

SPECTRUM SENSING AND SIGNAL CLASSIFICATION: AN OUT-OF-DISTRIBUTION DETECTION APPROACH

BY YU ZHOU

A thesis submitted to the Graduate Program in Electrical And Computer
Engineering in conformity with the requirements for the Degree of Master of
Applied Science

Queen's University
Kingston, Ontario, Canada
August, 2023

Copyright © Yu ZHOU, 2023

Abstract

This thesis presents an innovative approach for spectrum sensing and signal classification based on out-of-distribution (OOD) detection. Our proposed scheme showcases remarkable resilience in low signal-to-noise ratio (SNR) environments, attaining 100% accuracy in spectrum sensing and 97.7% accuracy in signal classification when a SNR as low as -14dB in our simulation settings. Our method incorporates a voting mechanism for both spectrum sensing and signal classification, removing the need for any classification layer, such as fully connected or softmax layers. A distinguishing feature that sets our research apart from existing studies is its capability to detect a new class of samples that have not been encountered during training. This added capability enhances the robustness, reliability, and security of our system, rendering it highly suitable for real-world applications. Furthermore, our scheme exhibits excellent scalability and flexibility, allowing for the assignment of weights to specific signals for sensing and classification, if necessary. This feature enhances the adaptability of our approach, enabling tailored optimization according to specific requirements.

Acknowledgments

In presenting this research thesis, I would like to extend my sincere gratitude to all those who have contributed to this journey of mine.

First and foremost, I would like to express my deepest appreciation to my academic supervisor, Dr. Il-Min Kim, for his invaluable guidance, unwavering support, and insightful feedback throughout the entire research process. His dedication and expertise have truly been instrumental in shaping the direction and scope of this thesis.

I would also like to extend my heartfelt thanks to the participants of this study, whose willingness to share their experiences and insights has been essential to the successful completion of this research. Without their contributions, this thesis would not have been possible.

Furthermore, I would like to acknowledge the support and encouragement of my family and friends, whose unwavering belief in me has been a constant source of motivation and inspiration. Their love and encouragement have sustained me throughout this challenging journey.

Lastly, I would like to express my sincere gratitude to my colleagues of Wireless Artificial Intelligence lab (WAI lab), whose collective efforts have created an environment that fosters learning, growth, and innovation. It is through their dedication and

commitment that I have been able to pursue this research, and I am honored to be a part of this community.

Once again, thank you to everyone who has contributed to this research thesis. Your support and encouragement have been invaluable, and I am truly grateful for all that you have done.

Statement Of Originality

I hereby declare that the research thesis presented here is my original work, except for the acknowledged citations and references. All references and contributions of other individuals has been cited and sourced appropriately, as defined by the IEEE Citation Reference manual. This thesis has not been submitted, in whole or in part, for any other academic degree or award. I have conducted this research with integrity and in accordance with the ethical principles and guidelines of my academic institution. I take full responsibility for the authenticity and originality of this thesis and acknowledge the contributions of those who have supported me in this endeavor.

Contents

Abstract	i
Statement Of Originality	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
Glossary of Terms	ix
Chapter 1: Introduction	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Thesis Contributions	5
1.5 Thesis Outline	6
Chapter 2: Background	7
2.1 Spectrum Sensing and Signal Classification	7
2.1.1 Classical Methods	9

2.1.2	AI-based Methods	10
2.2	Out-of-distribution Detection	17
2.2.1	Classification-based Methods	19
2.2.2	Density-based Methods	20
2.2.3	Distance-based Methods	21
2.2.4	Reconstruction-based Methods	22
2.3	Data Preprocessing	22
2.3.1	Short-Time Fourier Transform (STFT)	23
2.4	Performance Metrics	25
2.4.1	Confusion matrix [1]	25
2.4.2	Performance Metrics for Spectrum Sensing [2,3]	26
2.4.3	Area Under Receiver Operating Characteristic (AUROC) [4–6]	27
2.5	Summary	29
Chapter 3: Methodology		30
3.1	System Model of Spectrum Sensing	30
3.2	Hypotheses and Confusion Matrix of OOD Detection	33
3.3	System Architecture: Joint Spectrum Sensing and OOD Detection	34
3.4	Data Preprocessing	39
3.5	Element-wise Weighted Mean Square Error (EWMSE)	40
3.6	Extended Multi-threshold ROC and AUROC	43
Chapter 4: Experiments and Results		45
4.1	Implementation Details	45
4.1.1	Simulation Settings	45

4.2	Main Results	51
4.2.1	Performance of Spectrum Sensing	52
4.2.2	Performance of ID Signal Classification (Classification among H2)	53
Chapter 5: Conclusion		58
5.1	Future Work and Conclusions	58
Bibliography		60
Chapter A: Appendix		71
A.1	OOD (unseen signal) detection and ID (seen signals) classification ex- amples of the scenarios of vote-a to h	71
A.2	The effect of the parameter of window length	76
A.3	Introduction of background and foreground separation algorithm	77
A.4	Performance comparison between MSE and EWMSE	78
A.5	Detailed numeric results of spectrum sensing and signal classification	80
A.6	Experiments of different signal representations	80

List of Tables

2.1	Confusion matrix.	25
3.1	The confusion matrix of spectrum sensing.	31
3.2	The confusion matrix of OOD detection.	33
3.3	Possible votes received at the OOD detection arbitrator.	36
A.1	Classification performance of sequential screening of MSE.	81
A.2	Classification performance of sequential screening of EWMSE.	82
A.3	Detailed experiment results of voting mechanism.	83

List of Figures

1.1	Dynamic spectrum access.	4
2.1	STFT frequency - time resolution trade-off illustration.	24
2.2	Receiver operating characteristic (ROC) curve illustration.	28
3.1	System architecture: voting mechanism overview.	35
3.2	An example of the vote triggering the decision that OOD signal is detected.	37
3.3	An example of vote-d: ID signals classification result is S2.	38
3.4	Foreground and background separation of the STFT plot of a given signal.	40
3.5	Reconstruction error distribution of MSE loss.	42
3.6	Reconstruction error distribution of EWMSE loss.	42
3.7	Algorithm of multi-threshold ROC.	44
4.1	RF signals and hypotheses of spectrum sensing.	46
4.2	RF signals and hypotheses of OOD detection.	47
4.3	Dataset split overview.	48
4.4	The typical structure of a convolutional autoencoder.	50
4.5	The detailed structure of the adopted autoencoder.	51

4.6	An example of multi-threshold ROC plot process.	54
4.7	AUROC of OOD detection (detection of H3 between H2 and H3).	55
4.8	Performance of spectrum sensing (sensing H1 between H0 and H1).	55
4.9	Performance of in-distribution signal classification (classification among H2).	56
4.10	Confusion matrixes of four-signal classification.	57
A.1	Scenario of vote-a: triggers the decision of OOD signal is detected.	72
A.2	Scenario of vote-b: ID signals classification result is S2.	72
A.3	Scenario of vote-c: ID signals classification result is S1.	73
A.4	Scenario of vote-d: ID signals classification result is S2.	73
A.5	Scenario of vote-e: ID signals classification result is Noise.	74
A.6	Scenario of vote-f: ID signals classification result is S2.	74
A.7	Scenario of vote-g: ID signals classification result is Noise.	75
A.8	Scenario of vote-h: ID signals classification result is S2.	75
A.9	An example of the how the parameter of window length affects STFT presentation plots.	77
A.10	Algorithm for background and foreground separation.	78
A.11	Sequential screening overview.	79
A.12	Performance comparison of sequential screening between MSE and EWMSE.	80
A.13	Spectrum sensing performance comparison of simple fully connected neural network (sensing H1 between H0 and H1).	84

Glossary of Terms

AI Artificial Intelligence. [2–4](#), [7](#), [10](#), [18](#), [29](#), [58](#)

AMR Automatic Modulation Recognition. [11](#)

AUROC Area Under Receiver Operating Characteristic. [27](#), [28](#), [33](#), [34](#), [43](#), [51](#), [52](#)

AWGN Additive White Gaussian Noise. [31](#), [45](#), [47](#)

BDT Binary Decision Tree. [14](#)

CFO Carrier Frequency Offset. [16](#)

CLDNN Convolutional Long short-term Deep Neural Networks. [11](#)

CM Covariance Matrix. [15](#)

CNN Convolutional Neural Network. [11](#), [12](#), [16](#)

DL Deep Learning. [10](#), [13](#)

DNN Deep Neural Networks. [5](#), [11](#), [12](#), [14](#), [15](#), [21](#)

DSA Dynamic Spectrum Access. [3](#), [8](#)

EWMSE Element-wise Weighted Mean Square Error. 5, 6, 35, 38, 40–42, 49, 50, 77–79

FCC Federal Communications Commission. 8

FFT Fast Fourier Transform. 81

FN False Negatives. 26, 33

FP False Positives. 25, 33

FPR False Positive Rate. 28, 34, 43, 52

FSA Fixed Spectrum Allocation. 7

GMM Gaussian mixture model. 17

HMM Hidden Markov Model. 5

ID In-distribution. 1, 2, 17, 20–22, 27, 31, 33, 34, 36–38, 42, 46–48, 53, 71, 78

IoT Internet of Things. 13

KNN K-Nearest Neighbors. 15, 18

LSTM Long Short-Term Memory. 13, 17

LTE Long-Term Evolution. 12, 17

ML Machine Learning. 10, 14, 17

MLPN Multilayer Linear Perceptron Network. 14

MSE Mean Square Error. 41, 42, 50, 78, 79

OFDM Orthogonal Frequency-Division Multiplexing. 16

OOD Out-of-distribution. i, vi, 1, 2, 5–7, 17–22, 27–31, 33–38, 41–43, 46–48, 50–52, 58, 71, 78

P_{cd} Probability of correct decision. 26, 27

P_d Probability of detection. 26, 27, 52

P_{fa} Probability of false alarm. 26, 27, 52

P_{md} Probability of miss detection. 26, 27

PU Primary Users. 3, 9, 14–16, 27, 30, 32, 48, 52

RF Radio Frequency. 17, 24, 45, 46

ROC Receiver Operating Characteristic. 27, 28, 33, 43, 51, 52

SAE Stacked Autoencoder. 15

SCF Spectral Correlation Function. 11

SISO Single Input and Single Output. 11

SNR Signal-to-noise Ratio. i, 1, 11, 13–16, 24, 46, 51–53, 58, 59, 76, 81, 82

SSD Single Shot Multibox Detector. 12

STFT Short-Time Fourier Transform. 23–25, 34, 39–41, 47, 50, 59, 76, 80–82

SU Secondary Users. [3](#), [9](#), [30](#)

SVM Support Vector Machine. [4](#), [14](#), [15](#)

SWMSE Sample-wise Weighted Mean Square Error. [41](#)

TN True Negatives. [25](#), [33](#)

TP True Positives. [25](#), [34](#)

TPR True Positive Rate. [28](#), [34](#), [43](#), [52](#)

TREE Decision Trees. [14](#), [15](#)

UDSS Unsupervised Deep learning-based Spectrum Sensing. [15](#)

UMTS Universal Mobile Telecommunications Service. [12](#)

VAE Variational Autoencoder. [21](#)

VAE-GMM Variational Autoencoder-Gaussian Mixture Model. [15](#)

VOS Virtual Outlier Synthesis. [19](#)

Wi-Fi Wireless Fidelity. [17](#)

Chapter 1

Introduction

1.1 Introduction

In this thesis, we conduct a thorough investigation of our proposed approach for **OOD** detection-based spectrum sensing and signal classification. Our system demonstrates remarkable resilience in low **SNR** environments. The outcomes of our study provide compelling evidence of the effectiveness of our method, showcasing an impressive achievement of 100% accuracy in spectrum sensing and 97.7% accuracy in signal classification, even when dealing with a **SNR** as low as -14dB, as we simulated in our experiments.

A pivotal aspect to highlight is the composition of our test set, which includes both seen (in-distribution, **ID**) and unseen (out-of-distribution, **OOD**) signals. During the training phase, we exclusively utilize seen signals, while the unseen ones remain untainted by the training process. This deliberate design choice equips our proposed system with the capability to detect the unseen class of signals that have never been encountered during training. This distinguishing feature sets our research apart from

existing studies in the field. It is worth noting that the ability of an artificial intelligence (AI) system to detect unseen samples enhances its robustness, reliability, and security.

To ensure optimal scalability, we train multiple models individually, guaranteeing a robust framework that readily adapts to varying number of signals and requirements. This feature also gives us the flexibility to assign specific weights to signals, enhancing the system's ability for precise sensing and classification, when the need arises.

Undoubtedly, we firmly believe that our findings hold significant potential to advance cognitive radio technology and optimize spectrum utilization. By coping the pressing need for more reliable and sustainable wireless communication systems, we contribute to the ongoing evolution of this critical field.

1.2 Motivation

Our objective is to develop an innovative and efficient generative AI-based system that can detect and classify both seen (ID) and unseen (OOD) signals, without any prior knowledge of the signals being analyzed. This endeavor is motivated by our desire to explore the possibilities of a generative AI-based spectrum sensing and signal classification system, with the goal of advancing cognitive radio technology and optimizing spectrum utilization. Through this research, we aim to contribute the pressing need for wireless communication systems that are more efficient, reliable, and sustainable.

1.3 Problem Statement

Two primary objectives in the design of contemporary wireless communication systems are enhancing spectrum efficiency and incorporating AI capabilities. With the rapid progression of technology and the escalating demand for wireless service bandwidth, the spectrum is becoming increasingly fragmented, posing significant challenges in terms of efficient spectrum allocation and management. To address this challenge, cognitive radio technology presents a promising solution by enabling secondary users (SU) to access the spectrum dynamically (see Figure 1.1, dynamic spectrum access (DSA) [7]), while avoiding any detrimental interference to primary users (PU).

The definition of primary user (PU) and secondary user (SU) is: “Primary user is a transceiver unit that is licensed to use a specific wireless channel in a cognitive communication system. Further, the secondary user refers to a radio unit that adapts its communication settings in order to transmit its data over the primary channel even though the secondary user is not licensed to operate over that channel” [8].

Spectrum sensing holds a pivotal role in cognitive radio technology as it empowers SU to detect the presence or absence of PU within a given frequency band. Precise and timely detection of PU is crucial to prevent any interference with the transmissions of SU. By leveraging the insights derived from spectrum sensing, effective and substantial enhancements to spectrum efficiency can be achieved through adept spectrum management techniques.

Conventionally, various techniques have been employed for spectrum sensing, including energy detection [9], cyclostationary feature analysis [10], matched filtering [11], and compressive sensing [12]. The emergence of AI has brought about a

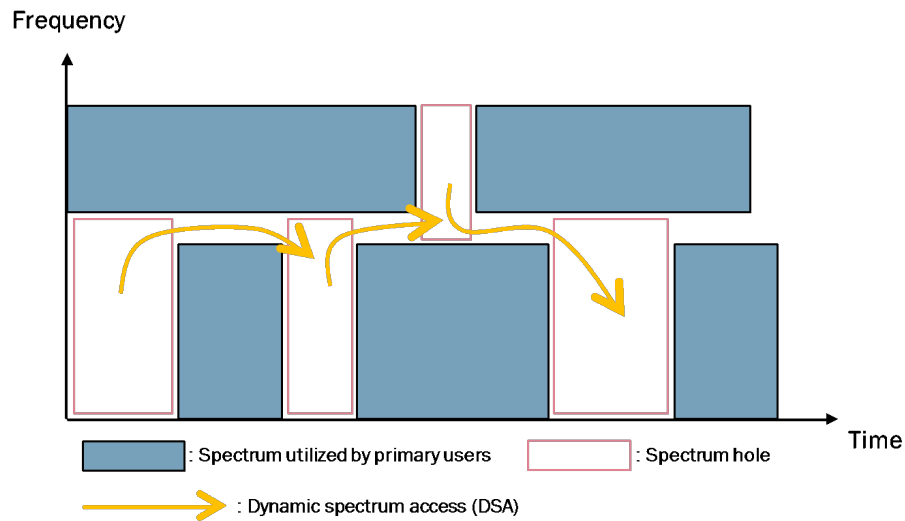


Figure 1.1: Dynamic spectrum access.

transformative impact on wireless communication, enabling the development of more efficient and reliable cognitive radio systems. AI techniques have been applied across multiple domains within wireless communication, including spectrum sensing and signal classification. AI-based spectrum sensing techniques have been devised to enhance the accuracy and effectiveness of spectrum sensing. Furthermore, the simultaneous integration of signal classification with spectrum sensing is crucial, as it provides valuable information for spectrum sensing applications. Several widely utilized AI-based spectrum sensing techniques include:

- Support vector machine (SVM)-based spectrum sensing [13]: This technique uses machine learning algorithm of SVM to classify the received signals as either occupied or unoccupied. SVMs are particularly useful for linearly separable data and can be trained to detect signals in the presence of noise.

- Fuzzy logic-based spectrum sensing [14]: This technique uses fuzzy logic to make decisions about the presence or absence of signals in the spectrum. Fuzzy logic can be used to handle uncertainties and variations in the received signals and is particularly useful when dealing with signals that have a varying power level.
- Hidden Markov model (HMM)-based spectrum sensing [15]: This technique uses a statistical model called a HMM to detect the presence of signals in the spectrum. HMMs can model the temporal behavior of the signals and can be used to detect signals that have a periodic or non-periodic behavior.
- Deep learning-based spectrum sensing [16–26]: This technique uses deep learning algorithms to extract features from the received signals, which are then used to detect the presence of signals in the spectrum. Deep neural networks (DNN) have shown promising results in detecting signals even in noisy environments.

1.4 Thesis Contributions

Our contributions in this research encompass the following:

- We have introduced a customized function called Element-wise Weighted Mean Square Error (EWMSE), which serves a dual purpose as the customized loss function during training and the OOD score calculation function during inference. The incorporation of EWMSE greatly enhances the performance of our proposed system in spectrum sensing and signal classification.

- We have developed a robust, reliable, and secure [OOD](#) detection-based approach for blind detection and classification of transmitting signals, including signals that have never been encountered during the training phase.

1.5 Thesis Outline

The remaining sections of this thesis are organized as follows:

Chapter 2, Background: provides a comprehensive review of the existing literature in the field, specifically focusing on studies related to the problem of spectrum sensing and signal classification, and [OOD](#) detection. The adopted data preprocessing technique and commonly used performance metrics are introduced.

Chapter 3, Methodology: presents the system model, detailed system architecture, the customized function ([EWMSE](#)) and the extended [OOD](#) detection performance metric employed in our research.

Chapter 4, Experiments and Results: outlines the implementation details of our proposed method and presents the corresponding performance evaluation based on our simulation settings.

Chapter 5, Future Work and Conclusions: discusses several potential extensions and improvements that can be explored in future research. Finally, the thesis concludes with a summary of our proposed method and its contributions.

