN in this study has improved spectrum utilization and reduced sensing energy.

1. **Improvement in Spectrum Utilization**

The long-term spectrum occupancy prediction model improves spectrum utilization by allowing SUs to select and use idle PU channels. This allows the SUs to choose the appropriate channels efficiently to reduce channel-switching costs and increase network throughput. In a CRN the PUs have two states, but the SUs can sense only one channel at a time. The CRN has two types of SUs the CRsense and the CRpredict. The CRsense randomly selects a channel at every slot and senses the state of the channel, while the CRpredict senses only channels with an idle prediction state. According to **[4]**, spectrum utilization (SU) in the CRN can be defined as the ratio of the number of idle slots correctly identified and utilized by the SUs to the total number of idle slots available in the CRN.

$$SU = \frac{\text{Number of idle slots sensed}}{\text{Total number of idle slots in the band}} \quad 5.1$$

The improvement in spectrum utilization due to spectrum prediction can be expressed as

$$SU_{imp(\%)} = \frac{SU_{sense} - SU_{predict}}{SU_{sense}} \quad 5.2$$

Where SU_{sense} and $SU_{predict}$ represent the spectrum utilization for the CRsense and CRpredict, respectively. Substituting (5.1) in (5. 2), $SU_{imp(\%)}$ can be given by

$$SU_{imp(\%)} = \frac{I_{sense} - I_{predict}}{I_{sense}} \quad 5.3$$

Where I_{sense} and $I_{predict}$ represent the number of idle channels sensed by the CRsense and the number of idle channels predicted by the CRpredict respectively.

This analysis can be translated into a machine learning model and becomes equal with the specificity that measures the true negative rate, which is the fraction of negative values that were correctly predicted and can be calculated as expressed in equation 5.4.

$$SU_{imp(\%)} = Specifcity = \frac{TN}{TN+FP} \qquad 5.4$$

The $SU_{imp(\%)}$ improves the network throughput which shows the data rate achieved in the network due to the availability of more channels and can be calculated as expressed in equation 5.5.

Throughput = $SU_{imp(\%)}$* the number of channels in the spectrum band 5.5

2. Reduction in Sensing Energy

The prediction model reduces the sensing energy required by the SUs, because the SUs senses only the idle channels. The CRsense senses all the channels whereas the CRpredict only senses the channel that is predicted idle. In other words, when the channel state is predicted to be busy, the sensing operation is not performed to reduce energy. Assuming that one unit of sensing energy is required to sense one slot [4], then the total sensing energy required for a CRsense in a finite duration of time can be calculated as expressed in equation 5.6.

$$\mathrm{SE}_{\mathrm{sense=(Toatal\ number\ of\ slots\ in\ the\ duration)*(unit\ sensing\ \ enegry)}} \qquad 5.6$$

while the total sensing energy required by the CRpredict can be given by

$$\mathrm{SE}_{\mathrm{predict}} = \left(\mathrm{SE}_{\mathrm{sense}} - \left(\mathrm{B}_{\mathrm{preidct}}\right)\right) * \ \ (\mathrm{Unit\ sensing\ energy}) \qquad 5.7$$

Where $B_{preidct}$ is the total number of busy slots predicted by the CRpredict.

Therefore, using (5.6) and (5.7), the percentage reduction in the sensing energy can be given by

$$SE_{red(\%)} = \frac{SE_{sense} - SE_{predict}}{SE_{sense}} = \frac{B_{predict}}{Total\ no.of\ idle\ slots} \qquad 5.8$$

This can be translated to a machine learning model that measures a value by dividing the true positive value by the true negative plus the false positive values, even it doesn’t have an equivalent machine learning metric it can be calculated and expressed as shown in equation 5.9.

$$SE_{red(\%)} = \frac{TP}{TN+FP} \quad 5.9$$

The improvements achieved in spectrum utilization, the network throughput obtained, and the reduction of sensing energy on three models for all predictions results calculated based on equations 5.4, 5.5, and 5.9 are presented in the above Table 5.13.

