UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA

PROGRAMA DE DOCTORADO EN CIENCIAS

SENSADO ESPECTRAL BASADO EN LA DETECCIÓN INTELIGENTE DE ANOMALÍAS PARA REDES MÓVILES COGNOSCITIVAS B5G

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

DOCTOR EN CIENCIAS

Presenta:

CÉSAR MAURICIO PABLOS RAMÍREZ

Director de Tesis:

Dr. Ángel Gabriel Andrade Reátiga

Codirector de Tesis:

Dr. Guillermo Galaviz Yáñez

Mexicali, Baja California, México, 28 de junio, 2023.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA

DOCTOR OF SCIENCE PROGRAM

SPECTRUM SENSING BASED ON INTELLIGENT ANOMALY DETECTION FOR COGNITIVE B5G MOBILE NETWORKS

THESIS

in fulfillment of the requirements for the degree of

DOCTOR OF SCIENCE

Presents:

CÉSAR MAURICIO PABLOS RAMÍREZ

Thesis advisor:

Dr. Ángel Gabriel Andrade Reátiga

Thesis co-advisor:

Dr. Guillermo Galaviz Yáñez

Mexicali, Baja California, Mexico, June 28, 2023.

TESIS DEFENDIDA POR

César Mauricio Pablos Ramírez

Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. Ángel Gabriel Andrade Reátiga

Director del Comité

Dr. Guillermo Galaviz Yáñez

Codirector del Comité

Dr. Miguel Ángel García Andrade

Miembro del Comité

Dr. Edgar Leonel Chávez González

Miembro del Comité

Dr. Enrique René Bastidas Puga

Miembro del Comité

Dra. Wendy Flores Fuentes

Coordinadora de Posgrado e Investigación
Facultad de Ingeniería

Mexicali, Baja California, a 28 de junio del 2023.

RESUMEN de la tesis de **CÉSAR MAURICIO PABLOS RAMÍREZ**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en ELÉCTRICA con orientación en TELECOMUNICACIONES. Mexicali, Baja California, 28 de junio, 2023.

SENSADO ESPECTRAL BASADO EN LA DETECCIÓN INTELIGENTE DE ANOMALÍAS PARA REDES MÓVILES COGNOSCITIVAS B5G

Resumen aprobado por:

Dr. Ángel Gabriel Andrade Reátiga	Dr. Guillermo Galaviz Yáñez
Director de Tesis	Codirector de Tesis

El desarrollo de las comunicaciones móviles celulares ha cambiado la forma en la que nos comunicamos. Los sistemas posteriores a la quinta generación (B5G) visualizan aplicaciones tales como la conectividad vehicular, la realidad virtual y aumentada, las comunicaciones holográficas o el internet de los drones demandarán mayores tasas de transferencia de datos y espectro radioeléctrico. Sin embargo, la política de asignación estática del espectro muestra signos de llegar a sus límites. La compartición de espectro mediante la Radio Cognitiva (CR) es una alternativa para mejorar la eficiencia del uso del espectro radioeléctrico. La CR requiere de un proceso de percepción de espectro que identifique oportunamente bandas de frecuencia desocupadas. Existen diversos esquemas de detección de espectro que no siempre son capaces de resolver el problema de ocupación de espectro. De ahí que técnicas de inteligencia artificial (IA) se utilicen para modelar el problema de detección de espectro como un problema de clasificación. La teoría de detección de anomalías, rama de la IA, se utiliza para identificar valores atípicos en un conjunto de datos.

Esta tesis se fundamenta en que es suficiente entrenar un modelo de detección de espectro con un solo tipo de señal (una sola clase), en este caso se utilizaron solo señales de ruido que representan un canal libre y cualquier señal modulada presente en el canal se considera como una anomalía. Se utilizó el método de reconstrucción para modelar el detector de espectro, mediante un Autoencoder Convolucional (CAE) y, una memoria Larga de corto plazo (Long-Short Term Memory - LSTM) para detectar las desviaciones de los patrones de entrada. La ventaja de utilizar esta estrategia es que no se requiere recolectar una vasta cantidad de datos para entrenar el modelo, ni generar un modelo de detección de espectro complejo que requiere tiempos de entrenamiento excesivos. Los resultados de detección obtenidos con la teoría de detección anomalías superan a detectores convencionales como el detector de energía y otros detectores basados en IA como clasificadores de señales.

ABSTRACT of the thesis presented by **CÉSAR MAURICIO PABLOS RAMÍREZ**, in partial fulfillment of the requirements of the degree of DOCTOR OF SCIENCE in ELECTRICAL ENGINEERING with orientation in TELECOMMUNICATIONS. Mexicali, Baja California, June 28th, 2023.

SPECTRUM SENSING BASED ON INTELLIGENT ANOMALY DETECTION FOR COGNITIVE B5G MOBILE NETWORKS

Beyond 5G (B5G) mobile networks represent the next step in evolving wireless communication to transcend the limitations of current connectivity standards and improve speed, user capacity, reliability, and latency. Cognitive radio technology offers a promising solution to effectively utilize the available spectrum and mitigate the scarcity challenges its static administration presents. This can be achieved by enabling the coexistence and integration of diverse communication technologies and applications within the B5G ecosystem through intelligent dynamic spectrum access.

Spectrum sensing plays a critical role in enabling Dynamic Spectrum Access (DSA) for cognitive networks, allowing the network to identify available spectrum resources and determine user-free channels for usage. Therefore, accuracy in spectrum sensing is crucial to minimize potential interference that may arise from DSA operations. Numerous spectrum sensing strategies have been proposed in the literature due to the challenges associated with accurately detecting spectrum availability. With the recent advancements in computational intelligence, we see an increasing inclusion of Machine Learning (ML) algorithms for spectrum sensing. ML has been utilized to improve the performance of conventional sensing strategies or approach the detection problem as a classification problem by building an ML model out of user signals and vacant channels consisting of Average White Gaussian Noise (AWGN) examples. However, the variety of modulation schemes increases the complexity of this training process, resulting in models for specific groups of signals, which could make user detection difficult in heterogeneous environments.

This thesis explores the proposal of approaching spectrum detection as an outlier detection problem. Anomaly detection models are trained with one-class data, which is the basis for learning normal behavior. These models then classify anything that deviates from what was learned from the training examples as an outlier or anomaly. In user detection, an outlier detection model is trained using vacant channels (noise) and deems vacant channels as anomalies. Outlier detection was approached by utilizing Convolutional Autoencoder (CAE) and Long-Short Term Memory (LSTM) models. Both models showed an improved detection rate over conventional energy-based spectrum sensing.