Chapter 2

Background

Efficient utilization of the radio spectrum holds paramount importance in the context of wireless communication systems. In order to achieve this goal, spectrum sensing and signal classification play a pivotal role in supporting the optimization of spectrum utilization. An avenue with great promise for enhancing the reliability and security of AI-based wireless communication systems is the incorporation with OOD detection. OOD detection empowers AI-based wireless communication systems to identify signals that have not been encountered previously, thus strengthening the system's resilience against adversarial attacks and false signals. In this section, we delve into the background of spectrum sensing, signal classification, and OOD detection, exploring their significance and relevance in the field of wireless communication.

2.1 Spectrum Sensing and Signal Classification

Spectrum sensing and signal classification are critical components of modern wireless communication systems. As the demand for wireless communication services continues to rise, the limited availability of radio spectrum has emerged as a significant challenge. In the United States, the fixed spectrum allocation (FSA) policy has

resulted in spectrum utilization ranging from 15% to 85%. Measurement studies conducted by the Federal Communications Commission (FCC) have revealed significant variations in spectrum utilization across different channels [27]. To address this issue and maximize spectrum efficiency, it is imperative to develop techniques that facilitate the efficient management of spectrum resources. Spectrum sensing and signal classification play a vital role in achieving this objective by enabling DSA, which allows users to access the available spectrum in real-time without causing harmful interference to other users [27, 28]. A thorough understanding of the principles and methodologies underlying spectrum sensing and signal classification is crucial for implementing DSA and facilitating efficient spectrum resource management in cognitive radio networks. These functions empower the network to identify and exploit the available spectrum in a dynamic and adaptive manner, thereby laying the foundation for the development of future wireless communication systems. Spectrum sensing plays a crucial role in various applications across the fields of wireless communications, radar systems, and cognitive radio networks, like 5G cellular radio system.

Deep learning excels in spectrum sensing due to its ability to automatically learn complex patterns from raw data, adapt to changing environments, and handle intricate signal characteristics. Unlike conventional methods that rely on hand-engineered features and may struggle with nonlinear or evolving signals, deep learning offers end-to-end learning, robustness to noise, and adaptive feature extraction. This results in improved detection performance, reduced manual intervention, and the potential for continuous improvement through retraining. However, while deep learning has advantages, factors like data availability and computational resources should be considered when choosing between deep learning and conventional methods.

2.1.1 Classical Methods

Spectrum sensing refers to the ability of a wireless system to detect the presence or absence of the **PU** signals in a given frequency band. By accurately sensing the spectrum, a wireless system can identify available frequency bands for **SU** and avoid interference with other wireless systems. The significance of spectrum sensing is particularly pronounced in cognitive radio systems, which aim to dynamically adapt to changing environments and optimize spectrum utilization. Over the years, researchers have developed a diverse array of techniques for spectrum sensing. Some of the classical techniques include:

- **Energy detection:** Energy detection is a spectrum sensing technique that involves measuring the energy of a received signal and comparing it to a predetermined threshold. If the energy of the received signal exceeds the threshold, it is assumed that a signal is present in the frequency band being monitored [9].
- **Cyclostationary feature detection:** Cyclostationary feature detection is a spectrum sensing technique that exploits the statistical properties of signals that are modulated by a periodic waveform. This technique involves analyzing the cyclic autocorrelation function of a received signal to detect the presence of a cyclostationary signal [10].
- **Matched filter detection:** Matched filter detection is a spectrum sensing technique that involves correlating a received signal with a reference signal. This technique is particularly effective for detecting signals that are known a priori, such as those used in radar systems [11].

- Compressive sensing: Compressive sensing is a relatively new spectrum sensing technique that involves compressing a received signal using a random projection matrix and then reconstructing the signal using compressed sensing algorithms. This technique is particularly effective for sparse signals, where only a small fraction of the available frequency spectrum is occupied by signals [12].

Signal classification, on the other hand, involves identifying the signal that has been detected. This information is crucial in cognitive radio networks as it enables more efficient communication by matching the modulation and coding schemes of the sensed signal to the channel conditions. Signal classification is also valuable for interference detection and mitigation. Traditional signal classification techniques include using statistical moments [29], genetic algorithms [30], k-nearest neighbors algorithms [31, 32] and support vector machines [33, 34].

2.1.2 AI-based Methods

Recently, with the revolution of AI in the field of wireless communication, researchers have shifted their attention to AI-based spectrum sensing and signal classification approaches. These research efforts encompass both supervised and unsupervised learning approaches in both deep learning (DL) and classical machine learning (ML) domains.

Supervised Learning Methods

The DL-based spectrum sensing and signal classification method has garnered significant attention from recent researchers. A majority of these recent works adopted

supervised learning methods. The authors in [16] developed DetectNet, a DNN, for performing noncooperative spectrum sensing as a binary hypothesis testing problem. DetectNet is a variant of convolutional long short-term deep neural networks (CLDNN). To further enhance the detection performance, the paper proposes a cooperative spectrum sensing DNN that consists of multiple DetectNets and a SoftCombinationNet. It is worth noting that the energy of the received signals is normalized before being inputted into the DNN, which is considered a significant contribution to improving performance. The simulation results demonstrate that the proposed DNN outperforms conventional noncooperative and cooperative energy detection methods, particularly in low SNR scenarios (SNR below -6dB).

Automatic modulation recognition (AMR) is a common task in signal classification. In [17], Jie Shi, Sheng Hong, et al. propose a DNN-based neural network capable of classifying various modulation schemes in single-input and single-output (SISO) wireless communication systems. The proposed DNN consists of two convolutional layers and three fully connected layers. Experimental results show that the proposed method outperforms traditional feature extraction-based methods, regardless of the presence or absence of phase offset in the modulated symbols.

The approach in [18] utilizes a convolutional neural network (CNN) to perform the tasks of spectrum sensing and signal identification. The spectral correlation function (SCF) of the received signal is calculated as the input feature for the CNN, enabling blind wireless signal identification and classification. To balance the trade-off between performance and computational complexity, two different settings are suggested and studied. Several comparison experiments are conducted to fine-tune the performance of the proposed CNN. The results indicate that the sensing performance is slightly

lower than that of conventional spectrum sensing methods, such as energy detectors and matched filter detectors. However, this research utilizes real-world cellular data (UMTS and LTE signals), which is more promising than studies using simulated or synthetic signals.

In [19], the authors propose a DNN consisting of 7 layers of CNN, capable of classifying non-orthogonal signals. To evaluate the classification ability of the proposed DNN, two types of non-orthogonal signals with different bandwidth compression factors are defined. Type-I signals exhibit large feature diversity, while Type-II signals have strong feature similarity. Comparison studies are performed to investigate the performance of signal feature extraction from the time domain and frequency domain separately. The results demonstrate that training for time domain feature extraction is crucial. Furthermore, an online test of the pre-trained model is conducted, revealing that applying transfer learning to the last two layers (the fully connected layer and softmax layer) significantly improves the performance for both types of non-orthogonal signals.

In [20], two neural networks are proposed: a SSD (Single Shot Multibox Detector) and a multi-input CNN. These networks are designed to perform the tasks of multi-signal detection and modulation recognition, respectively. Subsequently, these neural networks are integrated into the designed system for simulations. Specifically, the proposed SSD network utilizes the time-frequency characteristics to learn the center frequency and start-stop time of signals. However, improving accuracy in start-stop time estimation requires fine design of the feature default box by combining prior information. On the other hand, the multi-input CNN estimates the modulation configurations by learning from the eye diagrams (I-eye and Q-eye) and vector

diagram. The proposed algorithm demonstrates superior performance compared to selected benchmark algorithms, both traditional and DL-based. However, it should be noted that there was limited involvement of real channel data in the experiments. Nevertheless, the proposed algorithm achieves a good balance between computation complexity and accuracy, making it a practical choice for various communication systems.

Low cost and low power consumption are two attractive features of any Internet of Things (IoT) system. In [21], Sreeraj Rajendran, et al. address the challenge of modulation classification in a distributed wireless spectrum sensing network. Firstly, they propose a novel model for automatic modulation classification based on long short-term memory (LSTM) that is data-driven. The model extracts information from the time domain amplitude and phase of the modulation schemes present in the training data, eliminating the need for expert features such as higher-order cyclic moments. The analysis demonstrates that the proposed model achieves an average classification accuracy of nearly 90% across varying SNR conditions ranging from 0dB to 20dB. Additionally, the authors investigate the effectiveness of the proposed model in a scenario involving variable symbol rate. They show that an LSTM-based model can capture meaningful representations of time domain sequences of variable length, which proves advantageous in classifying modulation signals with different symbol rates. To minimize data communication overhead from the distributed sensors, the authors explore the possibility of using averaged magnitude spectrum data for classification, as well as online classification on low-cost spectrum sensors. Furthermore, the paper examines the potential of deploying quantized versions of the proposed models on sensors with limited processing power.

The use of multiple sensors to detect the PU is an effective way to improve detection accuracy [35]. However, in cooperative spectrum sensing, where sensors work together, there is a concern about data privacy since spectrum data is shared among the sensors. To address this issue, a recent study proposed the use of distributed federated learning over a multi-hop wireless network to collectively train a deep neural network for signal identification [22]. In this method, each sensor transmits its trained model to its neighbors, receives DNN models from them, and combines them to create its model for the next round of training. This process is repeated over time, creating a common DNN across the network while preserving the privacy of signals collected at different locations without exchanging any spectrum data. The study results demonstrate the feasibility of expanding cooperative spectrum sensing over a general multi-hop wireless network using federated learning. The results also indicate the ability of federated learning to withstand wireless network effects, enabling high accuracy with low communication overhead and energy consumption.

The application of supervised ML in spectrum sensing and signal classification has a long-standing history of investigation. In [33], Hao Hu, et al. present a method for signal classification that utilizes spectral correlation analysis and SVM. The approach involves selecting four spectral coherence characteristic parameters through spectral correlation analysis and employing a nonlinear SVM for offline calculations, thereby reducing computational complexity. Simulation results demonstrate that the proposed method outperforms existing methods, such as classifiers based on binary decision tree (BDT) and multilayer linear perceptron network (MLPN), particularly in scenarios with low SNR and limited amount of training data. In [36], three ML-based methods for spectrum sensing, namely SVM, decision trees (TREE), and k-nearest

neighbors ([KNN](#)), are examined and compared. The study reveals that [SVM](#) and [KNN](#) methods exhibit superior performance compared to [TREE](#) and certain other traditional detectors, leading to more accurate spectrum sensing operations.

Unsupervised Learning Methods

Unsupervised deep learning-based spectrum sensing ([UDSS](#)) holds great potential, particularly in situations where a large number of labeled sample signals are unavailable or costly. In a recent paper [23], an unsupervised deep learning-based spectrum sensing approach was proposed. The paper introduced an [UDSS](#) algorithm called [VAE-GMM](#) (Variational Autoencoder-Gaussian Mixture Model). To reduce the dimensionality of the raw collected data from a single sensing period, the covariance matrix ([CM](#)) vector of the collected samples was utilized. The unsupervised learning method resolves the binary hypothesis testing [PU](#) detection problem through clustering. Subsequently, a limited number of labeled noise-only samples were utilized to train a neural network in order to identify the physical meaning of each cluster. Simulation results indicate that [VAE-GMM](#) outperforms conventional model-based detection algorithms and approaches the performance of benchmark supervised [DNN](#)-based spectrum detection algorithms.

In [24], the authors proposed two methods based on stacked autoencoder ([SAE](#)) neural networks: [SAE-SS](#) (stacked autoencoder sensing method) and [SAE-TF](#) (stacked autoencoder based spectrum sensing method using time-frequency domain signals). These methods aim to sense the activities of the [PU](#) in systems. [SAE](#) is a fully connected network consisting of encoders and decoders. In low [SNR](#) scenarios, [SAE-TF](#) outperforms [SAE-SS](#), albeit with higher network complexity. Neither [SAE-SS](#) nor

SAE-TF requires prior knowledge of the PU. Simulation results indicate that the proposed neural networks exhibit greater robustness to timing delay, carrier frequency offset (CFO), and noise uncertainty compared to conventional OFDM spectrum sensing methods.

In [25], the authors presented a deep learning-based method for blind detection of underwater acoustic communication (UWAC) signals, which poses a challenging non-cooperative reception scenario. This method addresses two main challenges: highly impulsive noise contamination in UWAC signals and the scarcity of data. To tackle the impulsive noise challenge, the neural network utilizes a two-stage noise reduction design consisting of INP (impulsive noise preprocessor) and SDGAN (signal denoising generative adversarial network). INP suppresses high-amplitude impulsive noise, while the trained SDGAN removes remaining low-amplitude noise. This two-stage denoising significantly improves the SNR. To generate sufficient data from the target channel, a transfer data model inspired by transfer learning is adopted, simplifying the influence of the multipath channel. The transfer data model produces signals with the same components as the transmitted signals but with noise of different distribution and power. The detection problem is then redefined as a binary classification problem, distinguishing between UWAC signals and noise. The denoised signal is fed into a CNN-based classifier for automatic feature extraction and binary classification. Simulation and field test results demonstrate the superior performance of the proposed neural network compared to existing methods, both deep learning-based and non-deep learning-based, particularly in low SNR scenarios.

It should be noted that a certain amount of labeled signal samples are still required for [24] and [25], as the proposed system incorporates classification layers (fully

connected and softmax layer). [23] needs labeled data at the preprocessing stage.

The authors of [26] evaluate various types of autoencoders, such as deep, variational, and LSTM autoencoders, to distinguish and classify between LTE and Wi-Fi transmissions. They analyze the performance differences among these autoencoder types using an I/Q dataset containing LTE and Wi-Fi signals. Their proposed approach demonstrates promising results, with the models capable of being trained in 47 seconds or less. This makes them suitable for online learning and deployment in a dynamic RF environment.

Some of the unsupervised ML-based methods include K-means clustering and Gaussian mixture model (GMM). In [37], a spectrum sensing technique is proposed that utilizes the K-means clustering algorithm and signal features. The K-means algorithm is applied to categorize and analyze the extracted signal features. In [38], the authors employed the GMM to generate a combination of Gaussian probability distributions that accurately represents the energy level vector. Furthermore, this vector is used as input features for a classifier to determine the availability of the channel.

2.2 Out-of-distribution Detection

Out-of-distribution (OOD) detection refers to the identification of samples (the unseen class of signals) that do not conform to the same distribution as the training samples (seen signals). Samples that belong to the same distribution as the training data are known as in-distribution (ID) samples. In the early days of the discipline, OOD detection was not a major concern as models were typically trained on relatively small and homogeneous datasets. However, with the explosion in the availability of

data, the scale and complexity of machine learning applications have grown, and so has the need for models that can generalize to new and unforeseen inputs. This need arises due to the potential threats posed by anomalies and outliers, which can lead to the failure of deployed models.

On the other hand, deep learning has recently emerged as a dominant paradigm in AI, demonstrating the ability to achieve state-of-the-art performance on a wide range of tasks. Nonetheless, this success has come at a cost, as deep neural networks are notoriously susceptible to overfitting and generalization errors. Therefore, detecting instances that deviate from the learned distribution is a fundamental challenge in the field. Addressing this challenge is of paramount importance for preventing catastrophic failures in real-world systems and facilitating more robust and reliable decision-making. OOD detection techniques have wide-ranging applications, including computer vision, natural language processing, and speech recognition, where the ability to detect novel and unexpected inputs is essential for ensuring that models perform well in real-world scenarios. Specifically, some of the applications of OOD detection include anomaly detection, novelty detection, open set recognition, and outlier detection [39].

OOD detection has a rich history that can be traced back to the early days of pattern recognition and statistical modeling. During that time, OOD detection relied on simple rules-based approaches and heuristics. For instance, support vector learning in [40] and KNN in [41] can potentially be adopted for OOD detection. However, with the emergence of deep learning, OOD detection became a more intricate problem due to the increased complexity and non-linearity of deep neural networks. In recent years, as the field has evolved, numerous more sophisticated techniques have been

proposed for OOD detection in deep learning. These techniques include those based on classification, density estimation, distance metrics, and reconstruction metrics [42].

2.2.1 Classification-based Methods

Classification-based OOD detection is a popular research area. The paper [43] is recognized as the starting point of OOD detection in deep learning. In that paper, the authors proposed a baseline OOD detection method using probabilities from softmax distributions: “correctly classified examples tend to have greater maximum softmax probabilities than erroneously classified and out-of-distribution examples” [43].

The article [44] demonstrates the practical training of deep networks based on Bayesian principles. This study enhances uncertainties on OOD data by applying techniques such as batch normalization, data augmentation, and distributed training.

In a separate study, the authors of [45] proposed a novel approach to resampling data in order to obtain a compact yet representative set of background data points. By utilizing the resampled background data, they were able to improve the performance of OOD detection.

In [46], an OOD detector is proposed for the application of action recognition in videos. The authors suggest a separate treatment of seen and unseen action categories, with the latter being synthesized using generative adversarial networks trained on the former.

In [47], Xuefeng Du, et al. propose the Virtual Outlier Synthesis (VOS) method for OOD detection. Specifically, VOS samples virtual outliers from the low-likelihood region of the class-conditional distribution estimated in the feature space. The proposed method achieves competitive performance for both object detection and image

classification models.

Instead of detecting OOD samples from the feature space, [48] proposes an OOD detection approach that relies on the gradient space. The authors introduce the Grad-Norm method, which utilizes information obtained from the gradient space. Grad-Norm utilizes the vector norm of gradients backpropagated from the KL divergence between a uniform probability distribution and the softmax output. The method demonstrates superior performance.

2.2.2 Density-based Methods

More sophisticated techniques for detecting OOD samples, known as density-based methods, utilize explicit modeling of the probability distribution of the ID data. By doing so, they can identify test data residing in regions of low density as OOD samples.

In their proposal [49], the authors designed an unsupervised framework with a deep autoencoder. They used a parametric density estimator to learn the probability distribution underlying the latent representations. The optimization objective was to minimize the differential entropy of the distribution of latent vectors. The proposed approach delivers on-par or superior performances compared to state-of-the-art methods in one-class and video anomaly detection settings.

In [50], the authors studied the problem of density-based generative models mistakenly detecting inputs that significantly differ from training data as ID samples. They connected this problem with the input complexity of generative models' likelihoods. Furthermore, they proposed an efficient and parameter-free OOD score to estimate input complexity, which can be seen as a likelihood ratio.

Interestingly, in [51], the authors challenged the assumption of using the density

of input features for detecting OOD samples. They conducted extensive experiments on public datasets, which showed that the density learned by flow-based models, variational autoencoders (VAE), and PixelCNNs would assign a higher likelihood to OOD samples. Therefore, the authors caution against using density estimates from deep generative models for OOD detection.

2.2.3 Distance-based Methods

Distance-based methods are based on the fundamental concept that OOD samples should exhibit significant distance from the centroids of ID classes.

In [52], the authors propose the use of the minimum Mahalanobis distance to all class centroids as the score for OOD detection. This approach demonstrates greater resilience in the presence of class imbalance and training datasets with noisy labels.

In [53], the authors introduce deterministic uncertainty quantification (DUQ), a radial basis function (RBF) network capable of identifying OOD samples during the test phase of a given deterministic deep model. DUQ comprises a DNN model and a collection of feature vectors corresponding to different classes or centroids. Predictions are generated by computing a kernel function, which serves as a distance metric, between the feature vector produced by the model and the set of centroids.