Table 5.13 Quantified improvements achieved in the spectrum occupancy state prediction

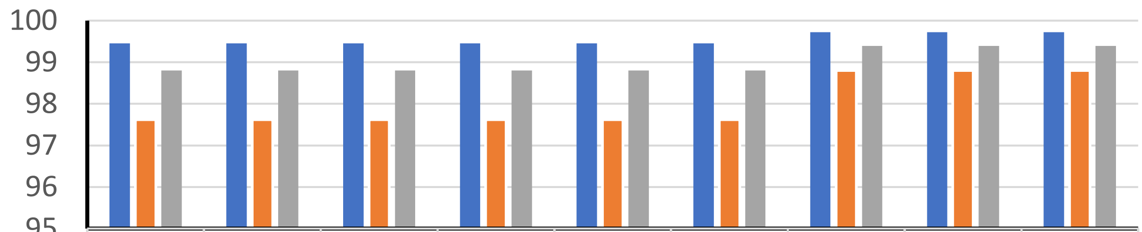
Model	**Length of Prediction**	$\boldsymbol{SU_{imp}}(\%)$	**Throughput**	$\boldsymbol{SE_{red}}(\%)$
LSTM	Short-Term	99.28	124.1	29.39
	Long-Term (3 hrs.)	99.28	124.1*45	29.39
	Long-Term (5 hrs.)	99.28	124.1*75	29.39
Bi- LSTM	Short-Term	99.28	124.1	29.39
	Long-Term (3 hrs.)	99.28	124.1*45	29.39
	Long-Term (5 hrs.)	99.28	124.1*75	29.39
Conv-LSTM	Short-Term	99.64	124.55	29.39
	Long-Term (3 hrs.)	99.64	124.55*45	29.39
	Long-Term (5 hrs.)	99.64	124.55*75	29.39

5.5.2. Comparison of Results

Across all three models, each term of prediction has equal performance results. However, the short-term prediction achieved its targeted accuracy earlier than the long-term 3-hour and 5-hour predictions. The 3-hour long-term prediction achieved its targeted accuracy earlier than the 5-hour long-term prediction. The Bi-LSTM model differs from the others by achieving its targeted accuracy earlier across all prediction scenarios. This may be attributed to its ability to process data in both forward and backward directions, potentially improving its ability to learn temporal relationships. Consequently, short-term predictions tend to achieve better accuracy earlier than long-term predictions. This tendency may arise from factors such as predicting the immediate future requires fewer data points and simpler relationships, and the available data for training the model can be enough for short-term prediction than for long-term prediction.

In the comparative analysis performed on the spectrum occupancy prediction conducted on LSTM, Bi-LSTM, and ConvLSTM models, performance evaluation metrics outlined in sections 3.6 and 3.7 were employed.

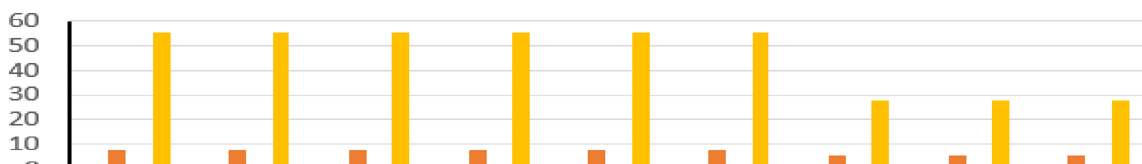
The comparative analysis is visually represented in Figures 6.20 to 6.21. Notably, both LSTM and Bi-LSTM exhibited identical results achieving 99.45% accuracy, 97.62% precision, 98.8% F1-Score, MSE 0.554017%, RMSE 7.44323%, MAE 0.554017%, and MAPE of 55.4017%. However, the ConvLSTM model outperformed that achieved 99.72% accuracy, 98.8% precision, and 99.39 F1-Score, MSE 0.277008%, RMSE 5.26316%, MAE 0.277008%, and a MAPE of 27.7008%.