Agradecimientos

Doy gracias a mis directores de tesis, el Dr. Ángel y el Dr. Guillermo por haber sido mi guía y mentores en esta larga jornada, desde mi llegada a la maestría hasta alcanzar mi grado de Doctor en Ciencias, pero sobretodo como amigos.

A mi familia que siempre hizo del hogar un refugio donde pude perdurar en los tiempos más difíciles y me apoyaron para que pudiera alcanzar mis metas. Sin nunca olvidar a mi padre, a quién siempre tuve presente.

A mi novia y mejor amiga Johana, una persona integral en mi vida con la que puedo siempre acudir para celebrar las buenas noticias y quien también siempre esta ahí para escucharme y darme ánimos en las malas.

A la Universidad Autónoma de Baja California (UABC) por todo lo aprendido. Me llevo, además, una grata experiencia.

Quiero agradecer también al ahora Consejo Nacional de Humaninades, Ciencias y Tecnologías (CONAHCYT) por el apoyo económico que hizo posible este logro y permitirme enfocar mi esfuerzo en concluir esta etapa de mi desarrollo académico.

List of Content

Page

List of Content (continued)

List of Figures

List of Figures (continued)

List of Figures (continued)

List of Tables

Chapter 1

Introduction

1.1 Background

The rapid growth of wireless communications and its pervasive use in all walks of life are changing the way we communicate in some fundamental ways. Today, wireless devices are everywhere, and cellular telephones are commonplace. Mobile phones are used to stream YouTube videos, watch TV shows, download and store songs and movies, take pictures of everything around us, play video games, and surf the Web. Future beyond fifth-generation (B5G) mobile systems visualize applications such as vehicular connectivity, virtual and augmented reality, holographic communications, or the internet of drones, to name a few, which will demand higher data transfer rates and consume a greater amount of radio bandwidth [1]. Regulators cannot expect to reduce demand, nor can they expand the overall radio spectrum supply. Therefore, exploring supplementary ultra-high-frequency bands or revising the utilization of currently available bands is essential.

The radio spectrum is a limited resource that enables data transmission between devices. A static spectrum allocation policy avoids potential interference where services and technologies are assigned a specific range of frequencies. Although this strategy permits their coexistence within the spectrum, it needs to be more flexible with the increase of demand over time due to new users utilizing the spectrum and the increased data consumption per user enabled by technology advancements and higher definition

media consumption. According to [2], there will be a growth of 41% in mobile data traffic and 237 exabytes (EB) per month by 2026 as new emerging technologies demand high-speed connectivity and low latency, as such, wireless networks need the capability to serve the exponentially growing traffic demand. Finding ways to use less radio bandwidth to carry out these communications is possible, but it does require adopting new technologies. Also, changes to current radio spectrum regulation policy are required for that to happen, which is still largely rooted in the analog paradigms of the last century. Regulator bodies allocate different frequency bands to different wireless applications, such as broadcasting, aeronautical communication, radar, and mobile communication, to name just a few. None of them can use the other's frequency band. This frequency band's dedication to specific purposes has been the convention of radio spectrum management to control harmful interference [3]. However, this traditional radio spectrum allocation shows signs of reaching its limits. Therefore, there are a few options to avoid the bandwidth crisis.

1.2 Dynamic Spectrum Access

An apparent spectrum scarcity may already occur for any service or technology when its assigned frequency range becomes saturated with high traffic. However, this is due to the inability of such services to use the spectrum fully. At the same time, spectrum utilization for some frequency ranges is only partial (also known as spectrum sub-utilization). This sub-utilization gives place to a need to make more efficient use of the spectrum, as mentioned in [4], [5], several models following the concept of Dynamic Spectrum Access (DSA), which aim to allow a more flexible spectrum usage are

proposed in the literature. For example, the Exclusive Use model is proposed with two approaches: spectrum property rights and dynamic spectrum allocation, where the former allows licensees to sell and trade spectrum freely. In contrast, the second approach aims to assign the radio spectrum according to services' spatial and temporal traffic statistics. The Open Sharing model, or spectrum commons, indicates an operating model where nobody can claim exclusive use of a shared resource. This model is supported by the success of wireless services operating in the unlicensed industrial, scientific, and medical (ISM) radio band. The Hierarchical Access model uses any band outside a fixed frequency range, thus allowing secondary users to coexist with primary users while minimizing harmful interference to their services. Hierarchical DSA can be approached in an underlay or overlay model, where the underlay model limits the secondary user's transmission at low power to be unperceived by the primary user. In contrast, the overlay model allows access to unoccupied primary user channels opportunistically.

Out of the previously mentioned DSA approaches, hierarchical access is considered the most compatible with the current spectrum management policies since it requires no changes. Although sufficient for small personal area networks, the transmission limitations of the underlay model make it less flexible in terms of network configurations compared to the overlay model. However, allowing the coexistence of both primary and secondary users calls for intelligent technology, such as Cognitive Radio to avoid any potential interference this approach may cause.

Spectrum sharing and DSA has been considered alternatives to increasing spectrum efficiency and, thus, the capacity of wireless networks [4],[6],[7],[8]. Spectrum sharing allows multiple classes of users to share the same frequency bands reliably, i.e., without interrupting the transmission of any user or causing harmful interference. Several

spectrum measurements in frequencies up to 6 GHz indicate that the utilization rate of users who currently maintain an exclusive license to use a spectrum band is 10–20 percent, which is relatively inefficient. Thus, much prime spectrum remains unused in the spatial and temporal domains, and there is growing pressure on commercial mobile bands, especially in dense urban areas. This is the primary motivation for the development of novel spectrum-sharing methods.

1.3 Cognitive Radio and Spectrum sensing

The DSA strategy is considered an approach in which radio spectrum is allocated where and when it is available and for the time it is needed. Cognitive Radio (CR) is defined in [9] as an intelligent wireless communication system, aware of its surrounding environment, capable of learning from it and adapting its operating parameters in real-time to deliver a highly reliable communication whenever and wherever needed and efficient utilization of the radio spectrum. Cognitive Radio is the DSA-enabling technology. It permits secondary users (SU), also called cognitive users, to sense the spectrum usage of surrounding primary users (PU) and whether any unused spectrum (spectral holes) exists. SU can then reconfigure its software and hardware parameters to utilize the idle spectrum in a way that does not interfere with PU. The core idea behind CR is to facilitate dynamic and flexible access to the idle spectrum of PU.