In [54], Yifei Ming, et al. propose a novel representation learning approach for detecting OOD samples that leverages hyperspherical embeddings of data. This method achieves superior performance by jointly optimizing a dispersion loss and a compactness loss.

2.2.4 Reconstruction-based Methods

Reconstruction-based methods rely on the core idea that reconstruction errors of **ID** samples should be larger than the reconstruction errors of **OOD** samples.

In [55], the authors incorporate reconstruction error and Mahalanobis distance in the latent space as indicators for detecting **OOD** samples. The approach shows a certain level of performance improvement over the baseline approach.

In [56], the authors propose strategies to enhance **OOD** detection performance, including semantic reconstruction, data certainty decomposition, and normalized L2 distance. The objective is to maximize the compression of the latent space while preserving its reconstructive power and semantic meaning. Substantial improvement is observed.

[57] introduces a novel approach for detecting **OOD** samples in image classification tasks. The method involves masking out a random portion of the input image, and then a generative model synthesizes the masked image based on the classification result. The semantic difference between the input image and the synthesized one is used as the score for **OOD** detection. The proposed approach achieves a significantly higher margin of **OOD** detection performance.

2.3 Data Preprocessing

Data preprocessing plays a vital role in the deep learning pipeline as it involves transforming raw data into a suitable format for training neural network models, especially in the context of deep learning. In a study conducted by [58], it was discovered that effective data preprocessing is crucial for achieving high accuracy in deep learning models. The study concluded that techniques such as normalization and data

augmentation can greatly enhance the performance of deep learning models. This finding aligns with a study [59] by LeCun et al., which emphasized the importance of effective data preprocessing, including data augmentation and normalization, for attaining high accuracy in object recognition tasks. The adopted data preprocessing technique of our system will be discussed in this section.

2.3.1 Short-Time Fourier Transform (STFT)

The **STFT** is a signal processing technique that enables the analysis of the frequency content of a non-stationary signal over time. It can be viewed as a modified version of the Fourier Transform, which is a mathematical tool used to represent a signal as a sum of sinusoidal components. The formula for the **STFT** is as follows:

$$X_{WF}(\omega, \tau) = \int_{-\infty}^{\infty} e^{-j\omega t} w(t - \tau) x(t) dt \quad (2.1)$$

The **STFT** can be considered a form of windowed Fourier transform, where $w(t - \tau)$ represents the window function. Consequently, $X_{WF}(\omega, \tau)$ denotes the Fourier transform of $x(t)$ after being windowed by $w(t)$ and shifted to the right by τ . The purpose of the window function is to extract a finite-length portion of the signal $x(t)$, ensuring that the spectral characteristics of the extracted section remain approximately stationary throughout the duration of the window. It is worth noting that if $w(t) = 1$, the **STFT** reduces to the conventional Fourier transform of $x(t)$ [60].

Unlike the Fourier Transform, which analyzes the entire signal at once, the **STFT** divides the signal into smaller segments, referred to as windows, and applies the Fourier Transform to each of these windows individually. This approach enables us to investigate how the frequency content of the signal evolves over time (time-frequency

representation), offering valuable insights into the behavior of the signal.

One of the advantages of **STFT** is its robustness to abrupt changes in the signal [61]. In contrast, Fourier Transform faces challenges in analyzing abrupt changes in signals due to the infinite extent of the basis functions, which causes time-local information to be spread out across the entire frequency axis [60].

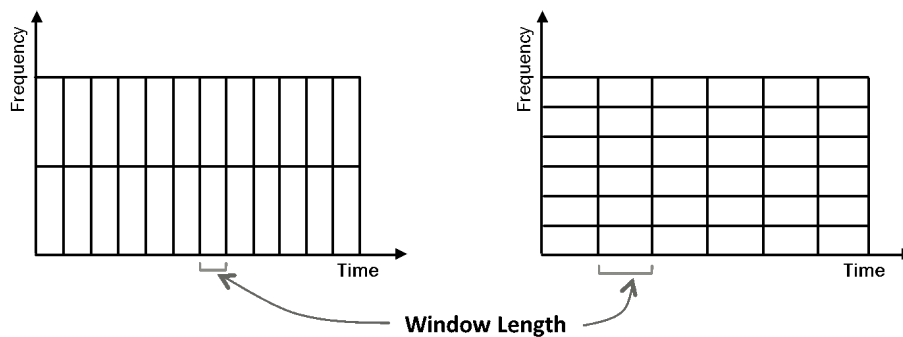


Figure 2.1: STFT frequency - time resolution trade-off illustration.

Additionally, **STFT** enables the analysis of signals in both the time and frequency domains. This is evident in Figure 2.1, which demonstrates how **STFT** can provide a time-frequency representation of **RF** signals. However, there exists a trade-off between frequency resolution and time resolution. This trade-off is controlled by the window length (or segment length) parameter. Figure 2.1 illustrates that a wider window (longer window length) yields better frequency resolution but poorer time resolution, while a narrower window offers improved time resolution but decreased frequency resolution. In our simulation, we carefully considered this frequency-time resolution trade-off and appropriately selected the window length parameter for each **SNR** point. For a more detailed examination of the impact of the window length parameter, please refer to Appendix A.2.

Confusion matrix		
Actual \ Predicted	Negative (0)	Positive (1)
Negative (0)	TN	FP
Positive (1)	FN	TP

Table 2.1: Confusion matrix.

By converting signals into images using the Short-Time Fourier Transform (STFT), we obtain time-frequency representations of the signals. The neural networks in our proposed systems can learn patterns and generate images that represent various aspects of the signal's content. When the STFT matrix is plotted as an image, the amplitude information of a signal is transformed into color information within the image. However, the phase information is not captured within the image.

2.4 Performance Metrics

2.4.1 Confusion matrix [1]

A confusion matrix is a table that is used to evaluate the performance of a classification algorithm. Each row of the confusion matrix represents instances in the actual class, while each column represents instances in the predicted class. True Positives (TP) are cases where the actual class of the data point is 1 (Positive) and the predicted class is also 1 (Positive), and hence, the prediction is true, i.e., the predicted class is the same as the actual class. True Negatives (TN) are cases where the actual class of the data point is 0 (Negative) and the predicted class is also 0 (Negative). False Positives (FP) are cases where the actual class of the data point is 0 (Negative) and the predicted class is 1 (Positive). 'False' indicates that the model has predicted

incorrectly, and ‘positive’ indicates that the predicted class was positive. False Negatives (FN) are cases where the actual class of the data point is 1 (Positive) and the predicted class is 0 (Negative). ‘False’ indicates that the model has predicted incorrectly, and ‘negative’ indicates that the predicted class was negative.

Based on the confusion matrix in Table 2.1, the following metrics can be defined:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}} \quad (2.2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \quad (2.3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \quad (2.4)$$

2.4.2 Performance Metrics for Spectrum Sensing [2, 3]

The performance of the sensing algorithm is measured in terms of the probability of detection (Pd), probability of correct decision (Pcd), probability of missed detection (Pmd), and probability of false alarm (Pfa). The corresponding performance metrics are defined as follows:

$$\text{Pd} = \Pr(\text{H1} | \text{H1}) \quad (2.5)$$

$$\text{Pcd} = \Pr(\text{H1}) \Pr(\text{H1} | \text{H1}) + \Pr(\text{H0}) \Pr(\text{H0} | \text{H0}) \quad (2.6)$$

$$\text{Pmd} = \Pr(\text{H0} | \text{H1}) \quad (2.7)$$

$$\text{Pfa} = \Pr(\text{H1} | \text{H0}) \quad (2.8)$$

Furthermore:

$$\text{Pd} + \text{Pmd} = 1 \quad (2.9)$$

$$P_{fa} + \Pr(H_0 | H_0) = 1 \quad (2.10)$$

where H_1 and H_0 are the binary hypotheses of spectrum sensing, see Section 3.1.

Here, P_d represents the probability of detecting the presence of PU signal given that it is actually present. P_{cd} represents the probability that the sensing result matches the actual presence or absence of the PU signal. P_{md} represents the probability of detecting the absence of the PU signal given that it is actually present. P_{fa} represents the probability of detecting the presence of PU signal given that it is actually absent. The sum of P_d and P_{md} is 1 ($P_d + P_{md} = 1$), and the sum of P_{fa} and $\Pr(H_0 | H_0)$ is 1 ($P_{fa} + \Pr(H_0 | H_0) = 1$). It is important to note that a good spectrum sensing algorithm aims to achieve high P_d and low P_{fa} simultaneously.

2.4.3 Area Under Receiver Operating Characteristic (AUROC) [4–6]

AUROC stands for Area Under the Receiver Operating Characteristic. It is a measure of the performance of a binary classification model, specifically its ability to discriminate between positive examples and negative examples. The **AUROC** is computed by calculating the area under the **ROC** (Receiver Operating Characteristic) curve, and it provides a scalar value between 0 and 1.

AUROC serves as a performance metric for evaluating the model's discrimination capability. The higher the **AUROC**, the better the performance of a binary classification model. For instance, an **AUROC** of 0.8 implies that the model exhibits good discriminatory ability. In 80% of cases, the model will accurately assign a higher score to an **OOD** signal compared to **ID** signals, in our problem setting. A value of 1 indicates a perfect **OOD** detector, while a value of 0.5 suggests the **OOD** detector performs no better than random guessing.

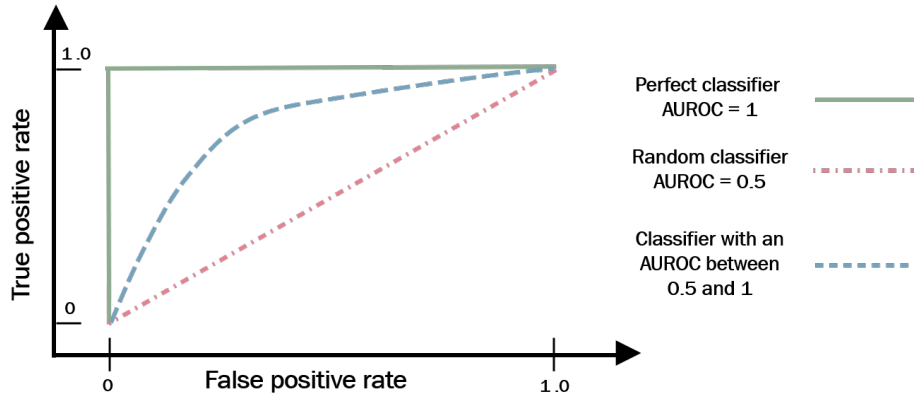


Figure 2.2: Receiver operating characteristic (ROC) curve illustration.

On the other hand, the ROC curve is a visual representation of a binary classification model's performance across various classification thresholds. It illustrates the relationship between the true positive rate (TPR, see equation (2.11)) and the false positive rate (FPR, see equation (2.12)) as the threshold for classification is adjusted [4, 5]. Figure 2.2 provides an example of ROC curves with different AUROC values, which plots the TPR against the corresponding FPR. By examining different points on the curve, one can assess the trade-off between the TPR and the FPR for different classification thresholds. Note that the classification thresholds are implicitly shown in the ROC curves.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.11)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (2.12)$$

Note that AUROC is adopted as the metric for measuring the performance of OOD detection.

2.5 Summary

Numerous spectrum sensing and signal classification applications exist in the field of wireless communication, encompassing cognitive radio networks, spectrum monitoring, and interference detection [62]. However, the implementation of these techniques presents challenges due to the intricate nature of wireless signals, which exhibit variations in frequency, modulation, and power.

The efficient utilization of spectrum through cognitive radio networks holds the potential to revolutionize wireless communication by optimizing spectrum allocation, mitigating interference, and enhancing network performance. The success of cognitive radio networks critically depends on effective spectrum sensing and signal classification techniques. Consequently, ongoing research in this area aims to continuously improve the performance of these techniques by developing more robust and accurate algorithms that leverage the advantages of AI.

OOD techniques inherently contribute to bolstering the security and reliability of numerous real-world AI applications. This thesis explores a deep learning-based OOD detection for spectrum sensing and signal classification.

Chapter 3

Methodology

In this section, we will explain in detail the methodology of our proposed out-of-distribution (OOD) detection-based spectrum sensing and signal classification approach.

3.1 System Model of Spectrum Sensing

For spectrum sensing, the system is modeled as the following binary hypothesis testing problem:

$$y(n) = \begin{cases} w(n) : & H_0 \\ h * S_i(n) + w(n) : & H_1 \end{cases} \quad (3.0.1)$$

where $y(n)$ is the n -th sensed signal sample, $w(n)$ is the additive noise of the n -th sensing period, h is channel gain, and $S_i(n)$ is the signal of the i^{th} primary user (PU) of the n -th sensing period. H_1 and H_0 stand for the hypotheses of the presence of the primary signal and noise only (absence of the primary signal), respectively. Note that the definition of PU and secondary user (SU) is: “Primary user is a transceiver unit

Confusion matrix of spectrum sensing					
Actual \ Predicted		H0	H1		
		Noise (H0)	S1 (H1-1)	S2 (H1-2)	S3 (H1-3)
H0	Noise (H0)	TN	FP ₀₁	FP ₀₂	FP ₀₃
H1	S1 (H1-1)	FN ₁₀	TP ₁₁	TP ₁₂	TP ₁₃
	S2 (H1-2)	FN ₂₀	TP ₂₁	TP ₂₂	TP ₂₃
	S3 (H1-3)	FN ₃₀	TP ₃₁	TP ₃₂	TP ₃₃

Table 3.1: The confusion matrix of spectrum sensing.

that is licensed to use a specific wireless channel in a cognitive communication system. Further, the secondary user, refers to a radio unit that adapts its communication settings in order to transmit its data over the primary channel even though the secondary user is not licensed to operate over that channel [8]”. It is worth noting that the waveform of $S_i(n)$ is unknown.

In this thesis, we consider the system setting assuming three information-bearing signals and one noise signal: S1, S2, S3, and Noise. Specifically, S1 is a non-linear frequency modulation pulse, Equation 3.0.2; S2 is a linear frequency modulation pulse, Equation 3.0.3; S3 is a combined signal of S1 and S2, i.e., $S3=S1+S2$; and Noise is an **AWGN**. Throughout this thesis, we will assume that S1, S2, and Noise are considered as seen signals (in-distribution, **ID**) and will participate in neural network training. On the other hand, S3 is an unseen class of signals (out-of-distribution, **OOD**) that will not be involved in training the models.

$$s(t) = \exp \left(j2\pi \int_0^t (f_0 + \beta\tau^2) d\tau \right) \quad (3.0.2)$$

where $s(t)$ represents the non-linear frequency modulation pulse signal, f_0 is the starting frequency of the pulse, β is a parameter controlling the non-linear frequency

change rate, and t is the time variable.

$$s(t) = \exp\left(j2\pi \int_0^t (f_0 + \beta\tau) d\tau\right) \quad (3.0.3)$$

where $s(t)$ represents the linear frequency modulation pulse signal, f_0 is the starting frequency of the pulse, β is the frequency rate of change, and t is the time variable.

For spectrum sensing in our problem settings, there are three types of H1 hypotheses (S1, S2, and S3) representing different PU signals and one type of H0 hypothesis (Noise) representing noise. The binary hypothesis of spectrum sensing and corresponding confusion matrix are shown in Table 3.1. The performance metrics utilized for spectrum sensing are introduced in Section 2.4.2. Specifically, the adapted performance metrics to our problem settings of spectrum sensing are,

$$Pd = \frac{\sum_{i=1}^3 \sum_{j=1}^3 TP_{ij}}{\sum_{i=1}^3 \sum_{j=1}^3 TP_{ij} + \sum_{i=1}^3 FN_{i0}} \quad (3.0.4)$$

$$Pcd = Pd + \frac{TN}{TN + \sum_{j=1}^3 FP_{0j}} \quad (3.0.5)$$

$$Pmd = \frac{\sum_{i=1}^3 FN_{i0}}{\sum_{i=1}^3 \sum_{j=1}^3 TP_{ij} + \sum_{i=1}^3 FN_{i0}} \quad (3.0.6)$$

$$Pfa = \frac{\sum_{j=1}^3 FP_{0j}}{TN + \sum_{j=1}^3 FP_{0j}} \quad (3.0.7)$$

Note that Equations 3.0.4, 3.0.5, 3.0.6 and 3.0.7 are derived from the notations presented in Table 3.1.

Confusion matrix of OOD detection					
Actual \ Predicted		H2			H3
		Noise (H2-0)	S1(H2-1)	S2 (H2-2)	S3 (H3)
H2	Noise (H2-0)	TN ₀₀	TN ₀₁	TN ₀₂	FP ₀₃
	S1 (H2-1)	TN ₁₀	TN ₁₁	TN ₁₂	FP ₁₃
	S2 (H2-2)	TN ₂₀	TN ₂₁	TN ₂₂	FP ₂₃
H3	S3 (H3)	FN ₃₀	FN ₃₁	FN ₃₂	TP

Table 3.2: The confusion matrix of OOD detection.

3.2 Hypotheses and Confusion Matrix of OOD Detection

OOD detection can also be formulated as a binary hypothesis (ID and OOD) testing problem, distinct from that of spectrum sensing as discussed in Section 3.1. The binary hypothesis for OOD detection and its corresponding confusion matrix are presented in Table 3.2. We consider three types of H2 hypotheses (Noise, S1, and S2; ID) and one type of H3 hypothesis (S3; OOD). The performance metric employed for OOD detection is AUROC, calculated using the extended multi-threshold ROC method introduced in Section 3.6. Specifically, OOD samples are treated as positive examples, while ID samples are considered negative examples. The adapted definitions of TN, FP and FN to our problem settings in OOD detection are,

$$\text{TN} = \sum_{i=0}^2 \sum_{j=0}^2 \text{TN}_{ij} \quad (3.0.8)$$

$$\text{FP} = \sum_{i=0}^2 \text{FP}_{i3} \quad (3.0.9)$$

$$\text{FN} = \sum_{j=0}^2 \text{FN}_{3j} \quad (3.0.10)$$

Note that Equations 3.0.8, 3.0.9 and 3.0.10 are derived from the notations presented in Table 3.2. Utilizing these equations and the TP defined in Table 3.2, TPR and FPR can be calculated using Equations 2.11 and 2.12, respectively, in order to compute the AUROC for OOD detection.

3.3 System Architecture: Joint Spectrum Sensing and OOD Detection

Recall that, in this thesis, spectrum sensing refers to sensing the hypothesis of H1 between hypotheses H0 and H1. Whereas, OOD detection refers to detecting the hypothesis of H3 between hypotheses H2 and H3. ID signals classification refers to the classification among signals: S1, S2 and Noise which are considered as the hypothesis of H2.

To simultaneously detect the presence of primary users and classify each signal in the simulation setting, we addressed the problem in two steps:

1. **OOD detection: this step involves detecting the unseen class of signals (H3: S3).**
2. **ID signals classification: here, we classify those seen signals (H2: Noise, S1 and S2).**

The corresponding system architecture is depicted in Figure 3.1, showcasing our proposed approach named the ‘voting mechanism’. This mechanism efficiently tackles the tasks of OOD detection and ID signals classification in a parallel screening style. The input to our system consists of images (STFT plots). As detailed in Section 2.3.1, first, the raw sensed I/Q data will be preprocessed by STFT then the output matrix of STFT will be plotted into images.