	LSTM			Bi-LSTM			ConvLSTM		
	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)
Accuracy	99.45	99.45	99.45	99.45	99.45	99.45	99.72	99.72	99.72
Precision	97.62	97.62	97.62	97.62	97.62	97.62	98.8	98.8	98.8
F1-Score	98.8	98.8	98.8	98.8	98.8	98.8	99.39	99.39	99.39

Accuracy Precision F1-Score

Figure 5.18 Comparison of models in Accuracy, Precision, and F1-Score



	LSTM			Bi-LSTM			ConvLSTM		
	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)	Short-Term	Long-Term (3hrs)	Long-Term (5hrs)
MSE	0.554017	0.554017	0.554017	0.554017	0.554017	0.554017	0.277008	0.277008	0.277008
RMSE	7.44323	7.44323	7.44323	7.44323	7.44323	7.44323	5.26316	5.26316	5.26316
MAE	0.554017	0.554017	0.554017	0.554017	0.554017	0.554017	0.277008	0.277008	0.277008
MAPE	55.4017	55.4017	55.4017	55.4017	55.4017	55.4017	27.7008	27.7008	27.7008

MSE RMSE MAE MAPE

Figure 5.19 Comparison of models in MSE, RMSE, MAE, and MAPE

Chapter Six

6. Conclusion and Future Work

6.1. Conclusion

The primary challenge in spectrum utilization arises from the rigidity of the FSA policy, which makes the underutilization of the available scarce spectrum resource. This study addresses the challenges posed by the rigidity of FSA policies and paves the way for more effective and efficient spectrum utilization that optimizes scarce spectrum resources. This study has improved spectrum utilization by predicting spectrum occupancy state using a long-term adaptive LSTM model and a historical dataset obtained through a reliable spectrum sensing method. Going beyond conventional prediction approaches, this study identified potential candidate spectrum bands through a comprehensive spectrum utilization and techno-economic analysis. The temporal data is input into the LSTM model through an input gate, enabling the network to capture and learn patterns over extended periods. The model undergoes extensive training steps comprising 400 epochs, each having 128 batches. The data passes over multiple layers of the LSTM model, each designed to extract and process relevant features for optimal prediction outcomes.

The comparative analysis performed on LSTM, Bi-LSTM, and ConvLSTM models reveals the proposed model has remarkable performances for spectrum occupancy state prediction. Although the proposed LSTM and the Bi-LSTM model exhibit strong performance, the ConvLSTM model stands out with a superior accuracy of 99.72%, surpassing the 99.45% accuracy achieved by the LSTM and Bi-LSTM models. The proposed model demonstrates notable accuracy evidenced by the low error values of 0.554017% MSE, 7.44323% RMSE, 0.554017% MAE, and 55.4017% MAPE. Employing five days' worth of historical spectrum measurement data, the model successfully predicted the spectrum occupancy state for the subsequent five hours with an accuracy of 99.45%. The low error values, indicating close alignment between the model's predictions and actual spectrum occupancy state, underscore the model's accuracy in predicting future spectrum occupancy states. Therefore, the proposed LSTM model and its counterparts are considered robust models for spectrum occupancy state prediction.

This study presented significant progress in enhancing the reliability of spectrum sensing and the accuracy of spectrum occupancy state prediction. Employing DL models for spectrum occupancy state prediction for the CRN has improved spectrum utilization boosts from a mere 20.47% to an impressive 98.28% and reduced the sensing energy by 29.39% compared to real-time sensing. This significant enhancement underscores the inefficiency of the traditional FSA policy that inadvertently leads to now artificial spectrum scarcity. Integrating DL models with reliable spectrum sensing methods and identifying potential candidate spectrum bands presents a promising avenue for augmenting the efficiency of deploying CRs. This synergy addresses the limitations of conventional approaches and unlocks new possibilities for optimizing spectrum utilization and overcoming challenges associated with spectrum scarcity. The study's findings suggest that embracing advanced technologies such as DL in the context of CR can pave the way for more adaptive and spectrum resource-efficient wireless communication technologies.

6.2. Future Work

Spectrum occupancy state prediction for deploying a CR becomes a foundation for application development like smart city. The applications of a smart city can be traffic management, efficient parking solutions, public safety, waste management, remote monitoring, environmental well-being, and improvements in public transport systems. Future studies can focus on enhancing the predictability of further occupancy lengths up to days for integrating CR with the Internet of Things, which creates a synergistic system known as the CR Internet of Things (CRIoT). This integrated IoT and CR amplifies smart cities' capability, providing a comprehensive and interconnected infrastructure for effective and efficient urban management.