From this definition, the tasks of a CR network involve observing the surrounding spectral environment to find unoccupied channels, learn from them, make decisions based on user necessities, and finally adjust its transmission parameters to establish a

desired communication quality. Since CR is constantly observed, these tasks can be defined as a Cognitive Cycle. 1.

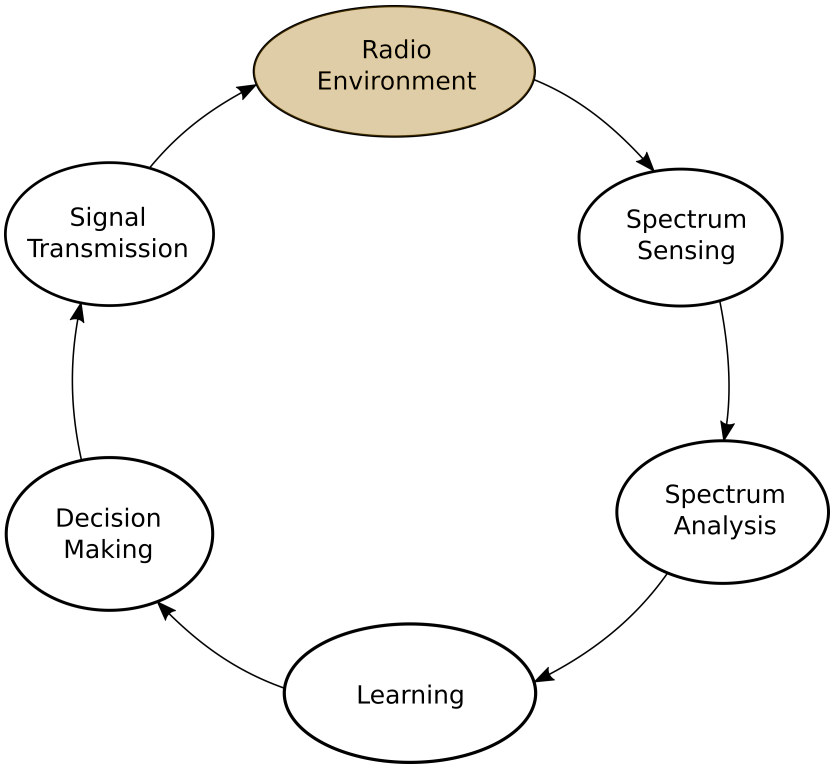


Figure 1: **Cognitive Cycle.**

The importance of a precise spectrum observation is emphasized due to the need to avoid any potential user interference that may occur through the overlay DSA model. The primary users have priority over their licensed channels. This priority means the secondary users must avoid occupied channels and vacate any channels they use when the primary user requires them. Otherwise, this would harm any ongoing communication. Therefore, spectrum detection plays an essential role in a CR system, and achieving a reliable detector is the focus of many studies within the field.

According to the literature, spectrum sensing algorithms fall into two broad categories, knowledge-aided and blind. Knowledge-aided algorithms require a priori knowl-

edge of the features within the signal they aim to detect. Some of the most studied include cyclostationary detection thanks to its high probability of detection on low Signal-to-Noise-Ratio (SNR) conditions [10], as well as matched filter detection [11].

However, the diversity of signals usually found in a public environment makes blind detection a desirable feature, as they require no previous knowledge of the user signal and usually base their detection criteria based on the noise contained within the vacant channels instead. This type of sensing usually comes at the cost of lower user detection rates or higher processing complexity, which delays the detection process. Some examples of blind sensing include eigenvalue detection, which manages to reach higher detection rates under highly adverse conditions [12]. However, this method requires a higher computational complexity [13], [14].

Energy detection is one of the most widely studied detection algorithms [15] thanks to its ease of implementation and low computational complexity. However, its performance is more easily degraded in a practical environment. Energy detection owes its low complexity to only needing the energy found in the sensing channel to decide by comparing it to a threshold defined from the perceived noise variance. On the other hand, energy is a property easily influenced by noise. Ever since its early research in [16], numerous methods are presented in the literature which aims to improve the detection performance of energy detection, such as incorporating an adaptive threshold [17], giving the ability to adjust its threshold in response to noise fluctuations. However, these methods enhance the detection rate of the energy detector; they also add more complexity to the detection process.

Cooperative sensing is another explored solution to enhance the detection rate of the energy detector, as well as the previously mentioned methods. Cooperative sensing involves two or more cognitive users performing channel analysis from different geolog-

ical locations. This analysis either consists of the occupancy state of the channel (hard data) or a measurement of a channel feature such as energy (soft data). This data is then forwarded through a reporting channel to a fusion center or in a decentralized manner to determine the state of the channel. However, a noisy or faded report channel presents an additional obstacle to reaching a reliable spectrum decision [18].

With the advances in Computational Intelligence and the now more accessible tools to build learning-based algorithms, we see an inclusion of such technology benefits in the field of SS research, where various methods, from Machine Learning (ML) to Bio-inspired algorithms, have been evaluated. ML is a subset of Artificial Intelligence that analyzes and interprets patterns and structures in data enabling learning, reasoning, and decision making, and has received increased interest for research in the context of cognitive radio networks [19].

ML models have been implemented to improve the conventional SS methods. In [20], Support Vector Machines (SVM) are utilized to reduce cooperative SS overhead and improve detection performance. Random forests, another ML algorithm, are implemented in [21] for user signal recognition in low SNR. Channel occupancy prediction is implemented in [22] [23] to minimize the time spent on channel sensing and increase occupancy decisions' reliability.

Neural Networks (NN), a subset of ML, has been applied in the context of SS for user activity prediction within a band to identify potentially unoccupied channels [24]. In [25], NN separates the signal components from a noise signal at the receiver to estimate the signal attributes required for conventional energy detection. NN is also used for Automatic Modulation Classification (AMC) as seen in [26].

1.4 Problem Statement

An ML algorithm creates a mathematical model out of examples through an iterative process called training, where with each iteration, an evaluation is performed to measure the error of the model performance. In response, the model parameters are adjusted as the ML algorithm learns from the examples (training data) as it attempts to minimize the error through a learning algorithm.