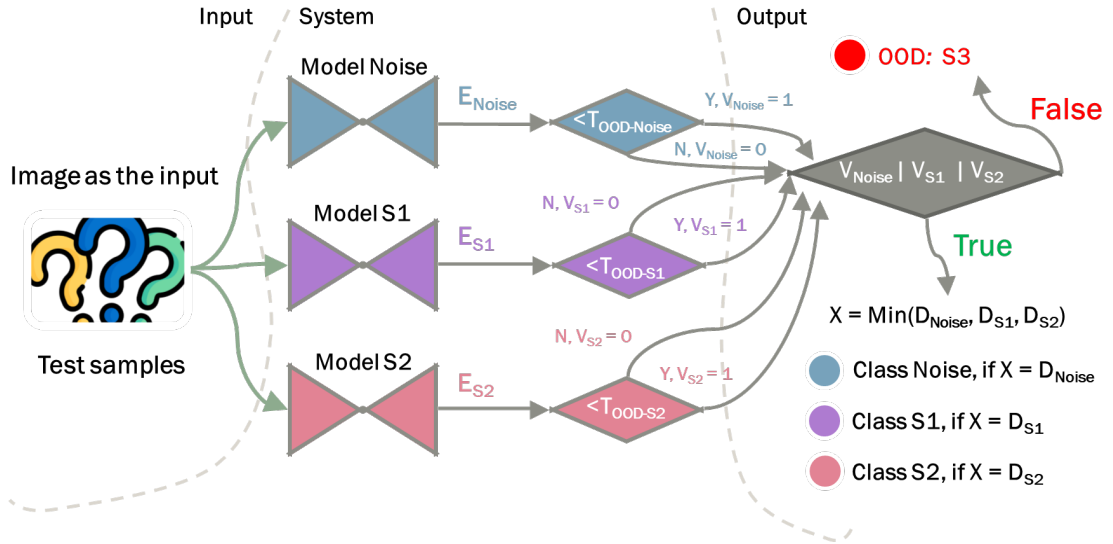


Figure 3.1: System architecture: voting mechanism overview.

During training, we employ a separate model for each seen signal (Noise, S1, or S2). As a result, after thorough training, we possess three individually trained models: model Noise, model S1, and model S2. Additionally, a carefully tuned threshold is assigned to each corresponding trained model: $T_{\text{OOD-Noise}}$, $T_{\text{OOD-S1}}$, and $T_{\text{OOD-S2}}$.

During inference, a test sample passes through these three trained models simultaneously. Each model calculates the reconstruction error of the test sample using the customized OOD score function called Element-wise Weighted Mean Square Error Loss (EWMSE), whose details are available in Section 3.5. By comparing the reconstruction error (OOD score) of the test sample with the assigned threshold, each model generates its decision or vote regarding whether the test sample belongs to the signal class it was trained with. Specifically, if the reconstruction error of the test sample for a particular model exceeds the model's threshold, the model votes logic bit 0; otherwise, it votes logic bit 1. The votes (V_{Noise} , V_{S1} , and V_{S2}) from all models

Autoencoder (Model)	Possible received votes							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

Table 3.3: Possible votes received at the OOD detection arbitrator.

are collected at the OOD detection arbitrator (the grey diamond box in Figure 3.1), where the decision regarding whether the test sample is an OOD sample (the unseen class of signals) is made.

The OOD detection arbitrator performs a logical ‘OR’ operation on the received votes (consisting of three logic bits: V_{Noise} , V_{S1} , and V_{S2}) from the models. If the logical ‘OR’ operation results in a False logic, it signifies that the test sample is an OOD signal. Consequently, the test sample is screened out and further processing is halted. On the other hand, if the logical ‘OR’ operation yields a True logic, the test sample proceeds to the stage of ID signals classification in the subsequent step. This indicates that the test sample belongs to one of the ID signals (seen signals).

From the perspective of each individual model, the trained model essentially makes a binary decision regarding whether the test sample belongs to the signal class it was trained with. Table 3.3 presents all eight possible votes (vote-a to h) from these three trained models. Only the vote-a will result in a decision of the detection of an OOD signal (the unseen class of signals).

Figure 3.2 illustrates an example of the reconstruction errors (OOD scores) for vote-a, which triggers the detection of an OOD signal. Vote-a represents the scenario where the reconstruction errors (E_S , where $S \in [\text{Noise}, \text{S1}, \text{S2}]$) of all the models

exceed their respective thresholds ($T_{\text{OOD-S}}$, where $S \in [\text{Noise}, S1, S2]$).

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

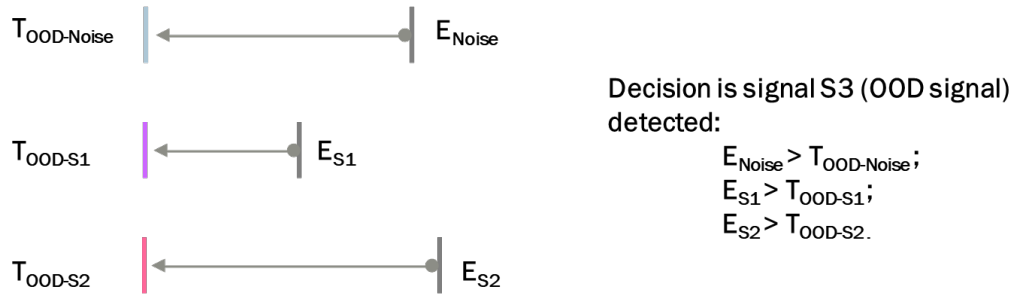


Figure 3.2: An example of the vote triggering the decision that OOD signal is detected.

Up to this point, the task of OOD detection has been completed, and ideally, no OOD samples will proceed further as they are screened out upon OOD detection performed at the OOD detection arbitrator. The next step involves classifying ID signals, determining which specific ID signal (Noise, S1, or S2) a test sample belongs to. This classification is done using a hard rule that compares the differences between the OOD score (reconstruction error) of a test sample and the corresponding threshold of each model. The bottom right part of Figure 3.1 illustrates how ID signals classification is performed. This classification process also considers those introduced three independent thresholds ($T_{\text{OOD-Noise}}$, $T_{\text{OOD-S1}}$, and $T_{\text{OOD-S2}}$).

First, we calculate D_S , where $S \in [\text{Noise}, S1, S2]$. D_S is defined as follows:

$$D_S = E_S - T_{\text{OOD-S}} \tag{3.0.11}$$

where $S \in [\text{Noise}, S1, S2]$. E_S represents the OOD score (EWMSE is the OOD score function) of a test sample for model S , while $T_{\text{OOD-S}}$ denotes the assigned threshold of model S . Please note that the values of D_S can be either negative or positive.

Next, the decision for ID signals classification is made by identifying the signal class ($S \in [\text{Noise}, S1, S2]$) of the trained model with the smallest D_S as the signal class of the test sample. In other words, the signal class of a test sample is determined by finding the minimum value (D_x) among D_{Noise} , D_{S1} , and D_{S2} , where D_x is given by:

$$D_x = \text{Min} (D_{\text{Noise}} , D_{S1}, D_{S2}) \tag{3.0.12}$$

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

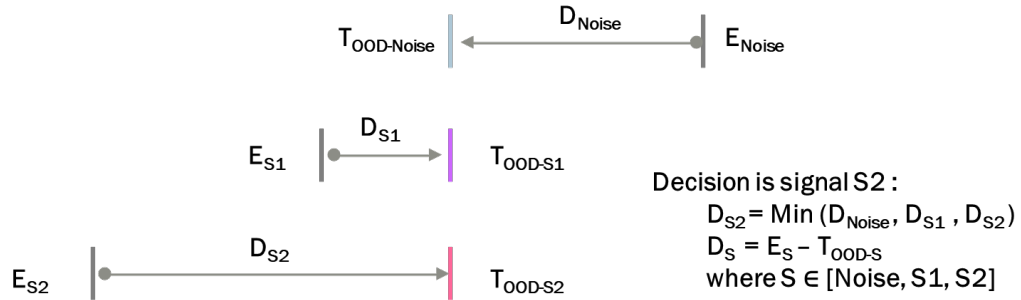


Figure 3.3: An example of vote-d: ID signals classification result is S2.

Figure 3.3 illustrates an example of the ID signals classification process when the vote-d is generated. In this particular scenario, the signal type of the test sample is classified as S2, as the smallest D_S value corresponds to D_{S2} , where $S \in [\text{Noise}, S1, S2]$. It is worth mentioning that additional examples of ID signals classification for

the scenarios of vote-b to h can be found in Appendix [A.1](#).

Analyzing the inference time computational complexity of a model is essential to understand its efficiency and feasibility for real-time applications. From the system architecture shown in Figure [3.1](#), we can deduce that the majority of time consumed by inference samples comes from the neural networks (detailed parameters of each autoencoder are available in Figure [4.5](#)) of the proposed system. The time consumed by operations such as logic OR and MIN is negligible. On your experimental machine (with an NVIDIA RTX A6000 GPU card and 48 GB of Graphics RAM), the inference time for one test sample of the proposed system averages about 40 milliseconds. Given that the system sensing time is 60ms, the spectrum holes should be longer than 100ms in the time domain.

3.4 Data Preprocessing

Through several toy experiments, introduced in Appendix [A.6](#), involving time domain processing and frequency domain processing, we have determined that the short-time Fourier transform ([STFT](#)) is the most suitable method for preprocessing our raw data (sensed I/Q data). Then the [STFT](#) matrix of a sensed signal will be plotted into an image which will be the input data format of the neural networks of our proposed system. Details on [STFT](#) are in Section [2.3.1](#).

In our proposed system, instead of directly inputting the raw [STFT](#) matrices of the sensed signals into the neural networks, we first plot the [STFT](#) matrices of the sensed signals as images and then feed these images into our proposed model. This process can be considered as a dimensionality reduction operation since the number of possible values of the elements in the raw matrices is larger than the number of

possible values of the pixels in an image (which typically ranges from 0 for black to 255 for white). Our experimental results indicate that this dimension reduction improves the performance.

3.5 Element-wise Weighted Mean Square Error (EWMSE)

The idea of introducing [EWMSE](#) is intuitively inspired by the following thought: when presented with an image of a signal, humans naturally focus on the signal area (the foreground) to extract meaningful information for signal recognition. Similarly, in our problem, we aim to direct the neural network's attention towards the foreground information. To achieve this, we assign a relatively larger weight to the pixels in the foreground during both training and inference processes by introducing customized function of [EWMSE](#). Figure 3.4 provides an illustration of the foreground and background separation of the [STFT](#) plot of a given signal. More details of foreground background separation algorithm are available in Appendix [A.3](#).



Figure 3.4: Foreground and background separation of the STFT plot of a given signal.

The formula of [EWMSE](#) function is as follows:

$$\frac{1}{NP} \sum_{j=1}^N \sum_{i=1}^P W_{ij} (y_{ij} - \hat{y}_{ij})^2 \quad (3.0.13)$$

where N represents the number of input vectors (Y_1, \dots, Y_n), and P represents the number of elements ($1, \dots, p$) of each input vector Y_i . The corresponding N reconstructed vectors are $\hat{Y}_1, \dots, \hat{Y}_n$. y_{ij} refers to the j^{th} element of the i^{th} input vector, and \hat{y}_{ij} refers to the j^{th} element of the i^{th} reconstructed vector. W_{ij} represents the weight of j^{th} element of the i^{th} input vector. The ranges of i and j are $i \in [1, n]$ and $j \in [1, p]$, respectively.

If all the P elements of an input vector Y_i has a same weight (W_i), the [EWMSE](#) function reduces to Sample-wise Weighted Mean Square Error ([SWMSE](#)) function. The [SWMSE](#) function can be defined as follows:

$$\frac{1}{N} \sum_{i=1}^N W_i \left\| Y_i - \hat{Y}_i \right\|^2 \quad (3.0.14)$$

In this case, all the elements of an input vector share the same weight W_i .

As we can see from equation [3.0.13](#), the [EWMSE](#) is a customized version of the vanilla mean square error ([MSE](#)). We introduced [EWMSE](#) as a customized loss function that allows the neural networks in our system to focus more attention (assign more weight) to the foreground area of the input, which in this case is the [STFT](#) plots. In our proposed system, [EWMSE](#) is utilized both during training and inference. During training, [EWMSE](#) serves as a customized loss function, while during inference, it acts as the [OOD](#) score function, serving as an indicator upon which the [OOD](#) detection decision is made.

The distribution of reconstruction errors differs when applying the [MSE](#) loss compared to the [EWMSE](#) loss. Figure [3.5](#) illustrates that when using the regular [MSE](#) loss, each pixel's reconstruction error is given equal weight, regardless of whether it

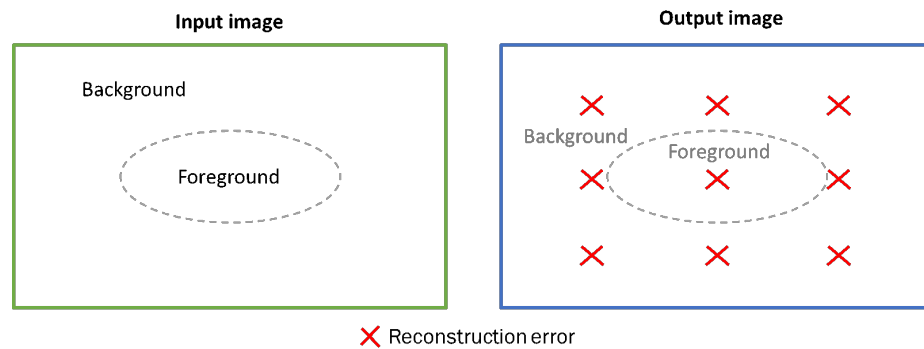


Figure 3.5: Reconstruction error distribution of MSE loss.

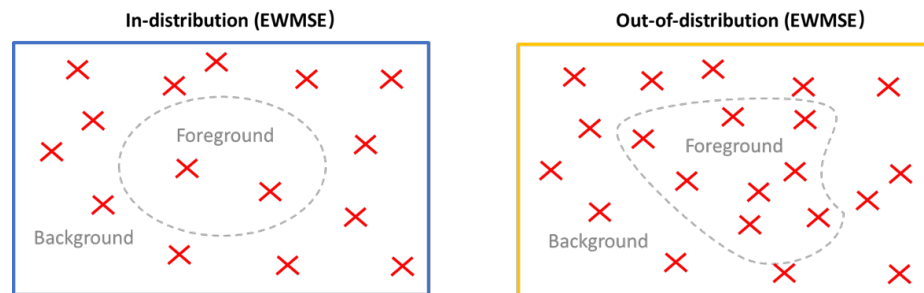


Figure 3.6: Reconstruction error distribution of EWMSE loss.

belongs to the foreground or background of the image. However, as shown in Figure 3.6, when EWMSE loss is utilized, the distribution of reconstruction errors varies for the foreground and background regions of an image. In the foreground, the reconstruction errors of the seen signals (ID samples) are smaller compared to those of the unseen class of signals (OOD samples). Conversely, in the background, the reconstruction errors of the ID signals are relatively similar to those of the OOD samples. This disparity arises because EWMSE loss instructs the neural network to prioritize the foreground area during training by assigning a larger weight to the foreground pixels. The performance comparison between MSE and EWMSE shows that EWMSE has superior performance compared to MSE. More details are available

in Appendix [A.4](#).

3.6 Extended Multi-threshold ROC and AUROC

Conventionally, the [ROC](#) curve represents the performance of a model using a single threshold. However, our proposed system has three individual thresholds to consider. To evaluate the discriminability of the [OOD](#) detector of our proposed system, we have extended the concept of [ROC](#) curve to accommodate multiple thresholds models. The algorithm for generating the multi-threshold [ROC](#) is provided in Figure [3.7](#). Firstly, collect the reconstruction errors (E_{Noise} , E_{S1} , E_{S2}) of the test samples from each individual neural network. Next, identify the value range of the thresholds for each individual model. Subsequently, enumerate all possible combinations of the thresholds within the range: $(T_{\text{OOD-Noise},i}, T_{\text{OOD-S1},j}, T_{\text{OOD-S2},k})$. Afterwards, calculate all combinations of true positive rates ([TPRs](#)) and false positive rates ([FPRs](#)) based on the enumerated threshold combinations. Then, eliminate the sub-optimal points from the raw [ROC](#). Finally, calculate the [AUROC](#) and draw the [ROC](#) curve accordingly. An example of the multi-threshold [ROC](#) curve is depicted in Figure [4.6](#) of Section [4.2](#).

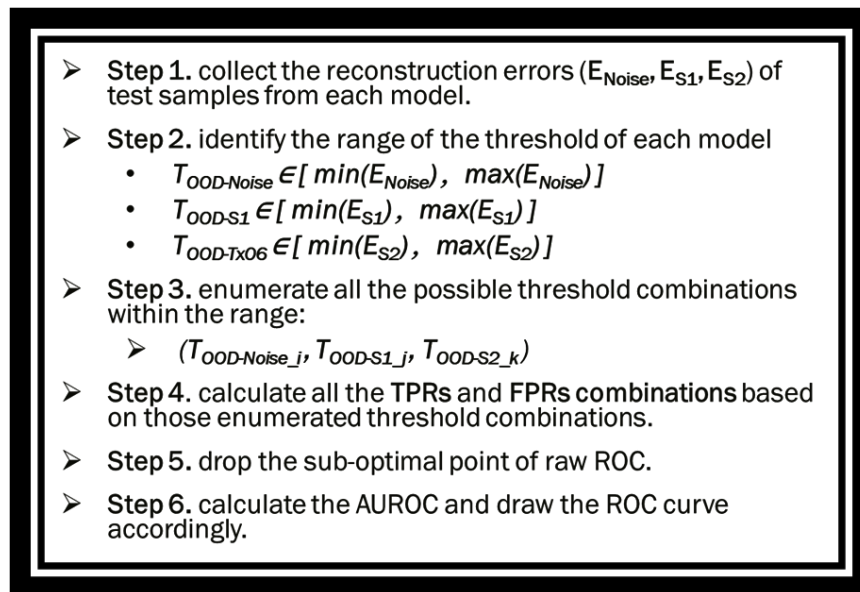
- 
- **Step 1.** collect the reconstruction errors ($E_{\text{Noise}}, E_{S1}, E_{S2}$) of test samples from each model.
 - **Step 2.** identify the range of the threshold of each model
 - $T_{\text{OOD-Noise}} \in [\min(E_{\text{Noise}}), \max(E_{\text{Noise}})]$
 - $T_{\text{OOD-S1}} \in [\min(E_{S1}), \max(E_{S1})]$
 - $T_{\text{OOD-TX06}} \in [\min(E_{S2}), \max(E_{S2})]$
 - **Step 3.** enumerate all the possible threshold combinations within the range:
 - $(T_{\text{OOD-Noise}_j}, T_{\text{OOD-S1}_j}, T_{\text{OOD-S2}_k})$
 - **Step 4.** calculate all the TPRs and FPRs combinations based on those enumerated threshold combinations.
 - **Step 5.** drop the sub-optimal point of raw ROC.
 - **Step 6.** calculate the AUROC and draw the ROC curve accordingly.