Introducing a new technology, such as a smart city, into a specific area necessitates a comprehensive techno-economic analysis to assess the feasibility based on technical, economic, environmental, social, and legal criteria. The area of Bole, where spectrum measurement data collection has been conducted, emerges as a feasible location for implementing a smart city project. This feasibility is attributed to the infrastructural development in the Bole area, which enhances the practicality of conducting a techno-economic analysis compared to other parts of the city.

Reference

[1] J. Kalliovaara, "Field Measurements in Determining Incumbent Spectrum Utilization and Protection Criteria in Wireless Co-existence Studies," Turku University of Applied Sciences, 2017. [Online]. Available: DOI:10.13140/RG.2.2.33351.80805

[2] F. Weidling, D. Datla, V. Petty, P. Krishnan, and G. J. Minden, "A Framework for R.F. Spectrum Measurements and Analysis," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, 2005. [Online]. Available: DOI: 10.1109/DYSPAN.2005.1542672

[3] T.Jeacock, "A Standard Approach for Assessing the Spectrum Management Needs of Developing Countries," Report on Telecommunication Development Sectror: ITU, 2016. [Online]. Available: www.itu.int

[4] V. K. Tumuluru, P. Wang, and D. Niyato, "A Neural Network Based Spectrum Prediction Scheme for Cognitive Radio," in *IEEE International Conference on Communications*, 2010. doi: 10.1109/ICC.2010.5502348.

[5] L.J.H.Viveros, D.A.L.Sarmiento, and N.E.V.Parra, "Modeling and Prediction Primary Nodes in Wireless Networks of Cognitive Radio Using Recurrent Neural Networks," *Contemporary Engineering Sciences*, 2018, doi: 10.12988/ces.2018.84164.

[6] N.G.Bara'u, W.Feng, and M. Almustapha, "Spectrum Hole Prediction Based on Historical Data: A Neural Network Approach," *Computer Science, Engineering*, 2014, [Online]. Available: Corpus ID: 6991758

[7] A. Shenfield, Z. Khan, and H. Ahmadi, "Deep Learning Meets Cognitive Radio: Predicting Future Steps," in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020. [Online]. Available: http://shura.shu.ac.uk/25962/

[8] B. S. Shawel, D. H.Woledegebre, and S.Pollin, "Deep-learning based Cooperative Spectrum Prediction for Cognitive Networks," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018.

[9] A. Saad, B. Staehle, and R. Knorr, "Spectrum Prediction using Hidden Markov Models for Industrial Cognitive Radio," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016. [Online]. Available: DOI: 10.1109/WiMOB.2016.7763231

[10] T. Wysocki and B. J. Wysocki, “Spectrum Occupancy Prediction Using a Hidden Markov Model,” in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016. Accessed: Mar. 12, 2024. [Online]. Available: DOI: 10.1109/ICSPCS.2015.7391772

[11] N.Balwani, D.K.Patel, B.Soni, and M.L.Benítez, “Long Short-Term Memory based Spectrum Sensing Scheme for Cognitive Radio,” in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019. [Online]. Available: DOI: 10.1109/PIMRC.2019.8904422

[12] M. Sardana and A. Vohra, “Analysis of Different Spectrum Sensing Techniques,” in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017. [Online]. Available: DOI: 10.1109/COMPTELIX.2017.8004006

[13] P. S. Yawada and A. J. Wei, “Comparative Study of Spectrum Sensing Techniques Base on Techniques Non-cooperative in Cognitive Radio Networks,” in *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*, 2016. [Online]. Available: DOI: 10.1109/ICCSNT.2016.8070212

[14] M. Z. Alom, T. K. Godder, and M. N. Morshed, “A Survey of Spectrum Sensing Techniques in Cognitive Radio Network,” in *2015 International Conference on Advances in Electrical Engineering (ICAEE)*, 2015. [Online]. Available: DOI: 10.1109/ICAEE.2015.7506821

[15] V. Amrutha and K. V. Karthikeyan, “Spectrum Sensing Methodologies in Cognitive Radio Networks: A Survey,” in *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, 2017. [Online]. Available: DOI: 10.1109/ICIEEIMT.2017.8116855

[16] M. A. Aygül, M. Nazzal, M. İ. Sağlam, D. B. da Costa, H. F. Ateş, and H. Arslan, “Efficient spectrum Occupancy Prediction Exploiting Multidimensional Correlations through Composite 2D-LSTM Models,” *Sensors (Switzerland)*, 2020, doi: 10.3390/s21010135.