Training can be done in a supervised or unsupervised manner. The first consists of labeling the training data, where the algorithm learns to create a model that relates the data to their respective label. Unsupervised training uses unlabeled data, where the algorithm extracts the most predominant features from data to identify patterns without human intervention.

Spectrum occupancy can be modeled as a classification problem, where Class 0 may represent a vacant channel and Class 1 a user signal. However, due to the variety of possible user signals in terms of modulation types, this is a case where many possible modes are present. Some may not be known a priori, making the conventional multi-class classification schemes unsuitable [27]. In such cases, Anomaly Detection (AD) may be the solution.

AD is defined in [28] as the problem of finding patterns in data that do not conform to expected behavior. AD learns to recognize only the patterns found within the training data that represent the indicative of normal. At the same time, everything that differs from it is therefore deemed an anomaly according to an abnormality or novelty score. AD is sometimes referred to as novelty or outlier detection.

Detection of anomalous signals has been approached in the literature for various applications, including the medical field [29],[30],[31], structural damage detection [32],

spectrum sensing applications for intrusion detection [33],[34],[35] among others [27]. The performance of some of the implemented AD models is measured in terms of accuracy, such as [36],[37],[38]. This evaluation and optimization are usually practiced in ML algorithms, measuring the rate at which the model makes the correct assessments.

On the other hand, spectrum sensing traditionally utilizes a different approach by measuring the detection rate (P_D) and false alarm rate P_{FA}, which can be interpreted as wrongly determining signal detection when the channel is vacant. At the same time, accuracy gives a summarized evaluation of the model, considering jointly P_D and $1 - P_{FA}$.

The anomaly detection technique is based on unsupervised learning, so it does not require extracting the signal's characteristics to be identified. Still, anomalies are detected from training raw (or unprocessed) data from one or more multiple classes. The rationale of this thesis is that it is enough to train the spectrum detection model with a single type of signal (a single class). We will use only noise signals representing a free channel in this case. Any modulated signal (those transmitting information) present in the channel will be considered an anomaly. After extensive research of the literature, it was found that channel occupancy decision has yet to be approached as an AD problem. Moreover, detection optimization in terms of accuracy would minimize the detection error, offering a more balanced trade-off between interference prevention through P_D and vacant channel utilization through $1 - P_{FA}$.

Provided the reviewed literature, the following research question is formulated: **How can anomaly detection techniques be utilized in cognitive radio systems to enhance the identification of spectrum use opportunities, resulting in improved vacant channel detection performance?**

As such, in this thesis, vacant channel detection is being modeled as an AD problem

as a contribution to the field, first using the more traditional CFAR approach, where a comparison is made with the energy detector regarding detection rate under adverse conditions in terms of SNR. The implemented model is optimized and evaluated in terms of accuracy.

Using anomaly detection techniques in cognitive radio systems will improve detection performance by enhancing the identification of spectrum use opportunities. Specifically, we hypothesize that:

- H1: Cognitive radio systems incorporating anomaly detection techniques will achieve a higher probability of correctly detecting spectrum use opportunities than systems relying solely on traditional spectrum sensing methods such as the energy detector.

- H2: Integrating anomaly detection techniques in cognitive radio systems will reduce the probability of false alarms in identifying spectrum use opportunities, leading to more accurate detection performance.

1.5 Thesis objectives

1.5.1 General Objective

Develop and optimize in terms of accuracy a spectrum sensing method based on the identification of Additive White Gaussian noise using anomaly detection for its application in cognitive radio systems.

1.5.2 Specific Objectives

- Investigate and select suitable anomaly detection algorithms for identifying Additive White Gaussian noise in the spectrum.

  Conduct a comprehensive literature review to identify state-of-the-art anomaly detection techniques applicable to spectrum sensing. Select the most suitable anomaly detection algorithm(s) for further development and optimization.

- Develop and optimize a spectrum sensing method based on the selected anomaly detection algorithm(s) for identifying Additive White Gaussian noise.

  Conduct simulations and experiments to evaluate the performance of the developed method and iteratively refine it for enhanced accuracy.

- Validate the effectiveness and accuracy of the optimized spectrum sensing method in cognitive radio systems.

  Evaluate the performance of the optimized spectrum sensing method using real-world cognitive radio datasets. The developed method's accuracy, reliability, and robustness in identifying Additive White Gaussian noise in various wireless environments is evaluated.

1.6 Main contributions

- The development of the selected anomaly detection algorithm(s) in the cognitive radio system framework.

- A framework to fine-tune and optimize the parameters of the anomaly detection method to improve its accuracy in identifying Additive White Gaussian noise.

- A comparison of the performance of the developed method with existing spectrum sensing techniques, including traditional methods and other anomaly detection-based approaches.

1.7 Thesis outline

Various research works that propose solutions to the spectrum sensing problem by applying artificial intelligence techniques are described in Chapter 2. Chapter 3 describes the general spectrum sensing model and how it is formulated as an anomaly detection problem. In Chapter 4, the generic architecture and functioning of a neural network (NN), as well as the training process, is explained. Also, it explains the architecture of Convolutional Autoencoder (CAE) and the Long-Short Term Memory (LSTM) Recurrent Neural Network model based on entropy features, the convolution process, and its integration into the encoding and decoding process. The utilized hardware and the dataset generation procedure for training and evaluating the Anomaly detection-based spectrum sensing are presented in Chapter 5. The threshold selection process for CFAR and accuracy optimization are also explained. Chapter 6 describes the evaluation of the spectrum sensing capabilities of the CAE and LSTM algorithms. The CAE and LSTM models are optimized for accurately detecting modulated-plus-noise signals under varying SNR conditions. The conclusion and future research work are presented in Chapter 7.

Chapter 2

Literature review

2.1 Introduction

Every wireless application works best within a specific range of frequencies. Radio waves above 3500 MHz do not penetrate buildings well, leading to frustrating dead spots. Lower frequencies are better in this regard but require huge antennas for efficient transmission; 300 MHz is the lowest frequency supported by a reasonably efficient antenna that is small enough to fit on a portable device. The so-called millimeter-wave (mmW) bands, ranging from 30 to 300 GHz, can provide greater bandwidths than the current cellular allocations constrained to the sub-3 GHz band. Due to the smaller wavelengths of mmW signals, greater antennas can be established in the same dimensions, leading to very high gains. However, despite these encouraging developments, challenges in using mmW bands still exist that limit their large-scale implementation. For example, due to the directional behavior of mmWs, specific changes have to be applied to current cellular systems; mmW bands are susceptible to shadowing and can be readily affected by multiple objects, and due to the short coherence time of mmW bands, channels sustain rapid fluctuations, leading to intermittent connectivity and necessitating adaptable communication [39]. In addition to the challenges presented by using such high frequencies, it would not be possible for portable devices to have the appropriate technology to operate at these frequency bands in the short term.