Figure 3.7: Algorithm of multi-threshold ROC.

Chapter 4

Experiments and Results

4.1 Implementation Details

4.1.1 Simulation Settings

As shown in Figure 4.1 and Figure 4.2, our simulation settings include four signals: S1, which is a non-linear frequency modulation pulse; S2, which is a linear frequency modulation pulse; S3, a combined signal of S1 and S2; and Noise, an [AWGN](#), noise signal.

Specifically, the properties of S1 and S2 are as follows:

For S1:

- The pulse repetition interval is 10 ms.
- Each pulse has a duration of 1 ms.
- Each pulse exhibits non-linear frequency modulation spanning 8 MHz over 1 ms.
- The [RF](#) of each pulse is 9360 MHz.

For S2:

- The pulse repetition interval is 5 ms.
- Each pulse has a duration of 1 ms.
- Each pulse has linear frequency modulation spanning 8 MHz over 1 ms.
- The RF of each pulse is 9356 MHz.

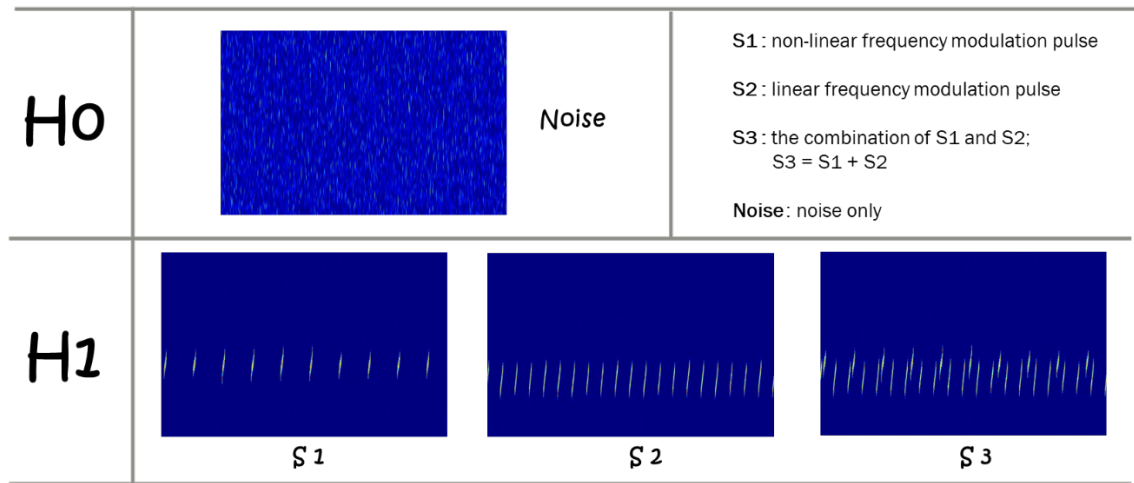


Figure 4.1: RF signals and hypotheses of spectrum sensing.

For each SNR scenario, there are three types of H1 hypotheses (S1, S2, and S3) and one type of H0 hypothesis (Noise), see Figure 4.1. Furthermore, S1, S2, and Noise are considered as seen signals (in-distribution, ID) and will participate in neural network training. On the other hand, S3 is an unseen class of signals (out-of-distribution, OOD) that will not be involved in training the models. As shown in Figure 4.2, we consider three types of H2 hypotheses (Noise, S1, and S2) and one type of H3 hypothesis (S3).

In this thesis, spectrum sensing refers to sensing the hypothesis of H1 between hypotheses H0 and H1, whereas OOD detection refers to detecting the hypothesis of

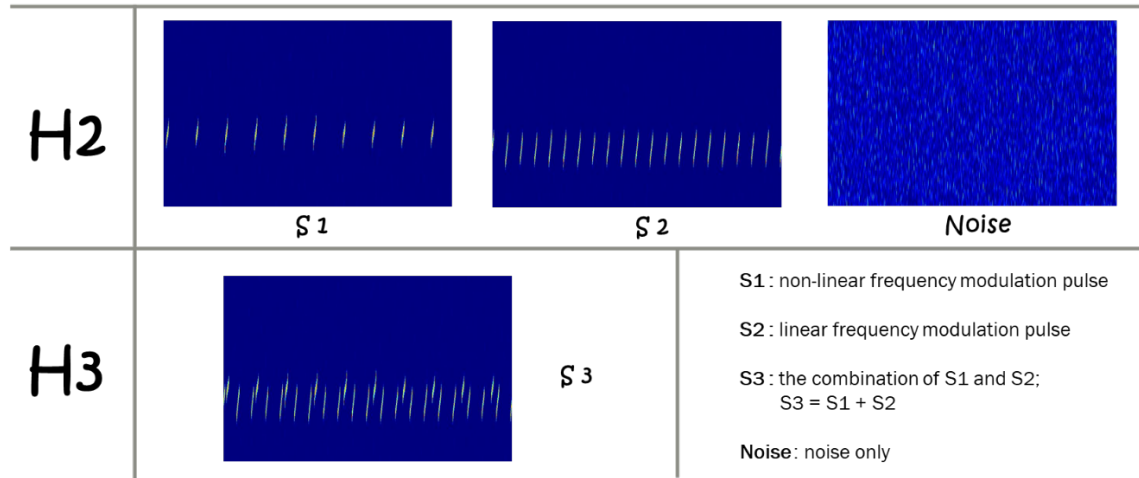


Figure 4.2: RF signals and hypotheses of OOD detection.

H3 between hypotheses H2 and H3. **ID** signals classification refers to the classification among signals: S1, S2 and Noise which are considered as the hypothesis of H2.

Dataset Split

As described in Section 4.1.1, our simulation settings involve four signals: S1, a non-linear frequency modulation pulse; S2, a linear frequency modulation pulse; S3, a combined signal of S1 and S2; and Noise, an **AWGN**, noise signal. Noise, S1, and S2 are considered as the seen signals (**ID** signals), while S3 is the unseen class of signals (**OOD** signal). The detailed properties of each signal are available in Section 4.1.1.

Figure 4.3 provides an overview of the dataset split. The dataset consists of a total of 4000 samples (**STFT** plots of the sensed signals), with each signal comprising 1000 samples. For each **ID** signal (Noise, S1, or S2), 80% of the 1000 samples are allocated for training the corresponding model (model Noise, model S1, or model S2), and the remaining 20% are included in the test set. In the case of the **OOD** signal (S3), 20% of the 1000 samples are reserved as one-fourth of the test set. It is important to note

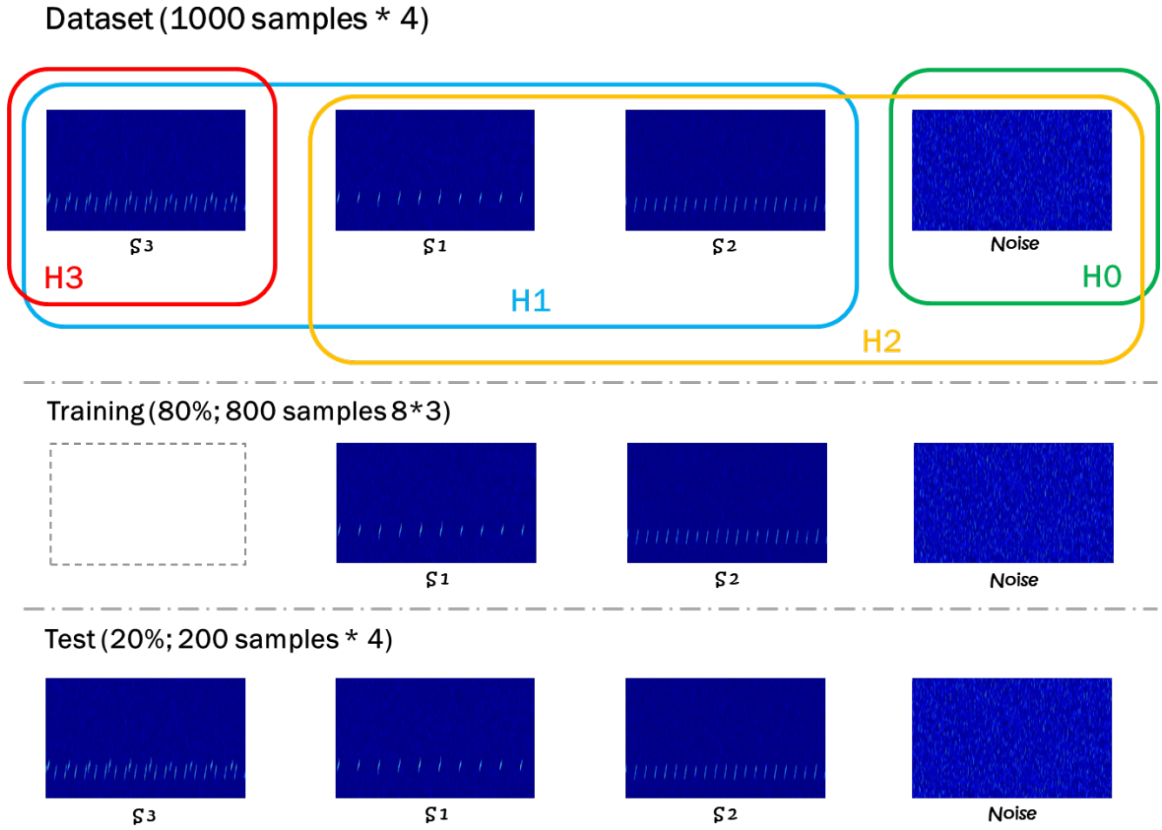


Figure 4.3: Dataset split overview.

that the unseen class of signals does not participate in the training of any neural network of our proposed system.

For spectrum sensing in our problem settings, there are three types of H1 hypotheses (S1, S2, and S3) representing different PU signals and one type of H0 hypothesis (Noise) representing noise. Whereas, for OOD detection, we consider three types of H2 hypotheses (Noise, S1, and S2; ID) and one type of H3 hypothesis (S3; OOD).

In our simulation settings, spectrum sensing refers to the classification between hypotheses H1 and H0, whereas OOD detection refers to the classification between hypotheses H3 and H2. ID signals classification refers to the classification among

signals: S1, S2 and Noise.

Neural Network Structure and Parameters

In our experiments, the system's sensing time is 60ms. We employ an autoencoder as the neural network architecture in our system. Autoencoders are a type of neural network used for unsupervised learning, specifically designed to efficiently encode and decode data. They consist of an encoder component responsible for mapping the input data to a lower-dimensional representation, and a decoder component that reconstructs the original input from the encoded representation [63].

Autoencoders are trained to learn compressed representations of the input data by encoding and decoding. The encoder takes the input data and maps it to a lower-dimensional latent space representation, while the decoder reconstructs the original input from the latent space. The purpose of an autoencoder is to effectively capture and preserve important features of the input data in the latent space [64].

Figure 4.4 illustrates the typical structure of a convolutional autoencoder, including an encoder, a code and a decoder component.

There are several variants of autoencoders, such as undercomplete autoencoders, sparse autoencoders, denoising autoencoders, contractive autoencoders, and variational autoencoders [63]. In this thesis, we have employed a basic standard convolutional autoencoder without explicit regularization terms. The fact that convolutional neural networks exhibit translation invariance is widely acknowledged. And Convolutional networks achieve this property by employing a combination of convolutional and pooling layers in their architecture. This choice allows us to better observe the impact of the customized function (EWMSE) by comparing its performance with

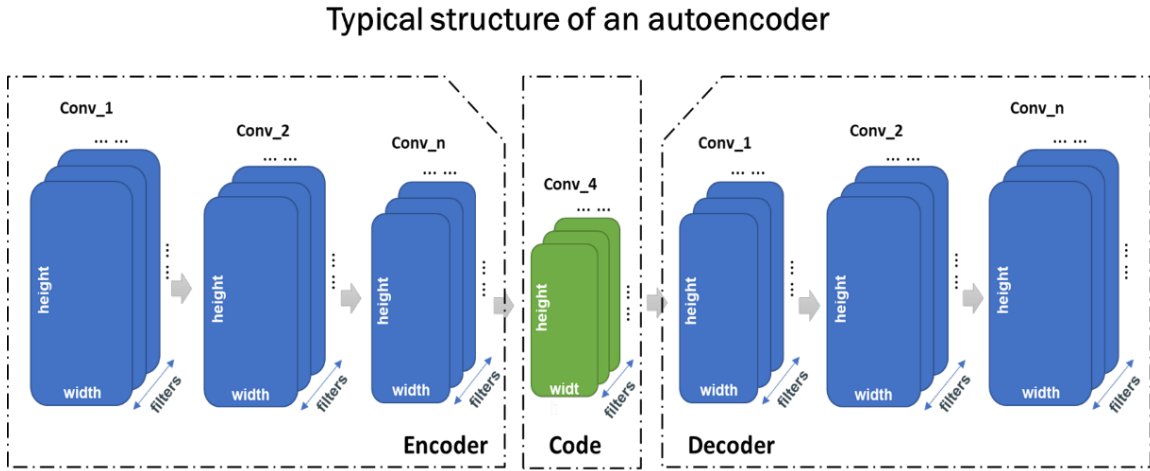


Figure 4.4: The typical structure of a convolutional autoencoder.

that of [MSE](#). Each model in our proposed system follows a same autoencoder structure, which is described in detail in [Figure 4.5](#). The encoder part of the autoencoder consists of three convolutional layers, each followed by a max pooling layer. On the other hand, the decoder part utilizes three transpose convolutional layers to ensure that the output reconstruction of an autoencoder matches the size of its input [STFT](#) plots. Throughout both the training and inference phases, [EWMSE](#) is employed. During training, [EWMSE](#) acts as a customized loss function, with foreground pixels assigned a weight of 50 and background pixels assigned a weight of 1. In the inference phase, [EWMSE](#) function is used to calculate the [OOD](#) scores, with foreground pixels assigned a weight of 1 and background pixels assigned a weight of 0. For neural network optimization, we utilize the Adam algorithm [\[65\]](#) with a learning rate of 0.0001.

Models: parameters of Noise model / S1 model / S2 model

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 492, 744, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 246, 372, 64)	0
conv2d_1 (Conv2D)	(None, 246, 372, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 82, 124, 128)	0
conv2d_2 (Conv2D)	(None, 82, 124, 512)	590336
max_pooling2d_2 (MaxPooling2D)	(None, 41, 62, 512)	0
conv2d_transpose (Conv2DTranspose)	(None, 82, 124, 512)	2359808
conv2d_transpose_1 (Conv2DTranspose)	(None, 246, 372, 128)	589952
conv2d_transpose_2 (Conv2DTranspose)	(None, 492, 744, 64)	73792
conv2d_3 (Conv2D)	(None, 492, 744, 3)	1731
=====		
Total params: 3,691,267		
Trainable params: 3,691,267		
Non-trainable params: 0		

Figure 4.5: The detailed structure of the adopted autoencoder.

4.2 Main Results

Our proposed system performs both spectrum sensing and signal classification tasks. In our experiments, we consider a range of SNR from -15dB to 10dB, and we have achieved promising performance in both tasks.

To evaluate the discriminability of the OOD detector of our proposed system, we have extended the concept of ROC curve to accommodate multiple thresholds models, see Section 3.6. An example of the multi-threshold ROC curve is depicted in Figure 4.6.

Figure 4.7 shows the AUROC of OOD signal (S3) detection. The results indicate that an AUROC of 0.92 and 0.99 has been achieved for SNR values of -15dB and -14dB, respectively. When the SNR is larger than -13dB, a perfect AUROC of 1.0 has been achieved. These results demonstrate that our proposed system exhibits strong

resistance to low **SNR** environments in detecting the **OOD** signal.

Base on the **ROC** curve, we can identify a suitable set of thresholds ($T_{\text{OOD-Noise}}$, $T_{\text{OOD-S1}}$, and $T_{\text{OOD-S2}}$) for calculating the performance of spectrum sensing and signal classification. If we obtain a perfect **AUROC** (1.0), we will use the set of thresholds that achieves 1.0 **TPR** and 0.0 **FPR** for calculating the performance of spectrum sensing and signal classification. However, if we have a non-perfect **ROC** curve (**AUROC** is smaller than 1.0), we will use the set of thresholds that achieves a 0.95 **TPR** for calculating the performance of spectrum sensing and signal classification.

4.2.1 Performance of Spectrum Sensing

The performance of spectrum sensing task is illustrated in Figure 4.8. It is evident that our proposed system successfully detects all the H1 samples (test samples of signals of S1, S2, and S3) in the test set when **SNR** is larger than -14dB. Analyzing the confusion matrices presented in Figure 4.10, we observe that as the **SNR** decreases, our system begins to exhibit confusion within different H1 signals at -10dB. However, it consistently distinguishes H1 signals (S1, S2, and S3) from the H0 signal (Noise) when the **SNR** is above -14dB.

Our proposed system achieves a remarkable **Pd** of **PU**, up to 93.8% even at an **SNR** as low as -15dB, while maintaining a **Pfa** of 0%. This demonstrates the exceptional performance of our system in the spectrum sensing task, combining high **Pd** and low **Pfa**. Furthermore, it showcases the system's resilience in low **SNR** environments for the task of spectrum sensing. For more detailed numerical results, please refer to Appendix A.5.

4.2.2 Performance of ID Signal Classification (Classification among H2)

The performance of the task of ID signal classification is presented in Figure 4.9, and the corresponding confusion matrices are provided in Figure 4.10. It is evident that our proposed system achieves exceptional signal classification accuracy when the SNR is above -13dB, with an average accuracy of 99.0%. For the scenarios with SNR of -14dB and -15dB, the average accuracies are 98.4% and 73.3% respectively. These results highlight the robustness of our proposed system in low SNR environments for signal classification task. For a more comprehensive and detailed analysis of the numerical results, please refer to Appendix A.5.