[17] N. Muchandi and R. Khanai, “Cognitive radio spectrum sensing: A survey,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016. [Online]. Available: DOI: 10.1109/ICEEOT.2016.7755301

[18] B. Soni, D. K. Patel, and M. L.Benitez, “Long Short-Term Memory based Spectrum Sensing Scheme for Cognitive Radio using Primary Activity Statistics,” *IEEE Access* , 2020. doi: 10.1109/ACCESS.2017.

[19] A. Sharmila and P. Dananjayan, "Spectrum Sharing Techniques in Cognitive Radio Networks-A Survey," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019. [Online]. Available: DOI: 10.1109/ICSCAN.2019.8878714

[20] G.Ding *et al.*, "Spectrum Inference in Cognitive Radio Networks: Algorithms and Applications," *IEEE Communications Surveys & Tutorials*, 2017. [Online]. Available: DOI: 10.1109/COMST.2017.2751058

[21] J. N. Javed, M. Khalil, and A. Shabbir, "A Survey on Cognitive Radio Spectrum Sensing: Classifications and Performance Comparison," in *2019 International Conference on Innovative Computing (ICIC)*, 2019. [Online]. Available: DOI: 10.1109/ICIC48496.2019.8966677

[22] B. G. Najashi and F. Wenjiang, "Cooperative Spectrum Occupancy based Spectrum Prediction Modeling," *Journal of Computational Information Systems*, 2014, doi: 10.12733/jcis10167.

[23] D. Das, D. W. Matolak, and S. Das, "Spectrum Occupancy Prediction based on Functional Link Artificial Neural Network (FLANN) in ISM Band," *Neural Comput Appl*, 2018, doi: 10.1007/s00521-016-2653-5.

[24] R. Fan, H. Guo, L. Di, and X. Ling, "Spectrum Occupancy State Predictor based on Recurrent Neural Network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, 2019. doi: 10.1088/1742-6596/1345/4/042020.

[25] W. Zhang, G. Huang, G. Wang, and Y. Wang, "Prediction High Frequency Parameters based on Neural Network," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1757-899X/631/5/052035.

[26] L. Xing, M. Li, Y. Wan, and Q. Wan, "Spectrum Prediction in Cognitive Radio Based on Sequence- to- Sequence Neural Network," in *International Conference on Advanced Hybrid Information Processing*, Springer, 2019, pp. 343–354. doi: 10.1007/978-3-030-36405-2_34.

[27] M.A.Aygül *et al.*, "Spectrum Occupancy Prediction Exploiting Time and Frequency Correlations Through 2D-LSTM," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020. [Online]. Available: DOI: 10.1109/VTC2020-Spring48590.2020.9129001

[28] L. Yu, Q. Wang, Y. Guo, and P. Li, "Spectrum Availability Prediction in Cognitive Aerospace Communications: A Deep Learning Perspective," in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, IEEE, 2017. [Online]. Available: DOI: 10.1109/CCAAW.2017.8001877

[29] L. Yu, J. Chen, and G. Ding, "Spectrum Prediction via Long Short- Term Memory," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017. [Online]. Available: DOI: 10.1109/CompComm.2017.8322623

[30] B. S. Shawel, D. H. Woldegebreal, and S. Pollin, "Convolutional LSTM-based Long-Term Spectrum Prediction for Dynamic Spectrum Access," in *2019 27th European Signal Processing Conference (EUSIPCO),* 2019. [Online]. Available: DOI: 10.23919/EUSIPCO.2019.8902956

[31] S. S. Shirgan and U. L. Bombale, "Hybrid Neural Network Based Wideband Spectrum Behavior Sensing Predictor for Cognitive Radio Application," *Sens Imaging*, Dec. 2020, doi: 10.1007/s11220-020-00293-4.