Research on Cognitive Radio focuses on detecting temporarily unoccupied spectrum (spectral holes). A temporarily unoccupied spectrum is understood as that part of the spectrum in a band designated for use by one or several applications that are not used at a given time and in a given geographical area. Spectrum sensing allows a mobile device to identify available frequency bands opportunely, then vacate the frequency band in case of that the conditions require it and immediately look for another frequency band to continue with its data transmission. Consequently, next-generation wireless communications must employ spectrum-sharing methods and accommodate intelligently collaborative radios to maximize spectral efficiency and avoid interference with other incumbents to reap the full potential of licensed and unlicensed frequency bands.

Today, there is a symbiosis between Artificial Intelligence (AI) and radio communications to augment spectrum-sharing methods and benefit the use of the radio spectrum [40]. The AI techniques can bring new solutions and opportunities, such as signal recognition, real-time monitoring of multiple equipment and self-signaling devices, and/or interference source identification. Spectrum sensing techniques that rely on artificial intelligence are still in an early stage. This chapter reviews various research works that propose solutions to the spectrum sensing problem by applying artificial intelligence techniques, such as artificial neural networks, deep learning, and Long short deep neural networks.

2.2 Conventional (non-intelligent) Spectrum Sensing techniques

Sharing the spectrum arises because it has been observed that most of the radio spectrum could be used more efficiently. For example, Cellular Network bands are overloaded in most parts of the world, but other frequency bands, such as military, amateur radio service, or localization, are often unused. The fixed spectrum allocation means that frequencies that are practically in disuse, assigned to specific services, cannot be used by users not authorized for that use, even if their transmission would not introduce any interference in this previously unoccupied service.

The idea of Cognitive Radio arose in 1999. Its creator Joseph Mitola conceived it as the technology capable of providing a wireless device with sufficient intelligence to understand the telecommunication needs of a mobile user and modify its operating parameters autonomously. Cognitive Radio transforms blind wireless nodes (protocol executors only) into intelligent agents, aware of their environment and reconfigurable of their transmission and/or reception parameters to be able to use the services that the user needs, even if that user does not know how to get them.

In spectrum sharing, cognitive radios must determine if a signal from a transmitter is locally present in a specific frequency band. Detecting unused spectrum and sharing it without harmful interference to other users is an important requirement of cognitive spectrum-sharing technology. Energy detection (ED) [16],[41], cyclostationary detection [10],[42], and matched filter detector [11],[43] are the most-studied spectrum sensing techniques.

The energy detector measures the energy in the frequency band that the cognitive device is interested in occupying. In ED, the binary hypotheses (H_0 and H_1) are considered for identifying the presence of a signal on the channel. Hypothesis H_0 says the channel is available, while H_1 says the channel is occupied. Hypothesis H_1 is true when the energy of received signals over N samples is computed and greater than or equal to the predefined threshold value. The selection of a threshold affects the spectrum sensing performance of cognitive users.

If the energy is strong enough, the detector will realize that that frequency band is occupied and will not attempt to use it. If the portable device decides to transmit its data, ignoring what the energy detector perceived will interfere with that user's transmission. On the other hand, if the energy in the frequency band is very low, almost imperceptible, no one is transmitting, and the frequency band is tagged as available. At that time, the mobile device will occupy the frequency band to transmit its data. The limitation that exists with this energy detection is that it is not very sensitive and is easily confused when the signal-to-interference noise ratio (SINR) is low, either because the signals transmitted in that band are affected by the interference of other signals or because they attenuate under multipath fading, shadowing, and non-line-of-sight communication.

The Matched filter and Cyclostationary spectrum sensing techniques are more accurate than energy detection. The Matched Filter technique uses a linear filter to optimize the output SINR for a given input signal. It is the foremost method for detecting the presence of a user if the transmitted signal is previously known. It uses the user's previous information and the agreed sense signal to evaluate the presence of a user signal

and improve the performance of the accuracy of the SINR ratio. This method optimizes detection in stationary Gaussian noise by reducing time spent on spectrum sensing and maximizing accuracy and efficiency. The drawbacks of this approach are that a CR receiver would require a dedicated receiver for each transmitter and previous information from the signal user, which will result in poor performance if the information is inaccurate [44]. The Cyclostationary spectrum sensing technique outperforms energy detection in low SINR regions. Therefore, it is more resilient to noise uncertainties. This technique divides the signal into normal recurrence or in-building periodicity and auto-stationary detection techniques. This method may tell the difference between CR transmissions and different forms of user signals. It is less likely to be mistaken about the occupancy of the frequency band. Still, the challenges of this technique are the computational difficulty and the need for extended sensing time.

Entropy-based spectrum sensing technique is proposed as an option to tackle the noise uncertainty negative effects [45],[46],[47]. Its implementation complexity is comparable to ED since it does not require any prior knowledge of the signal waveform to detect. According to information theory, entropy quantifies an event's predictability (or randomness). The less predictable an event is, the higher its entropy. Therefore, the entropy of a random variable depends on its statistical distribution. For any random variable, the maximum possible entropy occurs when it shows a uniform distribution (i.e., all its values have equal probability). Entropy, therefore, can be used as a metric to determine the presence of communication signals in a channel, considering the expected statistical behavior of these signals and noise.

Energy and entropy are unsuitable features to distinguish signal from noise for the

correlated signals case. To address the problem that Energy and Entropy detectors have degraded detection performance in the case of correlated signals, some detectors were designed by exploiting correlation features, such as covariance [48], [49]. Considering that the eigenvalues of the covariance matrix can capture the correlation between the received signals well, some detectors based on eigenvalues were proposed [50]. Generally, the detectors based on eigenvalues are divided into two categories according to the test statistics generated by all eigenvalues or extreme eigenvalues (maximum and minimum eigenvalues). However, the existing detectors resorting to extreme eigenvalues show different detection performances. Compared with the detectors using all eigenvalues, the detectors based on the extreme eigenvalues have degraded detection performance. The performance of the existing detectors based on extreme eigenvalues needs to be further improved significantly in the low SNR scenarios.