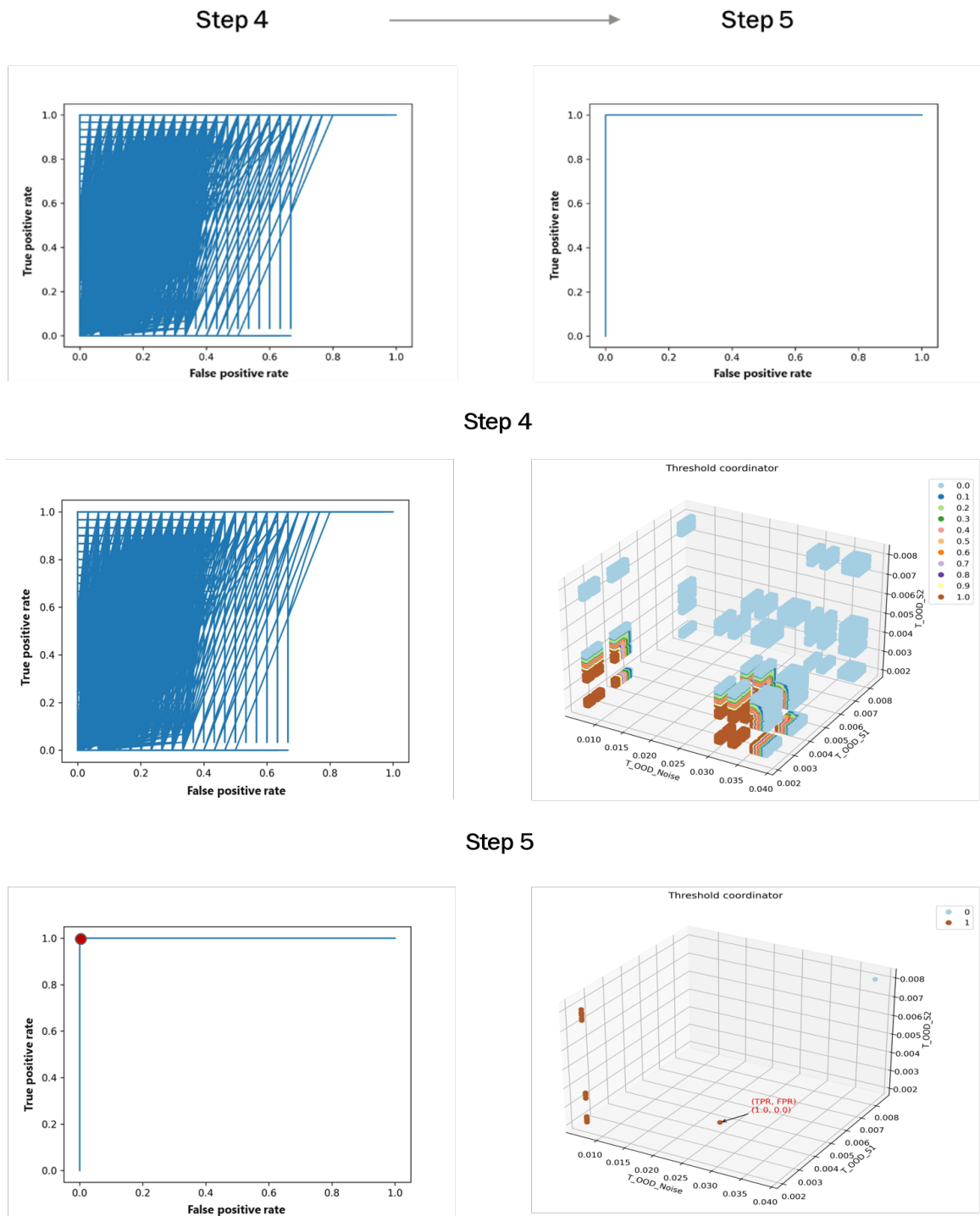


Figure 4.6: An example of multi-threshold ROC plot process.

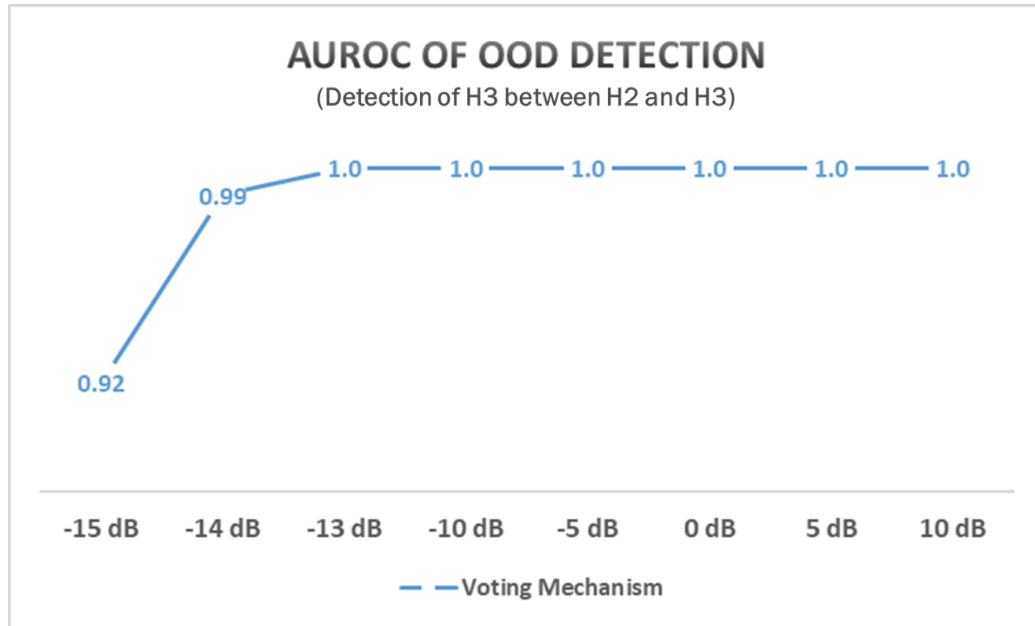


Figure 4.7: AUROC of OOD detection (detection of H3 between H2 and H3).

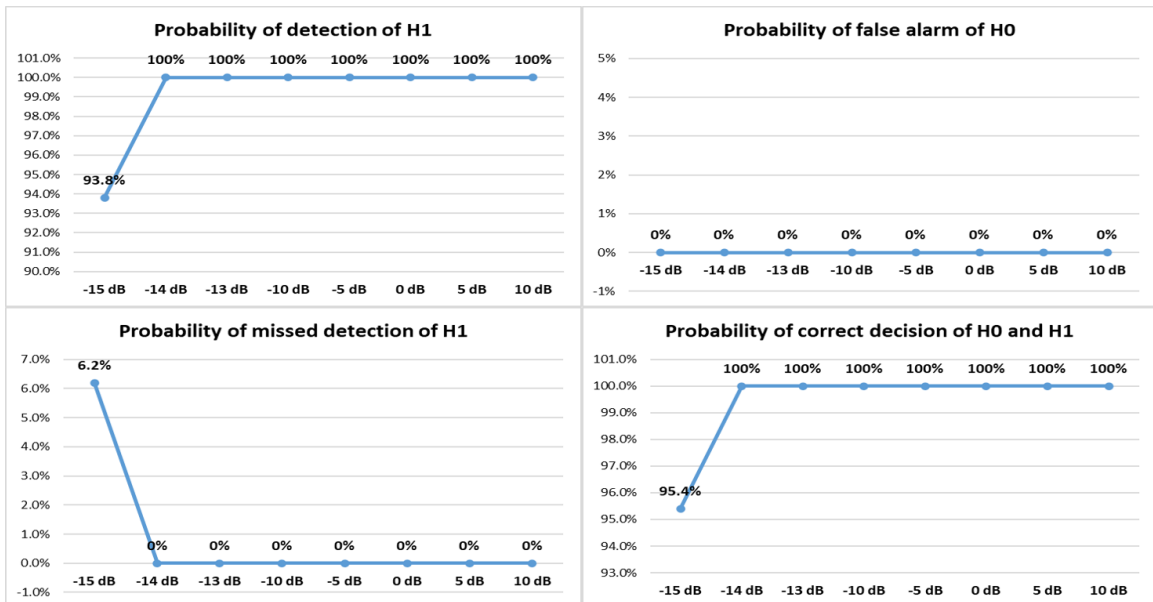


Figure 4.8: Performance of spectrum sensing (sensing H1 between H0 and H1).

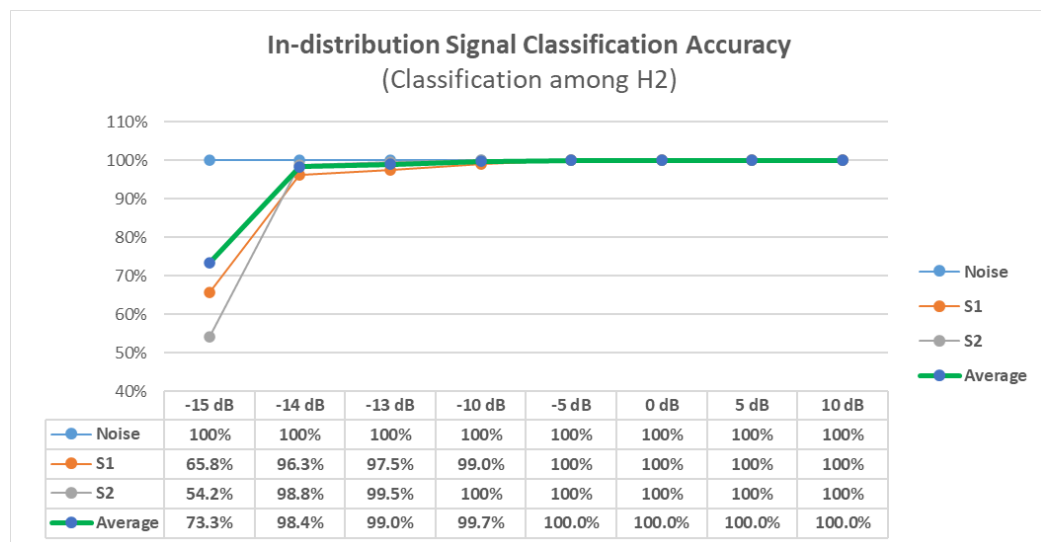


Figure 4.9: Performance of in-distribution signal classification (classification among H2).

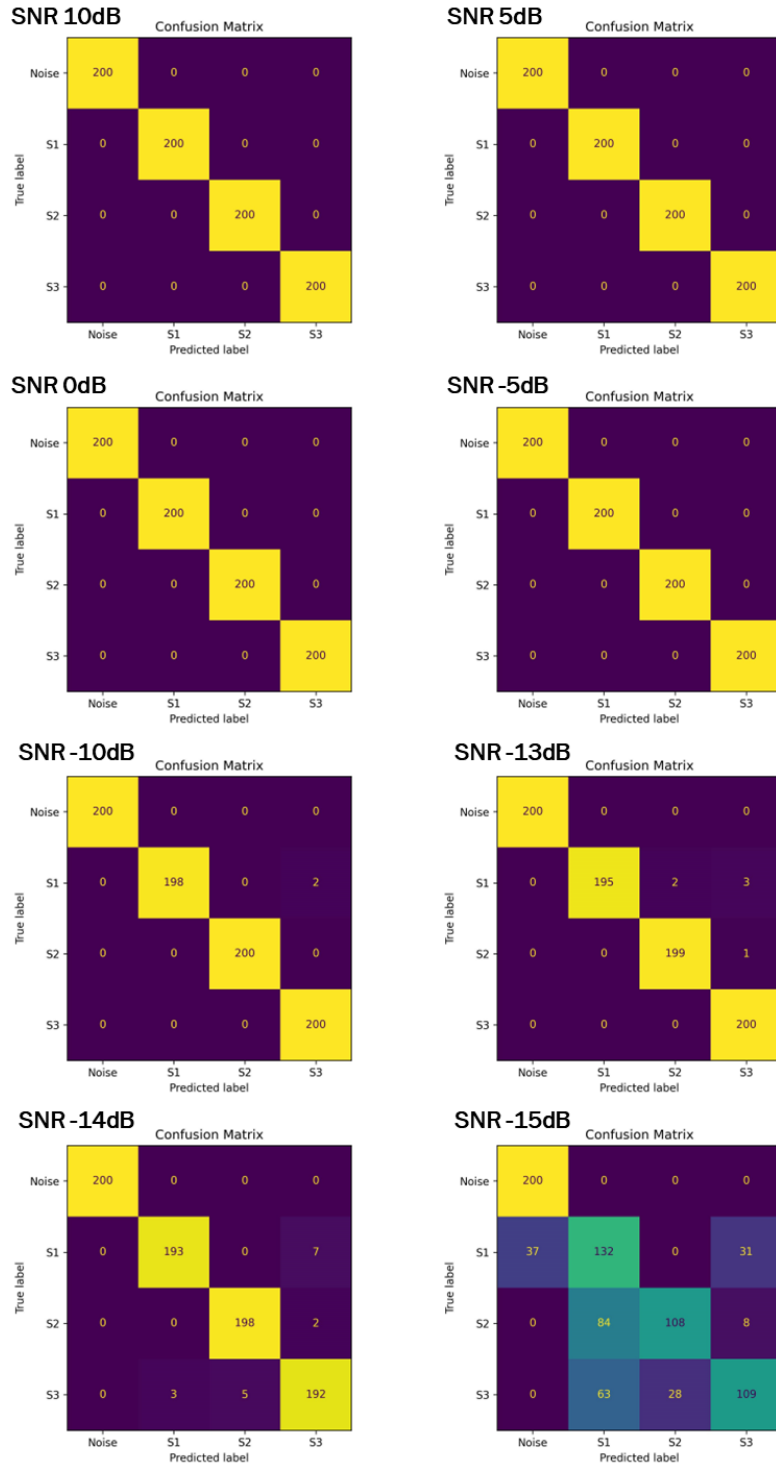


Figure 4.10: Confusion matrixes of four-signal classification.

Chapter 5

Conclusion

5.1 Future Work and Conclusions

In conclusion, this thesis has explored the application of out-of-distribution (OOD) detection-based spectrum sensing and signal classification. Through a comprehensive literature review and empirical investigation, this study has successfully addressed the objectives outlined at the beginning of the thesis.

Firstly, our proposed method has demonstrated exceptional effectiveness. Our scheme shows remarkable resilience in low SNR environments, achieving 100% accuracy in spectrum sensing and 97.7% accuracy in signal classification in our simulation settings, even at a SNR as low as -14dB. A key feature of our proposed system is its capability to detect and accurately categorize test samples that have never been encountered during training. This distinguishing feature sets our research apart from existing studies in the field and is essential for enhancing the robustness, reliability, and security of AI systems in general, particularly in real-world applications. Moreover, the design of our scheme exhibits good scalability and flexibility, facilitating the assignment of weights to specific signals for sensing and classification, if required.

Despite the promising performance of our proposed method, there are several aspects that can be further improved. One such aspect pertains to the assumption that the SNR is known at the receiver during experiments. In real-world deployments of spectrum sensing and signal classification systems, the receiver lacks knowledge of the SNR. Consequently, our proposed system must perform ‘blind’ detection, resulting in high computational complexity. Addressing this computational burden requires further research attention in the context of unknown SNR scenarios. Additionally, in the data preprocessing stage, we employed varying window lengths of Short-Time Fourier Transform (STFT) for different SNR ranges, which may not be practical for deployment purposes. Future research could focus on utilizing a unified setting of window length of STFT for all considered SNR ranges.

In general, having a large test dataset enhances the model’s reliability and ability to generalize to new data. We plan to apply our proposed scheme to a larger dataset. Additionally, our proposed system has the potential to detect modulation schemes, which could be a focus of our future work. Lastly, a comparison between our scheme and the work of DetectNet [16] has also been planned.

Overall, this thesis has contributed to the existing knowledge base by providing new insights into deep learning-based spectrum sensing and signal classification. We firmly believe that our findings will contribute significantly to the advancement of cognitive radio technology and the efficient utilization of the spectrum, thereby addressing the critical need for more reliable and sustainable wireless communication systems.

Bibliography

- [1] K. M. Ting, “Confusion matrix,” *Encyclopedia of Machine Learning and Data Mining*, pp. 260–260, 2017. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-1-4899-7687-1_50?view=modern
- [2] B. M. Pati, M. Kaneko, and A. Taparugssanagorn, “A deep convolutional neural network based transfer learning method for non-cooperative spectrum sensing,” *IEEE Access*, vol. 8, pp. 164 529–164 545, 2020.
- [3] D. McNicol, *A Primer of Signal Detection Theory*. Psychology Press, 01 2005. [Online]. Available: https://books.google.ca/books?hl=en&lr=&id=GYN5AgAAQBAJ&oi=fnd&pg=PA3&dq=A+Primer+of+Signal+Detection+Theory&ots=diOMRiF5Du&sig=W1s9wg-IClijNCsFuRdY1DX-LbA&redir_esc=y#v=onepage&q=A%20Primer%20of%20Signal%20Detection%20Theory&f=false
- [4] C. Sammut and G. I. Webb, Eds., *ROC Curve*. Boston, MA: Springer US, 2017, pp. 1116–1116. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_735
- [5] P. A. Flach, *ROC Analysis*. Boston, MA: Springer US, 2017, pp. 1109–1116. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_739

-
- [6] J. A. Swets, *Signal Detection Theory and ROC Analysis in Psychology and Diagnostics*. Psychology Press, 02 2014.
- [7] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE personal communications*, vol. 6, no. 4, pp. 13–18, 1999.
- [8] A. K. Farraj, E. M. Hammad, and S. L. Miller, “Performance studies for spectrum-sharing cognitive radios under outage probability constraint,” *Advances in Wireless Technologies and Telecommunication*, pp. 345–367, 2015.
- [9] S. Atapattu, C. Tellambura, and H. Jiang, *Energy detection for spectrum sensing in cognitive radio*. Springer, 2014.
- [10] K. Kim, I. A. Akbar, K. K. Bae, J.-S. Um, C. M. Spooner, and J. H. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. IEEE, 2007, pp. 212–215.
- [11] S. Kapoor, S. Rao, and G. Singh, “Opportunistic spectrum sensing by employing matched filter in cognitive radio network,” in *2011 International Conference on Communication Systems and Network Technologies*. IEEE, 2011, pp. 580–583.
- [12] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, “Application of compressive sensing in cognitive radio communications: A survey,” *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 1838–1860, 2016.
- [13] O. Mangasarian and D. Musicant, “Active support vector machine classification,” *Advances in neural information processing systems*, vol. 13, 2000.

-
- [14] W. Ejaz, N. ul Hasan, S. Aslam, and H. S. Kim, "Fuzzy logic based spectrum sensing for cognitive radio networks," in *2011 Fifth International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, 2011, pp. 185–189.
- [15] D. Treeumnuk and D. C. Popescu, "Using hidden markov models to enable performance awareness and noise variance estimation for energy detection in cognitive radio," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2012, pp. 1–5.
- [16] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727–1730, 2019.
- [17] J. Shi, S. Hong, C. Cai, Y. Wang, H. Huang, and G. Gui, "Deep learning-based automatic modulation recognition method in the presence of phase offset," *IEEE Access*, vol. 8, pp. 42 841–42 847, 2020.
- [18] K. Tekbıyık, Ö. Akbunar, A. R. Ekti, A. Görçin, G. K. Kurt, and K. A. Qaraqe, "Spectrum sensing and signal identification with deep learning based on spectral correlation function," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 514–10 527, 2021.
- [19] T. Xu and I. Darwazeh, "Deep learning for over-the-air non-orthogonal signal classification," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

-
- [20] X. Zha, H. Peng, X. Qin, G. Li, and S. Yang, "A deep learning framework for signal detection and modulation classification," *Sensors*, vol. 19, no. 18, p. 4042, 2019.
- [21] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.
- [22] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Federated learning for distributed spectrum sensing in nextg communication networks," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, vol. 12113. SPIE, 2022, pp. 472–478.
- [23] J. Xie, J. Fang, C. Liu, and L. Yang, "Unsupervised deep spectrum sensing: A variational auto-encoder based approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5307–5319, 2020.
- [24] Q. Cheng, Z. Shi, D. N. Nguyen, and E. Dutkiewicz, "Deep learning network based spectrum sensing methods for ofdm systems," *arXiv preprint arXiv:1807.09414*, 2018.
- [25] Y. Li, B. Wang, G. Shao, S. Shao, and X. Pei, "Blind detection of underwater acoustic communication signals based on deep learning," *IEEE Access*, vol. 8, pp. 204 114–204 131, 2020.