[32] L. Yin, S. Yin, W. Hong, and S. Li, "Spectrum Behavior Learning in Cognitive Radio based on Artificial Neural Network," in *2011 - MILCOM 2011 Military Communications Conference*, 2011. [Online]. Available: DOI: 10.1109/MILCOM.2011.6127671

[33] N.K. Chauhan and K.Singh, "A Review on Conventional Machine Learning vs Deep Learning," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018. [Online]. Available: DOI: 10.1109/GUCON.2018.8675097

[34] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature Selection in Machine Learning: A New Perspective," *Neurocomputing*, 2018, doi: 10.1016/j.neucom.2017.11.077.

[35] C. Janiesch, P. Zschech, and K.Heinrich, "Machine Learning and Deep Learning," *Electronic Markets*, 2021, doi: 10.1007/s12525-021-00475-2/Published.

[36] D. Zhou, X. Zuo, and Z. Zhao, "Constructing a Large-Scale Urban Land Subsidence Prediction Method Based on Neural Network Algorithm from the Perspective of Multiple Factors," *Remote Sens (Basel)*, Apr. 2022, doi: 10.3390/rs14081803.

[37] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3085085.

[38] Z. Wang and S. Salous, "Spectrum Occupancy Statistics and Time Series Models for Cognitive Radio," *J Signal Process Syst*, Feb. 2011, doi: 10.1007/s11265-009-0352-5.

[39] A. Agarwal, A. S. Sengar, and R. Gangopadhyay, "Spectrum Occupancy Prediction for Realistic Traffic Scenarios: Time Series versus Learning-Based Models," *Journal of Communications and Information Networks*, vol. 3, no. 2, Jun. 2018, doi: 10.1007/s41650-018-0013-6.

[40] B. B. H. D. Negash, "Techno-economic Analysis of LTE Deployment Scenarios for Emerging City: A Case of Adama, Ethiopia Addis Ababa, Ethiopia," in *International Conference on Information and Communication Technology for Development for Africa*, 2018. [Online]. Available: DOI:10.1007/978-3-030-26630-1_18

[41] A. Ghosh, S. Kasera, and J. V.Merwe, "Spectrum Usage Analysis And Prediction using Long Short-Term Memory Networks," in *Proceedings of the 24th International Conference on Distributed Computing and Networking*, 2023. [Online]. Available: https://doi.org/10.1145/3571306.3571412

Appendixes

Appendix A: Sample Code for Building, Creating, and Compiling the LSTM Model

```
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, mean_squared_error,
mean_absolute_error, confusion_matrix
import matplotlib.pyplot as plt
from tensorflow.keras.layers import LSTM, Dropout, Dense
from tabulate import tabulate
from tensorflow.keras.optimizers import Adam
colnames = ['Timestamp', '902.6', '902.7', '902.8', '902.9', '903.0', ...]  # Include all column names
data = pd.read_csv('addisu_main.csv', names=colnames)
# Data Preprocessing
X = data.iloc[:, 1:-1].values  # Exclude the 'Timestamp' column and target column
y = data.iloc[:, -1].values  # Assuming the last column is the target
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = tf.keras.Sequential([
    LSTM(128, input_shape=(X_train.shape[1], 1), activation='tanh', return_sequences=True),
    Dropout(0.2),
    LSTM(128, activation='tanh', return_sequences=True),
    Dropout(0.2),
    LSTM(128, activation='tanh'),  # Additional LSTM layer
    Dense(128, activation='relu'),  # Additional dense layer
    Dropout(0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])
optimizer = Adam(learning_rate=0.001)
```

```
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
def lr_scheduler(epoch, lr):
    if epoch % 10 == 0 and epoch > 0:
        lr *= 0.85 # Reduce the learning rate by 15% every 10 epochs
    return lr
lr_scheduler_callback = tf.keras.callbacks.LearningRateScheduler(lr_scheduler)  # Use tf.keras.callbacks.
# Train the model with learning rate scheduler
history = model.fit(
    X_train, y_train,
    epochs=400,
    batch_size=128,
    validation_split=0.2,
    verbose=2,
    callbacks=[lr_scheduler_callback]
)
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5).astype(int)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)  # Add F1 Score calculation
mse = mean_squared_error(y_test, y_pred)  # Calculate Mean Squared Error
mae = mean_absolute_error(y_test, y_pred)  # Calculate Mean Absolute Error
conf_matrix = confusion_matrix(y_test, y_pred)
specificity = conf_matrix[0, 0] / (conf_matrix[0, 0] + conf_matrix[0, 1])
TN = conf_matrix[0, 0]
FP = conf_matrix[0, 1]
FN = conf_matrix[1, 0]
TP = conf_matrix[1, 1]
FPR = FP / (FP + TN)
FNR = FN / (FN + TP)
FRN = TP / (TN+ FP)
metrics_data = [
    ["Accuracy", accuracy],
    ["Precision", precision],
```

```
    ["Recall", recall],
    ["F1 Score", f1],
    ["Specificity", specificity],
    ["False Positive Rate (FPR)", FPR],
    ["False Negative Rate (FNR)", FNR],
    ["False Rate Negative  (FRN)", FRN],
    ["Mean Squared Error (MSE)", mse],
    ["Root Mean Squared Error (RMSE)", np.sqrt(mse)],
    ["Mean Absolute Error (MAE)", mae],
    ["Mean Absolute Percentage Error (MAPE)", mae * 100]  # MAPE is usually expressed as a percentage
]
table = tabulate(metrics_data, headers=["Metric", "Value"], tablefmt="grid")
print(table)
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.title('Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.title('Accuracy Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.savefig('training_validation_accuracy.png')
plt.show()
```