For Cognitive Radio to be the technology that will make the use of the spectrum more efficient, it will need reliable signal detection, that is to say, that they do not make mistakes when they decide on the existence of a signal in the frequency band and that they carry out this decision as quickly as possible. The fundamental problem of detector design is choosing the favorable features that can capture signal and noise characteristics well and constructing the test statistics for decision-making. Model-driven (or model-based) approach has played a dominant role in the design of test statistics for many years. When the model is sufficiently accurate, the detector derived from the model-driven approach is generally expected to be optimal. However, the model-driven method is limited when the model is unavailable or inaccurate in complicated scenarios. On the contrary, the data-driven approach, independent of the powerful models, is an effective tool for data analysis and feature extraction, providing a new detector design

method.

2.3 Spectrum Sensing techniques based on Intelligence Computational

Recently, the Spectrum Sensing technique has been modeled as a classification problem, where Machine Learning (ML), especially deep learning (DL) algorithms, are applied to identify potentially unoccupied channels.

The proposed approach presented in [51] is composed of five Classification Modules (CM) based on subspace decomposition and Radial Basis Function Neural Networks (RBFNN) that decide about the presence of a user signal and identify the signal type, which is not possible through Maximum-minimum Eigenvalue Detection (MME) and Energy with Minimum Eigenvalue (MIN) [52]. User detection and identification are reduced to a simple Finite Impulse Response (FIR) filtering and a Neural Network (NN) classification procedure. The classifiers are trained with three partitions: WS_{S0}, WS_{SN}, WS_{AWGN}, considering three real-world operations SNR conditions to characterize user presence and absence, where WS_{s0} is obtained from the original unfiltered signal, WS_{SN} represents the user presence in a quite clear channel and WS_{AWGN} is composed of original modulated signal samples degraded by AWGN with a chosen SNR less than -5dB. However, this proposal can only classify five different types of primary signals. Therefore, if during the system operation, a new PU signal type may enter the Wireless Regional Area Network (WRAN) coverage area at any time, the trained model would

not be able to classify the user signal since the CM for the analysis of this new signal type is now within the set of CMs. Thus, in this case, the authors apply energy detection as a contingency method for detecting the presence of user signals which are still not recognized. However, the ED is not always able to discriminate modulated signals from a high level of noise or a high level of interference. In such an operation condition, the ED may indicate false detections.

The method proposed in [53] consists of a binary classification with signal and noise categories. They approach this by training a Convolutional Neural Network (ConvNet) in a supervised manner, where the signal class consists of as many types of wireless signals as possible, hoping that the trained network model can also be generalized to detect other untrained signal types. In contrast, the noise class training samples consist of pure noise data of the same size. The channel state decision is based on the confidence of the model's output where a threshold is set according to a desired P_{FA}. Additionally, transfer learning is utilized to increase its performance to real-world signals.

In [54], the spectrum sensing problem is transformed into an image recognition problem, and machine learning is employed to distinguish between noise and user signal presence by repurposing an existing ConvNet, AlexNet, to sense the spectrum for energy. The implementation also consisted of a binary classification approach for the presence and absence of user signals. The model was trained with 227 x 227 spectrograms representing signal plus noise for the user presence class and noise only for the user absence class. The spectrogram is calculated from signal samples using complex-values samples and is given to the ConvNet detector for classification. The spectrogram is then converted to decibels (dB). The dB values are then normalized by the largest dB value and then are scaled by 255, yielding a spectrogram image with pixel densities ranging from 0 to 255.

In [55], the authors propose a Deep Learning-based spectrum sensing system to recognize modulated signals. Unlike current Deep Learning-based spectrum sensing using expert features, this proposal uses raw signals as inputs to a Deep Neural Network. Although this deep sensing is effective when operating in the same scenario as the collected training data, the sensing performance fails when wireless conditions change. The authors incorporate transfer learning into the framework to adapt the learned models to new communications settings for improving robustness. The results are degraded when unsupervised data are used. Therefore, this proposal needs a small amount of labeled data. Therefore, the SU would require real labeled data in its actual environment.

In [56], the authors investigate using Deep Learning (DL) in modulation classification, a significant task in many communications systems. The DL does not require manual feature selections, significantly reducing the task complexity in modulation classification. This paper uses two convolutional neural network (CNN)-based DL models, AlexNet and GoogLeNet. Specifically, several methods are developed to represent modulated signals in data formats with gridlike-topologies for the CNN. The impacts of representation on classification performance are also analyzed. In addition, comparisons with traditional cumulant and ML-based algorithms are presented. Experimental results demonstrate the significant performance advantage and application feasibility of the DL-based approach for modulation classification.

A blind spectrum sensing method based on deep learning is proposed in [57]. The main idea of this work is well-handle a situation in which the prior information of the primary signal is lacking. This work introduces a deep learning method into the

spectrum sensing field to obtain better signal detection performance even if the SNR is low and prior information on the signal of interest is lacking. This paper applies the deep learning method to the design process of test statistics. It seeks to use the proven excellent feature extraction ability to complete the design of the test statistics. The deep neural network proposed in this paper for spectrum sensing consists of three parts: one-dimensional (1D) convolutional neural networks (1D CNNs), long short-term memory (LSTM), and fully connected neural networks (FCNNs). While in [58], authors developed a Spectrum Sensing framework based on CNN and LSTM models, where CNN extracts energy correlation features from the sensing data. Then these series of energy correlation features corresponding to multiple sensing periods are fed to the LSTM network to extract temporal features for learning the activity pattern of PU.

Due to the random nature of fading channels between the cognitive users (SU) and the licensed users (PU), detecting the spectrum for a single node SU is susceptible to error. To overcome this challenge, Cooperative Spectrum Sensing (CSS), where many SUs do the individual SS and contribute to making final decisions, has been used to improve the spectrum sensing performance of the system. However, designing the generic CSS strategy for different channel conditions is complicated as the scenarios for the SUs close to PU differ from those far away from PU. Recently, various deep/machine learning (DL) algorithms have been utilized to improve the performance of CSS. For example, in [59], the authors developed an optimization framework based on neural graph networks and reinforcement learning to solve the energy efficiency problem in a cooperative spectrum sensing network. As transmitted, signals can vary in several ways (e.g., alphabet sizes, coding schemes), and signal propagation depends on many factors (e.g., frequency, terrain profile); getting enough ground-truth labeled training

data across all possible scenarios is difficult. Large quantities of human-labeled data should always be readily available for learning a functional mapping between the training samples observed and the desired output. The advantage of supervised learning is that both the convergence speed and the action quality are high. However, they typically require a large amount of data to be labeled manually, thus making the data processing more complex. To facilitate mobile users to choose transmission parameters individually and intelligently, reinforcement learning is employed, a model-free method to achieve good system performance without relying on accurately labeled data.