-
- [26] S. Subray, S. Tschimben, and K. Gifford, "Towards enhancing spectrum sensing: signal classification using autoencoders," *IEEE Access*, vol. 9, pp. 82 288–82 299, 2021.
- [27] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012.
- [28] F. Salahdine, "Spectrum sensing techniques for cognitive radio networks," *arXiv preprint arXiv:1710.02668*, 2017.
- [29] S. S. Soliman and S.-Z. Hsue, "Signal classification using statistical moments," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 908–916, 1992.
- [30] E. Jones, P. Runkle, N. Dasgupta, L. Couchman, and L. Carin, "Genetic algorithm wavelet design for signal classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 890–895, 2001.
- [31] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and knn," *IEEE Transactions on wireless communications*, vol. 11, no. 8, pp. 2742–2750, 2012.
- [32] Y. Paul, V. Goyal, and R. A. Jaswal, "Comparative analysis between svm & knn classifier for emg signal classification on elementary time domain features," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*. IEEE, 2017, pp. 169–175.
- [33] H. Hu, Y. Wang, and J. Song, "Signal classification based on spectral correlation analysis and svm in cognitive radio," in *22nd International Conference on Advanced Information Networking and Applications (AINA 2008)*. IEEE, 2008, pp. 883–887.
-

-
- [34] D. C. Toledo-Pérez, J. Rodríguez-Reséndiz, R. A. Gómez-Loenzo, and J. Jauregui-Correa, “Support vector machine-based emg signal classification techniques: A review,” *Applied Sciences*, vol. 9, no. 20, p. 4402, 2019.
- [35] C. Sun, W. Zhang, and K. B. Letaief, “Cooperative spectrum sensing for cognitive radios under bandwidth constraints,” in *2007 IEEE wireless communications and networking conference*. IEEE, 2007, pp. 1–5.
- [36] M. Saber, A. El Rharras, R. Saadane, A. H. Kharraz, and A. Chehri, “An optimized spectrum sensing implementation based on svm, knn and tree algorithms,” in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2019, pp. 383–389.
- [37] Y. Zhang, P. Wan, S. Zhang, Y. Wang, N. Li *et al.*, “A spectrum sensing method based on signal feature and clustering algorithm in cognitive wireless multimedia sensor networks,” *Advances in Multimedia*, vol. 2017, 2017.
- [38] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, “Machine learning techniques for cooperative spectrum sensing in cognitive radio networks,” *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209–2221, 2013.
- [39] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection: A survey,” *arXiv preprint arXiv:2110.11334*, 2021.
- [40] B. Schölkopf, C. J. Burges, A. J. Smola *et al.*, *Advances in kernel methods: support vector learning*. MIT press, 1999.

-
- [41] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.
- [42] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun *et al.*, “Openood: Benchmarking generalized out-of-distribution detection,” *arXiv preprint arXiv:2210.07242*, 2022.
- [43] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” *arXiv preprint arXiv:1610.02136*, 2016.
- [44] K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota, “Practical deep learning with bayesian principles,” *Advances in neural information processing systems*, vol. 32, 2019.
- [45] Y. Li and N. Vasconcelos, “Background data resampling for outlier-aware classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 218–13 227.
- [46] D. Mandal, S. Narayan, S. K. Dwivedi, V. Gupta, S. Ahmed, F. S. Khan, and L. Shao, “Out-of-distribution detection for generalized zero-shot action recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9977–9985.
- [47] X. Du, Z. Wang, M. Cai, and Y. Li, “VOS: learning what you don’t know by virtual outlier synthesis,” *CoRR*, vol. abs/2202.01197, 2022. [Online]. Available: <https://arxiv.org/abs/2202.01197>

- [48] R. Huang, A. Geng, and Y. Li, “On the importance of gradients for detecting distributional shifts in the wild,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 677–689. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/063e26c670d07bb7c4d30e6fc69fe056-Paper.pdf
- [49] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 481–490.
- [50] J. Serrà, D. Álvarez, V. Gómez, O. Slizovskaia, J. F. Núñez, and J. Luque, “Input complexity and out-of-distribution detection with likelihood-based generative models,” *CoRR*, vol. abs/1909.11480, 2019. [Online]. Available: <http://arxiv.org/abs/1909.11480>
- [51] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, “Do deep generative models know what they don’t know?” 2019.
- [52] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf
- [53] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *Proceedings*

- of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9690–9700. [Online]. Available: <https://proceedings.mlr.press/v119/van-amersfoort20a.html>
- [54] Y. Ming, Y. Sun, O. Dia, and Y. Li, “How to exploit hyperspherical embeddings for out-of-distribution detection?” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=aEFaE0W5pAd>
- [55] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, “Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance,” *CoRR*, vol. abs/1812.02765, 2018. [Online]. Available: <http://arxiv.org/abs/1812.02765>
- [56] Y. Zhou, “Rethinking reconstruction autoencoder-based out-of-distribution detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 7379–7387.
- [57] Y. Yang, R. Gao, e. S. Xu, Qiang”, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, “Out-of-distribution detection with semantic mismatch under,” in *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 373–390.
- [58] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

- [59] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 05 2015.
- [60] Pace and P. E, *Detecting and Classifying Low Probability of Intercept Radar*. Artech House, 2009.
- [61] M. Vetterli and C. Herley, “Wavelets and filter banks: theory and design,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 2207–2232, 1992.
- [62] Z. SONG, Y. GAO, and R. TAFAZOLLI, “Ieice trans - a survey on spectrum sensing and learning technologies for 6g,” https://search.ieice.org/bin/summary.php?id=e104-b_10_1207, October 2021, (Accessed on 05/15/2023).
- [63] “Introduction to autoencoders.” Jeremy Jordan, 03 2018. [Online]. Available: <https://www.jeremyjordan.me/autoencoders/>
- [64] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 473–480. [Online]. Available: <https://doi.org/10.1145/1273496.1273556>
- [65] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [66] OpenCV, “Opencv: Image thresholding,” docs.opencv.org, 2023. [Online]. Available: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html
- [67] J. Yousefi, “Image binarization using otsu thresholding algorithm,” *Ontario, Canada: University of Guelph*, vol. 10, 2011.

-
- [68] S. L. Bangare, A. Dubal, P. S. Bangare, and S. Patil, “Reviewing otsu’s method for image thresholding,” *International Journal of Applied Engineering Research*, vol. 10, pp. 21 777–21 783, 05 2015.
- [69] “Morphological transformations — opencv-python tutorials beta documentation,” opencv24-python-tutorials.readthedocs.io, 2023. [Online]. Available: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

Appendix A

A.1 OOD (unseen signal) detection and ID (seen signals) classification examples of the scenarios of vote-a to h

Note that the OOD (unseen) signal is S3, while the ID (seen) signals include Noise, S1, and S2. The decision for detecting OOD samples is determined by Vote a. The classification of ID (seen) signals, on the other hand, is performed using the hard rule outlined in Section 3.3.

The scenarios of vote-a to h are illustrated in Figure A.1, Figure A.2, Figure A.3, Figure A.4, Figure A.5, Figure A.6, Figure A.7, Figure A.8.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

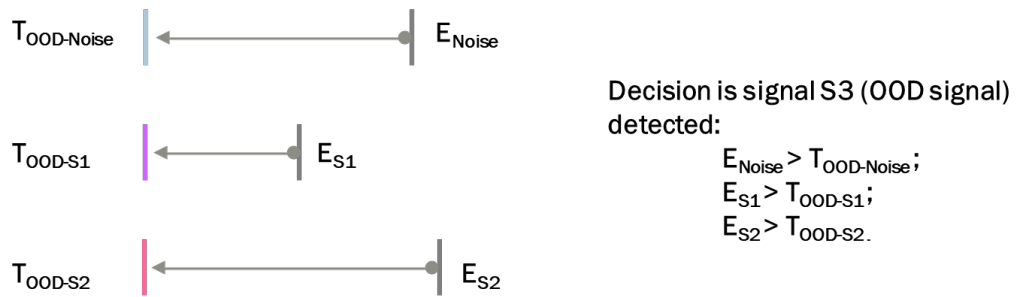


Figure A.1: Scenario of vote-a: triggers the decision of OOD signal is detected.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

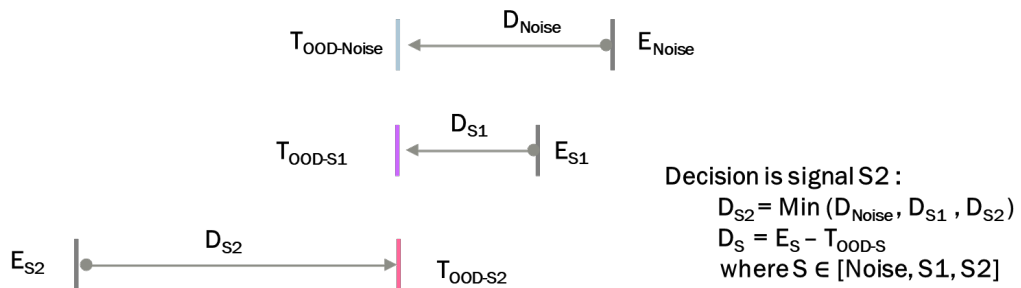


Figure A.2: Scenario of vote-b: ID signals classification result is S2.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

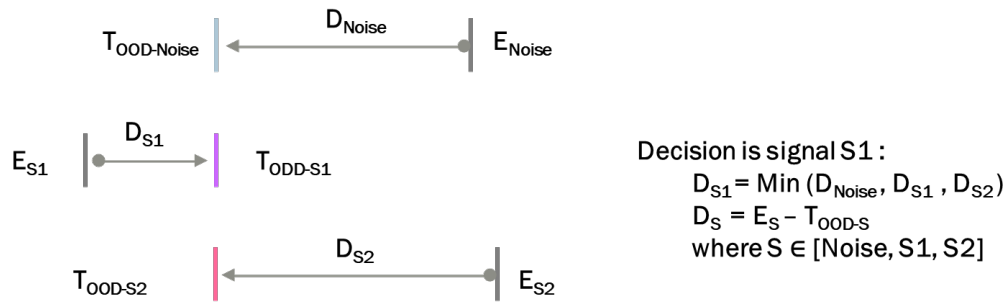


Figure A.3: Scenario of vote-c: ID signals classification result is S1.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

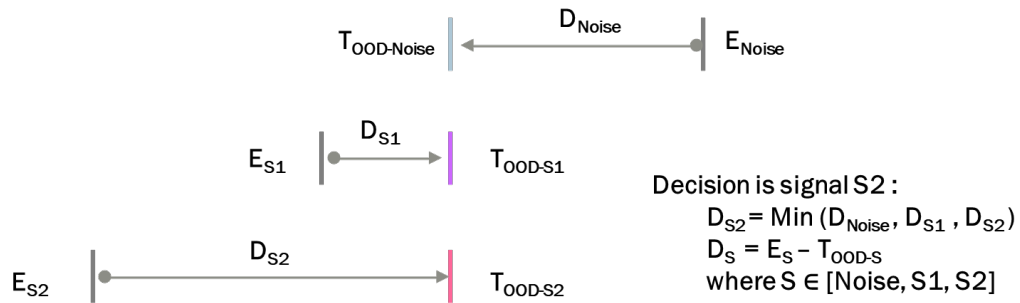


Figure A.4: Scenario of vote-d: ID signals classification result is S2.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

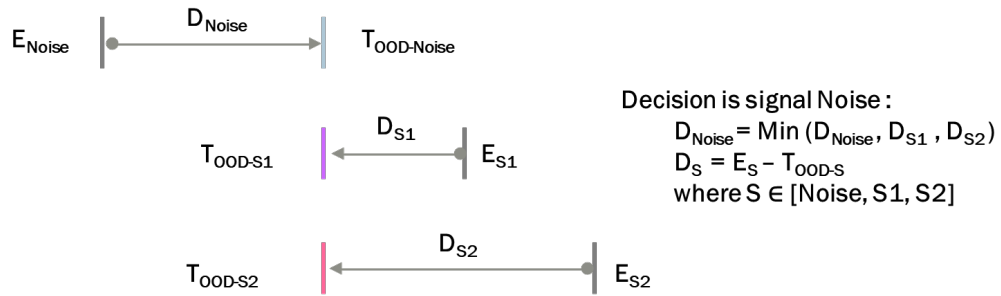


Figure A.5: Scenario of vote-e: ID signals classification result is Noise.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

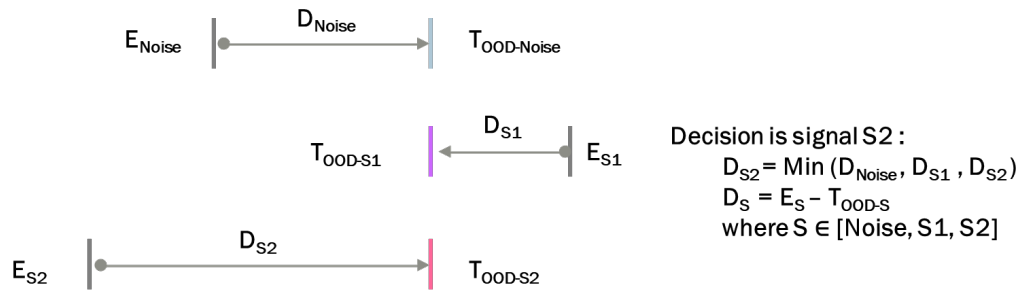


Figure A.6: Scenario of vote-f: ID signals classification result is S2.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

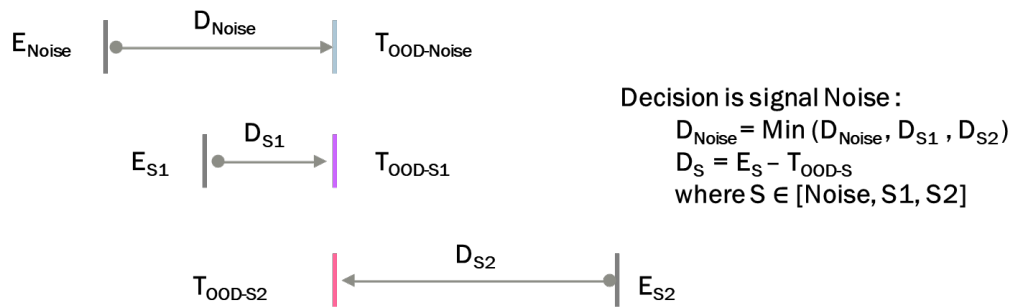


Figure A.7: Scenario of vote-g: ID signals classification result is Noise.

Autoencoder (Model)	Possible received votes at the OOD arbitrator							
	a	b	c	d	e	f	g	h
Noise	0	0	0	0	1	1	1	1
S1	0	0	1	1	0	0	1	1
S2	0	1	0	1	0	1	0	1

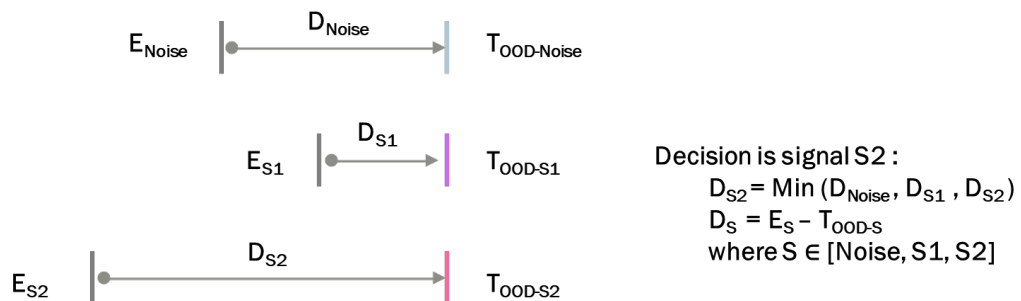


Figure A.8: Scenario of vote-h: ID signals classification result is S2.

A.2 The effect of the parameter of window length

Figure A.9 demonstrates an example of how the parameter of window length affects the STFT presentation plot of sensed signals. Taking signal S1 as an example, it can be observed that for a SNR of 10dB, as the window length increases, the signal area (the size of the signal appearing in the plot) in the STFT plot becomes smaller and smaller. On the other hand, for a SNR of -5dB, as the window length increases, the background noise becomes weaker and weaker. This phenomenon occurs because, with an increasing window length, the signal $X(t)$ approximates stationarity, resulting in fewer abrupt changes over the window's duration.

Therefore, we employ different window lengths for different SNR ranges. For high SNRs, we utilize relatively smaller window lengths, while in low SNR regions, we adopt relatively larger window length settings. The specific window length for different SNRs is shown in Table A.3.

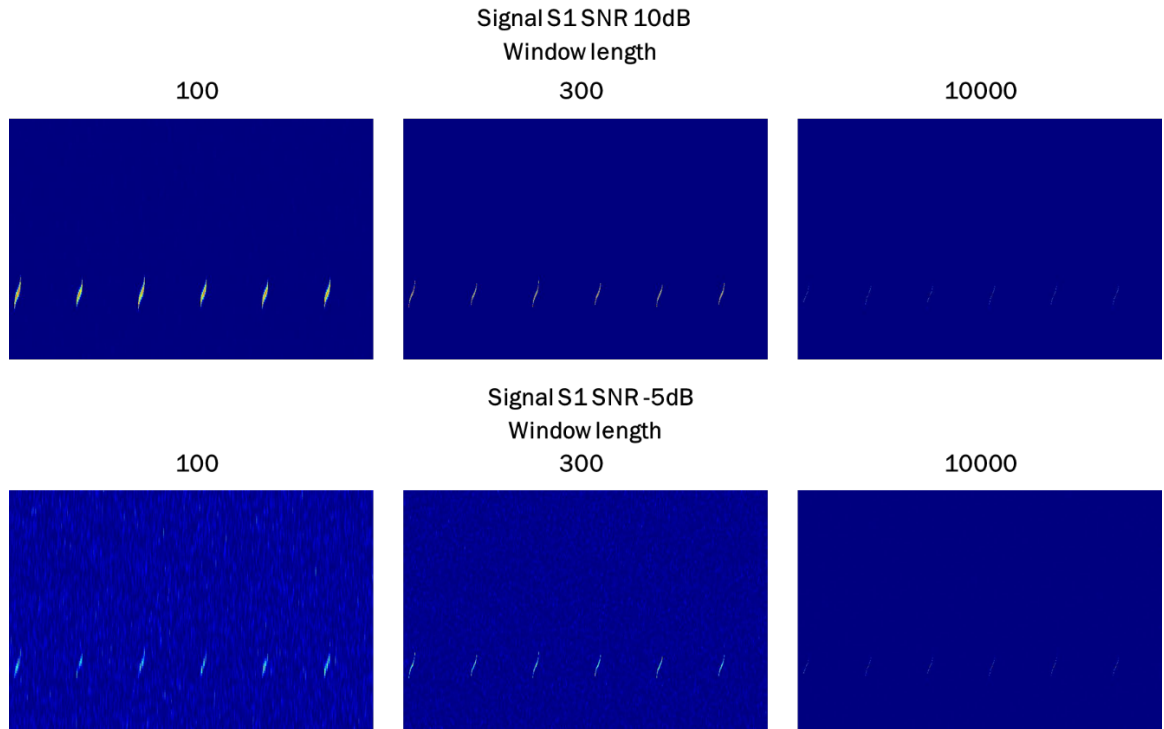


Figure A.9: An example of the how the parameter of window length affects STFT presentation plots.