Appendix B: Sample Code for Building, Creating, and Compiling the Bi-LSTM Model

```
import numpy as np

import pandas as pd

import tensorflow as tf

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, mean_squared_error,
mean_absolute_error, confusion_matrix

import matplotlib.pyplot as plt

from tensorflow.keras.layers import Bidirectional, LSTM, Dropout, Dense

from tabulate import tabulate

from tensorflow.keras.optimizers import Adam

from sklearn.preprocessing import StandardScaler

colnames = ['Timestamp', '902.6', '902.7', '902.8', '902.9', '903.0', ...]  # Include all column names

data = pd.read_csv('addisu_main.csv', names=colnames)

# Data Preprocessing

X = data.iloc[:, 1:-1].values  # Exclude the 'Timestamp' column and target column

y = data.iloc[:, -1].values  # Assuming the last column is the target

scaler = StandardScaler()

X = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = tf.keras.Sequential([
```

```
 Bidirectional(LSTM(128, activation='tanh', return_sequences=True), input_shape=(X_train.shape[1], 1)),
    Dropout(0.2),
    Bidirectional(LSTM(128, activation='tanh', return_sequences=True)),
    Dropout(0.2),
    Bidirectional(LSTM(128, activation='tanh')),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])
optimizer = Adam(learning_rate=0.001)
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
def lr_scheduler(epoch, lr):
    if epoch % 10 == 0 and epoch > 0:
        lr *= 0.85  # Reduce the learning rate by 15% every 10 epochs
    return lr
lr_scheduler_callback = tf.keras.callbacks.LearningRateScheduler(lr_scheduler)  # Use tf.keras.callbacks.
history = model.fit(
    X_train, y_train,
```

```
    epochs=400,
    batch_size=128,
    validation_split=0.2,
    verbose=2,
    callbacks=[lr_scheduler_callback]
)
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5).astype(int)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)  # Add F1 Score calculation
mse = mean_squared_error(y_test, y_pred)  # Calculate Mean Squared Error
mae = mean_absolute_error(y_test, y_pred)  # Calculate Mean Absolute Error
conf_matrix = confusion_matrix(y_test, y_pred)
specificity = conf_matrix[0, 0] / (conf_matrix[0, 0] + conf_matrix[0, 1])
TN = conf_matrix[0, 0]
FP = conf_matrix[0, 1]
FN = conf_matrix[1, 0]
TP = conf_matrix[1, 1]
FPR = FP / (FP + TN)
FNR = FN / (FN + TP)
```

```
FRN = TP / (TN+ FP)

metrics_data = [

    ["Accuracy", accuracy],

    ["Precision", precision],

    ["Recall", recall],

    ["F1 Score", f1],

    ["Specificity", specificity],

    ["False Positive Rate (FPR)", FPR],

    ["False Negative Rate (FNR)", FNR],

    ["False Rate Negative  (FRN)", FRN],

    ["Mean Squared Error (MSE)", mse],

    ["Root Mean Squared Error (RMSE)", np.sqrt(mse)],

    ["Mean Absolute Error (MAE)", mae],

    ["Mean Absolute Percentage Error (MAPE)", mae * 100

]

table = tabulate(metrics_data, headers=["Metric", "Value"], tablefmt="grid")

print(table)
```