The proposal in [60] uses sensing samples to train a Deep Cooperative spectrum sensing (DCS) based on Deep Neural Network (DNN) model. DCS can operate regardless of the sensing decisions at individual SUs, i.e., hard decision (HD) and soft decision (SD), and the spatial and spectral correlations of the channels are taken into account without needing to derive explicit mathematical formulations. The SUs perform individual spectrum sensing based on energy detection (ED). The signaling overhead in cooperative spectrum sensing is increased. Also, a fusion center of the Cognitive radio network (CRN) is necessary to determine the presence of the PU by combining the individual sensing outcomes. The proposal needs individual sensing outcomes from nearby SUs and adjacent bands to determine the presence of the PU. When the locations of SUs are not known such that the index of SUs is arbitrarily chosen, the benefit of using CNN can be diminished. In this case, the multiple CNN models with different permutations of SU index are trained simultaneously, and the index of SUs which achieves the highest accuracy can be chosen. However, the training of DCS requires a huge overhead.

A hierarchical cooperative long short-term memory (LSTM) network-based coop-

erative spectrum sensing (CSS) method, which utilizes a convolutional neural network (CNN) and LSTM network, is proposed in [59]. The authors considered a multi-antenna multi-SU scenario and developed a hierarchical cooperative LSTM-based CSS method that uses SU LSTM (SU-LSTM) to extract temporal features from the sensing data of individual SU over multiple sensing periods. Further, the cooperative LSTM models the SU-level temporal features of all cooperative SUs to learn the group-level PU activity pattern. The results show that the proposed hierarchical cooperative LSTM method provides improved detection performance compared to the conventional methods in terms of probability of detection, even in an imperfect reporting channel.

In [61], the authors propose a new cooperative spectrum-sharing scheme for the Cognitive Industrial Internet of Things (CIIoT) platform. They designed a learning game model using multi-agent reinforcement learning (MARL) and the negotiated aspirations bargaining solution (NABS). Individual CIIoT devices adaptively select their cooperative sensing policy in the learning mechanism according to the MARL model. The available spectrum resource is dynamically shared in the bargaining mechanism through the NABS, which is obtained based on the devices' selected sensing policy. Also, they apply the non-orthogonal multiple access (NOMA) as the control access method. The simulation results show that the learning game-based control protocol performs well regarding the normalized device payoff, CIIoT system throughput, and fairness.

The above-reported results show that using training data, including modulated and noise signals, for PU presence detection is better than training the model with features focused on a single-feature state. Spectrum awareness performs better if trained to detect a specific type of user signal modulation and loses reliability when achieving generalized signal detection. Modeling a generalized signal detector with ML requires

collecting all possible user data signals to identify PU presence with a binary classification model. Given the different signal types wireless communication systems use, the ML classifier training process under such an approach will need a large amount of data (labeled manually) to be trained. In CR networks, spectrum opportunity varies with time; therefore, building up a huge set of labeled data that includes all possible signal propagation scenarios is difficult. Once the synthetic and real data are trained in the classifier, it can perform online spectrum sensing. However, the spectrum environment may change after a while, causing the ML spectrum sensing model may not perform well. A supervised ML model relates the training dataset to the desired output. Thus, it is not explicitly programmed, while an unsupervised ML extracts the most predominant features from a training dataset to identify patterns used for classification.

In this sense, this research project is proposed to apply the anomaly detection (AD) theory [62], [63], [64] to perform spectrum detection. Anomaly detection theory is a branch of AI used to identify outliers in data or recognize new patterns with unusual behavior of time series of a given metric. This technique is based on unsupervised learning, so it does not require extracting the signal's characteristics to be identified. Still, anomalies are detected from training raw (or unprocessed) data from one or more multiple classes. The rationale of this thesis is that it is enough to train the spectrum detection model with a single type of signal (a single class). We will use only noise signals representing a free channel in this case. Any modulated signal (those transmitting information) present in the channel will be considered an anomaly. For this case, the techniques used in this type of detection are called semi-supervised since the data is only labeled as good since the starting data set contains data obtained in the normal operation of the process. For this type of fault detection, a classifier is trained

with the data corresponding to the system's normal operation. Therefore, an anomaly is identified when recorded data differs from the training pattern. On the other hand, depending on the approach given to the training data, there are three types of methods to address anomaly detection: prediction methods [65], limit setting methods [66], and methods reconstruction [67], [68]. The former tries to identify deviations from patterns based on their distribution. The second tries to establish spatial limits to separate normal data from an anomaly, and the third tries to reconstruct the input patterns and detect deviations based on said reconstruction. Anomaly detection techniques do not require collecting a vast amount of data from different types of modulated signals to train the detection model. It is optional to generate a complex spectrum detection model that requires excessive training times and avoids human errors during the training data labeling process. In the literature, several works that apply the anomaly detection theory to detect unusual patterns or behavior during the transmission of signals in wireless networks are reported [69], [70], [71], [72]. These works mainly focus on detecting fraudulent users, interfering signals, failures in a network node, or violating network security. However, none apply the anomaly detection theory to determine channel occupancy.

Chapter 3

Spectrum Sensing Through Machine Learning Classification

3.1 Background

Classification models have been modeled in the literature as a spectrum sensing problem for channel occupancy decisions.

Much like the previously presented studies, spectrum sensing can be formulated as an AD problem where the model decides on the observed channel; from here, the AD model perceives the empty channel as the normal state and any presence of user signal is then perceived as an anomaly. In the context of spectrum sensing, detecting an anomaly is therefore interpreted as the occupied status of the channel.