A.3 Introduction of background and foreground separation algorithm

To harness the capabilities of the [EWMSE](#) technique discussed in Section [3.5](#), we utilize an existing image thresholding algorithm [\[66\]](#) from the open computer vision library to separate the background and foreground in a given image. Specifically, we employ Otsu's binarization [\[67\]](#), which is an adaptive thresholding method for image binarization. Otsu's method considers all possible threshold values to distinguish the background from the foreground. It calculates the variance within each cluster and selects the threshold value that minimizes the weighted sum of these variances

[68]. Subsequently, we apply closing morphological transformations [69] to the output obtained from Otsu's method to obtain a closed form of the foreground in the input image. "Closing", which involves 'dilation' followed by 'erosion', is effective in closing small holes within foreground objects or small black points on the object" [69]. The algorithm for background and foreground separation is illustrated in Figure A.10.

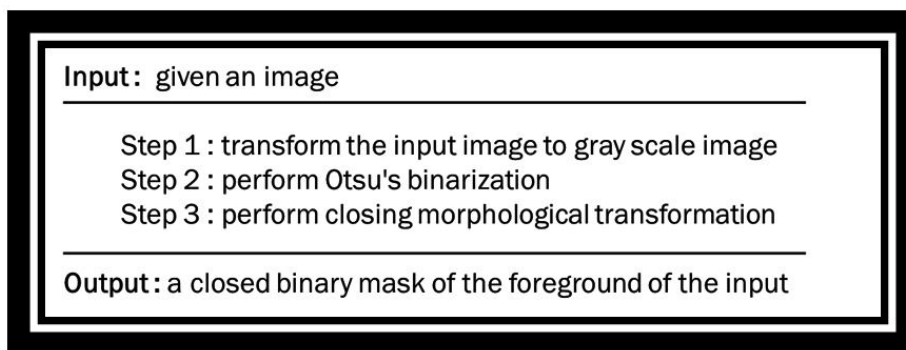


Figure A.10: Algorithm for background and foreground separation.

A.4 Performance comparison between MSE and EWMSE

This section aims to demonstrate the effectiveness of introducing the EWMSE function by comparing it with the MSE function. The system architecture utilized in this section is sequential screening, as depicted in Figure A.11. During training, each of the three models (model Noise, model S1 or model S2) is trained on a specific type of ID signals (Noise, S1, or S2) accordingly. During the testing phase, the test samples are sequentially processed by these three models. Each model is responsible for identifying the type of ID signals the model was trained with. If the reconstruction error (OOD score) of a test sample is sufficiently small (below a threshold), the test

sample is classified as the signal type associated with the corresponding model.

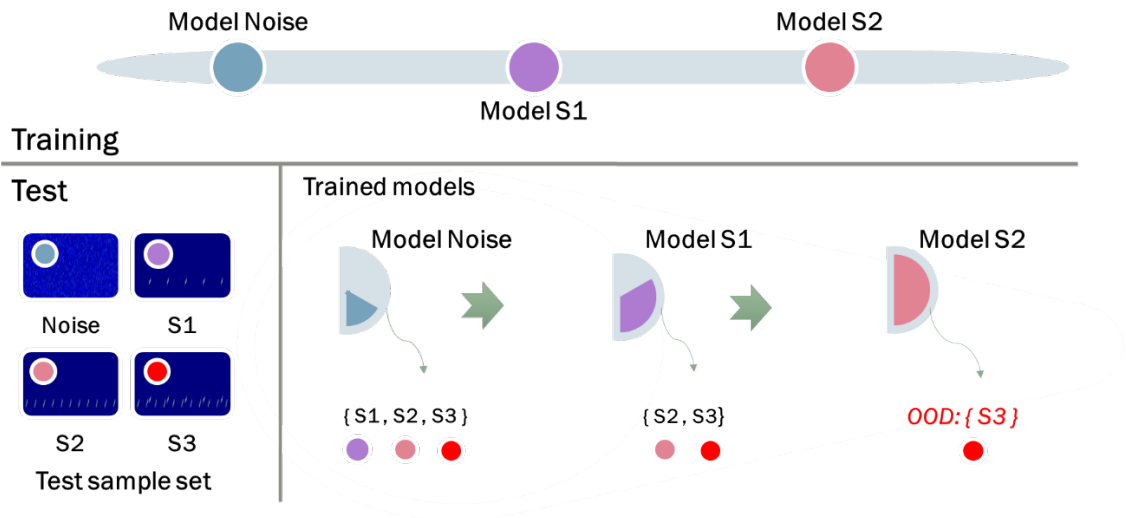


Figure A.11: Sequential screening overview.

To streamline the collection procedure for performance, we introduced the concept of an engineering mode test sample set. This test sample set consists of four samples, with each sample representing one of the four signals. It is important to note that the concept of the test sample set will be removed when evaluating the performance of our proposed system (voting mechanism, see Figure 3.1).

Note that the autoencoders adopted in this section are exactly identical with those autoencoders utilized in the voting mechanism shown in Figure 3.1.

Figure A.12 illustrates the performance comparison of sequential screening using MSE loss and EWMSE loss. It is evident from the results that the utilization of EWMSE loss leads to a notable enhancement in accuracy. More detailed results are

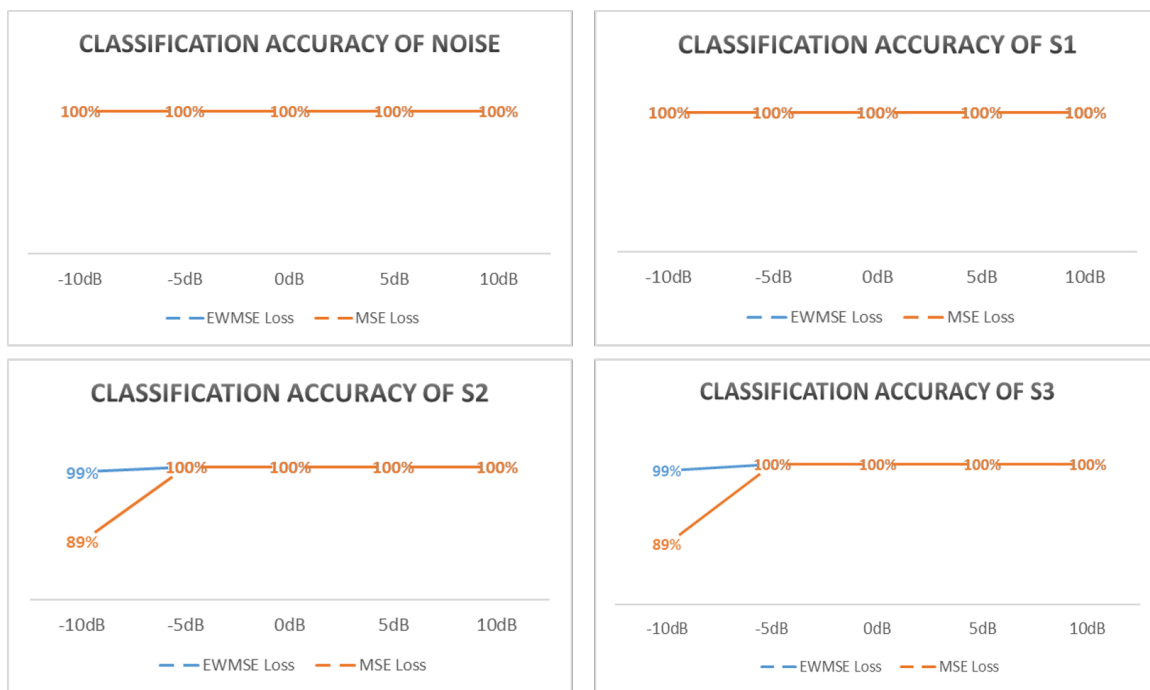


Figure A.12: Performance comparison of sequential screening between MSE and EWMSE.

available in Table A.1 and Table A.2

A.5 Detailed numeric results of spectrum sensing and signal classification

See Table A.3. Note that $STFT$ N_{seq} means $STFT$ window length in terms of number of I/Q samples.

A.6 Experiments of different signal representations

To investigate a suitable signal representation method for the input of a neural network system, we conducted the experiments described in this section. We utilized a simple fully connected neural network consisting of three fully connected layers,

Test sample set sequential through each trained model (MSE Loss)						
SNR	STFT Nseq	Signal	Four-signal classification performance			
			Accuracy	Run 1	Run 2	Run 3
10 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
5 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
0 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
-5 dB	300	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
-10 dB	600	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	0.887	0.9 (180/200)	0.86 (172/200)	0.9 (180/200)
		S3 (OOD)	0.887	0.9 (180/200)	0.86 (172/200)	0.9 (180/200)
Note	STFT Nseq means STFT window length in terms of the number of IQ samples.					

Table A.1: Classification performance of sequential screening of MSE.

followed by a softmax layer at the end of the last fully connected layer, to carry out the specific task of spectrum sensing. The hypothesis for this task was to consider signal S2 as H1 and signal Noise as H0. We experimented with three distinct signal representations: time domain representation (raw I/Q data), frequency domain representation ([FFT](#)), and time-frequency representation ([STFT](#)). Figure [A.13](#) displays the spectrum sensing performance for each signal representation type.

It is evident that the frequency representation ([FFT](#)) and time-frequency representation ([STFT](#)) exhibit similar performance. However, in scenarios with poor SNR, the [STFT](#) outperforms the [FFT](#). The [STFT](#) offers a significant advantage when a signal concentrates its energy within a specific time interval and frequency range [\[60\]](#). This characteristic of the [STFT](#) greatly contributes to spectrum sensing performance,

Test sample set sequential through each trained model (EWMSE Loss)						
SNR	STFT Nseq	Signal	Four-signal classification performance			
			Accuracy	Run 1	Run 2	Run 3
10 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
5 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
0 dB	100	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
-5 dB	300	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S3 (OOD)	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
-10 dB	600	Noise	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S1	1.0	1.0 (200/200)	1.0 (200/200)	1.0 (200/200)
		S2	0.992	0.995 (199/200)	1.0 (200/200)	0.98 (196/200)
		S3 (OOD)	0.992	0.995 (199/200)	1.0 (200/200)	0.98 (196/200)
Note	STFT Nseq means STFT window length in terms of the number of IQ samples.					

Table A.2: Classification performance of sequential screening of EWMSE.

particularly in low SNR conditions. Consequently, the STFT will serve as the data preprocessing technique for converting the raw sensed signals (raw I/Q data) into STFT plots, which will be utilized as the input data format for the neural networks in our proposed system.

Detailed Results of Voting Mechanism																
SNR	STFT Nseq	Sensing performance			AUROC	Signal	Four-signal classification performance									
		Pd	Pcd	Pmd			Pfa	OOD detection	Accuracy	Run 1	Threshold ₀₀₀	Run 2	Threshold ₀₀₀	Run 3	Threshold ₀₀₀	
10 dB	100	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.011652232	1.0 (200/200)	0.011652232	1.0 (200/200)	0.011652232	1.0 (200/200)	0.011652232
		H1	1.0	1.0	0.0	0.0	S1	1.0	1.0 (200/200)	0.00194706	1.0 (200/200)	0.00194706	1.0 (200/200)	0.00194706	1.0 (200/200)	0.00194706
							S2	1.0	1.0 (200/200)	0.002715289	1.0 (200/200)	0.002715289	1.0 (200/200)	0.002715289	1.0 (200/200)	0.002715289
5 dB	100	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.007573054	1.0 (200/200)	0.007573054	1.0 (200/200)	0.007573054	1.0 (200/200)	0.007573054
		H1	1.0	1.0	0.0	0.0	S1	1.0	1.0 (200/200)	0.001492182	1.0 (200/200)	0.001492182	1.0 (200/200)	0.001492182	1.0 (200/200)	0.001492182
							S2	1.0	1.0 (200/200)	0.001582922	1.0 (200/200)	0.001582922	1.0 (200/200)	0.001582922	1.0 (200/200)	0.001582922
0 dB	100	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.002761965	1.0 (200/200)	0.002761965	1.0 (200/200)	0.002761965	1.0 (200/200)	0.002761965
		H1	1.0	1.0	0.0	0.0	S1	1.0	1.0 (200/200)	0.001117291	1.0 (200/200)	0.001117291	1.0 (200/200)	0.001117291	1.0 (200/200)	0.001117291
							S2	1.0	1.0 (200/200)	0.001103659	1.0 (200/200)	0.001103659	1.0 (200/200)	0.001103659	1.0 (200/200)	0.001103659
-5 dB	300	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.010568072	1.0 (200/200)	0.010568072	1.0 (200/200)	0.010568072	1.0 (200/200)	0.010568072
		H1	1.0	1.0	0.0	0.0	S1	1.0	1.0 (200/200)	0.000908534	1.0 (200/200)	0.000908534	1.0 (200/200)	0.000908534	1.0 (200/200)	0.000908534
							S2	1.0	1.0 (200/200)	0.001381563	1.0 (200/200)	0.001381563	1.0 (200/200)	0.001381563	1.0 (200/200)	0.001381563
-10 dB	600	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.006131134	1.0 (200/200)	0.006131134	1.0 (200/200)	0.006131134	1.0 (200/200)	0.006131134
		H1	1.0	1.0	0.0	0.0	S1	0.99	0.99 (198/200)	0.000711949	0.99 (198/200)	0.000711949	0.99 (198/200)	0.000711949	0.99 (198/200)	0.000711949
							S2	1.0	1.0 (200/200)	0.001101605	1.0 (200/200)	0.00112532	1.0 (200/200)	0.001135233	1.0 (200/200)	0.001135233
-13 dB	1000	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.011431848	1.0 (200/200)	0.011431848	1.0 (200/200)	0.011431848	1.0 (200/200)	0.011431848
		H1	1.0	1.0	0.0	0.0	S1	0.975	0.975 (195/200)	0.000604566	0.975 (195/200)	0.000604566	0.975 (195/200)	0.000604566	0.975 (195/200)	0.000604566
							S2	0.995	0.995 (199/200)	0.00063711	0.995 (199/200)	0.000651883	0.995 (199/200)	0.000651883	0.995 (199/200)	0.000651883
-14 dB	1000	H0	1.0	1.0	0.0	0.0	Noise	1.0	1.0 (200/200)	0.038585369	1.0 (200/200)	0.038585369	1.0 (200/200)	0.038585369	1.0 (200/200)	0.038585369
		H1	1.0	1.0	0.0	0.0	S1	0.963	0.965 (193/200)	0.001179591	0.965 (191/200)	0.001178833	0.97 (194/200)	0.001198901	0.99 (198/200)	0.000962263
							S2	0.988	0.99 (198/200)	0.000962263	0.985 (197/200)	0.000960203	0.99 (198/200)	0.000962263	0.995 (191/200)	0.000962263
-15 dB	1000	H0	0.938	0.954	0.062	0.0	Noise	1.0	1.0 (200/200)	0.021615733	1.0 (200/200)	0.021615733	1.0 (200/200)	0.021615733	1.0 (200/200)	0.021615733
		H1	0.938	0.954	0.062	0.0	S1	0.658	0.655 (131/200)	0.00228544	0.66 (132/200)	0.002280104	0.66 (132/200)	0.002259641	0.66 (132/200)	0.002259641
							S2	0.542	0.555 (111/200)	0.00278735	0.56 (112/200)	0.002793627	0.51 (102/200)	0.002775062	0.51 (102/200)	0.002775062

STFT Nseq means STFT window length in terms of the number of IQ samples.

Table A.3: Detailed experiment results of voting mechanism.

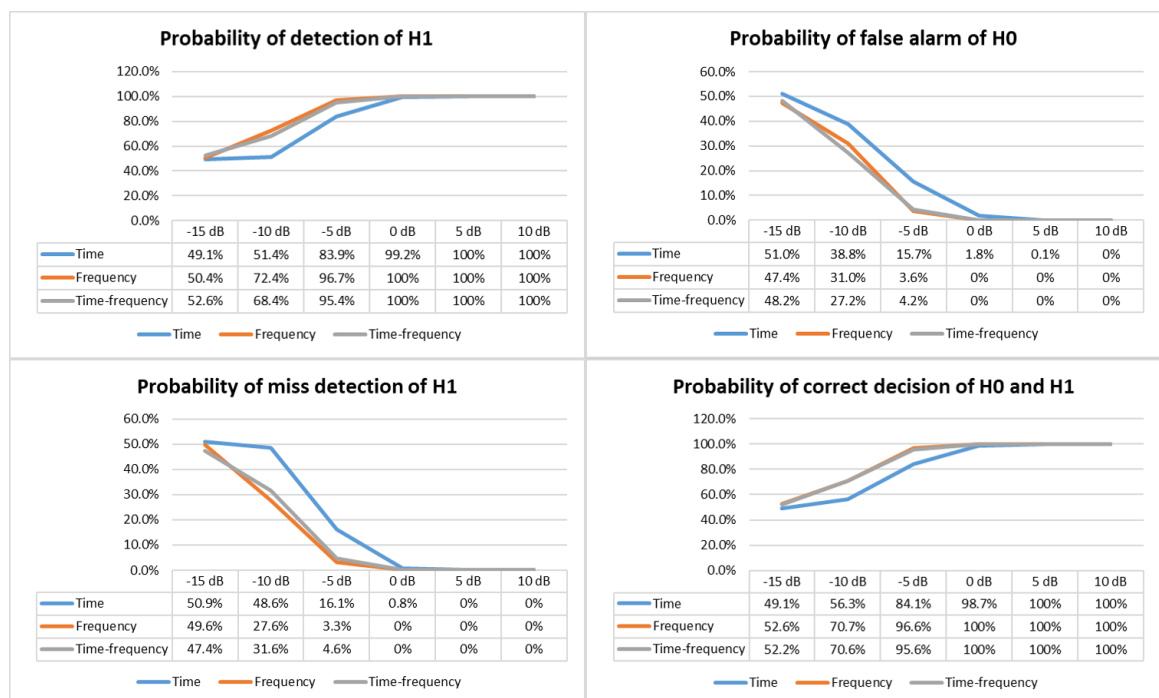


Figure A.13: Spectrum sensing performance comparison of simple fully connected neural network (sensing H1 between H0 and H1).