Appendix C: Sample Code for Building, Creating, and Compiling the ConvLSTM Model

```
from tabulate import tabulate

import matplotlib.pyplot as plt

colnames = ['Timestamp', '902.6', '902.7', '902.8', '902.9', '903.0', ...]  # Include all column names

data = pd.read_csv('addisu_main.csv', names=colnames)

X = data.iloc[:, 1:-1].values  # Exclude the 'Timestamp' column and target column

y = data.iloc[:, -1].values  # Assuming the last column is the target

scaler = StandardScaler()

X = scaler.fit_transform(X)

X = X.reshape(X.shape[0], X.shape[1], 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = tf.keras.Sequential([

    Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], 1)),

    LSTM(128, activation='tanh', return_sequences=True),

    Dropout(0.1),

    LSTM(128, activation='tanh', return_sequences=True),

    Dropout(0.1),

    LSTM(128, activation='tanh'),

    Dense(128, activation='relu'),

    Dropout(0.1),

    Dense(64, activation='relu'),
```

```
    Dropout(0.1),
    Dense(1)  # Assuming a single-output regression task
])
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(loss='binary_crossentropy', optimizer=optimizer , metrics=['accuracy'])
# Implement a learning rate scheduler
def lr_scheduler(epoch, lr):
    if epoch % 10 == 0 and epoch > 0:
        lr *= 0.85  # Reduce the learning rate by 15% every 10 epochs
    return lr
lr_scheduler_callback = tf.keras.callbacks.LearningRateScheduler(lr_scheduler)
history = model.fit(
    X_train, y_train,
    epochs=400,
    batch_size=128,
    validation_split=0.2,
    verbose=2,
    callbacks=[lr_scheduler_callback]
)
model.save('occupancy_prediction_model.h5')
y_pred = model.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

```
print(f"Root Mean Squared Error (RMSE) on Test Set: {rmse}")

# Predict occupancy on the test set

y_pred = model.predict(X_test)

y_pred = (y_pred > 0.5).astype(int)

accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)  # Add F1 Score calculation

mse = mean_squared_error(y_test, y_pred)  # Calculate Mean Squared Error

mae = mean_absolute_error(y_test, y_pred)  # Calculate Mean Absolute Error

conf_matrix = confusion_matrix(y_test, y_pred)

specificity = conf_matrix[0, 0] / (conf_matrix[0, 0] + conf_matrix[0, 1])

TN = conf_matrix[0, 0]

FP = conf_matrix[0, 1]

FN = conf_matrix[1, 0]

TP = conf_matrix[1, 1]

FPR = FP / (FP + TN)

FNR = FN / (FN + TP)

FRN = TP / (TN+ FP)

metrics_data = [

    ["Accuracy", accuracy],

    ["Precision", precision],
```

```
    ["Recall", recall],
    ["F1 Score", f1],
    ["Specificity", specificity],
    ["False Positive Rate (FPR)", FPR],
    ["False Negative Rate (FNR)", FNR],
    ["False Rate Negative  (FRN)", FRN],
    ["Mean Squared Error (MSE)", mse],
    ["Root Mean Squared Error (RMSE)", np.sqrt(mse)],
    ["Mean Absolute Error (MAE)", mae],
    ["Mean Absolute Percentage Error (MAPE)", mae * 100]  # MAPE is usually expressed as a percentage
]
table = tabulate(metrics_data, headers=["Metric", "Value"], tablefmt="grid")
```