3.2 Spectrum Sensing

Channel occupancy is made through a binary hypothesis. Since a degree of noise presence is found in any sensed channel, the null and alternative hypotheses are respectively stated as follows:

$$\begin{aligned} H_0 &: x(n) = \eta(n) \\ H_1 &: x(n) = s(n) + \eta(n) \\ n &= 0, 1, ..., N-1 \end{aligned} \tag{1}$$

Where $\boldsymbol{\eta}$ corresponds to Additive White Gaussian noise (AWGN) and $\boldsymbol{s}(n)$ corresponds to the user signal. The task of a spectrum detector is to distinguish H_0 from H_1

in the observed bandwidth by comparing a test statistic $\gamma(x)$ to a threshold λ, where the test statistic can be a feature measured from the user signal and the threshold is usually obtained from a vacant channel.

$$\gamma(x) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \tag{2}$$

The performance of any spectral detector is usually measured by its detection probability and its false alarm probability, defined respectively as being the probability that a user signal is detected, or $P_D = Pr[\gamma(x) > \lambda; H_1]$ and the probability that a vacant channel is erroneously detected as an occupied channel, or $P_{FA} = Pr[\gamma(x) > \lambda; H_0]$. Therefore, an ideal detector would have a $P_D = 1.0$ and a $P_{FA} = 0.0$, respectively. However, this outcome is compromised by a low signal-to-noise ratio (SNR), and as a result, a higher P_D usually comes at the expense of a higher P_{FA}.

3.2.1 Anomaly Detection for Spectrum Sensing

An unsupervised AD model automatically extracts the observed signal features and makes a decision based on the model's output by either calculating an anomaly score through an error measurement between the output signal and the expected outcome or a numeric prediction of the model.

The anomaly score serves as the test statistics $\gamma(x)$ and is compared to a threshold λ to make a channel decision as formulated in 1 and 2.

3.2.2 Threshold Selection

When adjusting a threshold, it is important to consider the trade-off, where it can be adjusted for a higher P_D rate at the expense of a higher P_{FA} and vice-versa. A higher

P_D increases the chances of finding a user in the spectrum, avoiding causing interference to the user by transmitting over it, while a lower P_{FA} increases the chances of finding unoccupied channels, thus increasing the spectrum efficiency. An SNR-wall is defined as the minimum SNR in which a detector will fail to be robust [73]. This is usually considered whenever the P_D falls below 0.9 and above 0.1 P_{FA}.

SS algorithms typically adjust their thresholds to obtain a Constant False Alarm Rate (CFAR), where it is common practice to be set at a P_{FA} of 0.1. Although to a lower extent, Constant Detection Rate (CDR) at a P_D of 0.9 is also considered in some studies. Since CFAR is more predominant in the literature, the anomaly detector implemented in this work uses this threshold selection method for some of the results for comparison purposes with other detectors.

A CFAR or CDR threshold limits the detector's accuracy in most cases, as they fix the P_{FA} or the P_D to a specific value. The next section explains that P_D and P_{FA} can be maximized by finding a threshold that gives the best compromise between these metrics.

Accuracy optimization

Sensitivity and specificity are widely used metrics that describe the accuracy of a binary outcome when evaluating underground truth samples. The combination of the possible outcomes is visualized in Table I

Ground Truth	Obtained result	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table I: **Ground truth table for the binary classification decision.**

In AD, a positive outcome refers to detecting an anomaly. The sensitivity or True

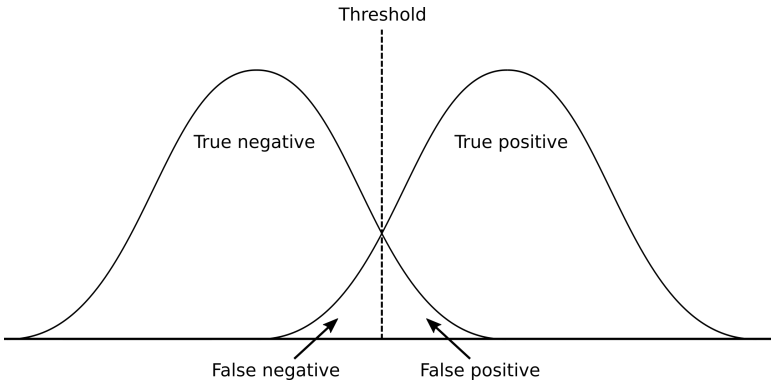


Figure 2: **Trade-off between classification performance in response to a threshold.**

Positive (TP) rate, also synonymous to P_D, is represented by the combination of obtained positive results and positive ground truths, while specificity is the True Negative (TN) rates, these being obtained negative results when they were expected, also described as $1 - P_{FA}$.

Model accuracy describes the rate at that the model makes no mistakes. Therefore its described by the sensitivity and specificity metrics as follows:

$$accuracy = \frac{TP + TN}{N} \tag{3}$$

Figure 2 represents the trade-off between the TP and TN rates. As both curves overlap, we see an increase of false negative (FN), the rate at which the ground class is mistakenly detected as negative, and the false positive rate, which is synonymous to P_{FA}. The thresholds shown in the figure represent the best compromise that maximizes the model's accuracy.

To find this threshold, Receiver Operating Characteristic (ROC) curves can be used to describe the classification performance for several discrimination thresholds. A Youden's index (J = [sensitivity + specificity]-1) [74] is then computed for all points of the ROC curve. The maximum value of the J index is used as a criterion for selecting

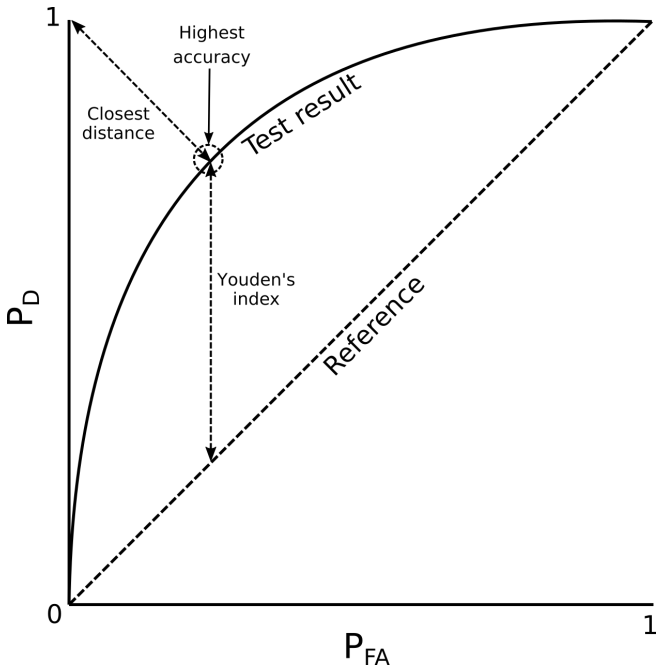


Figure 3: **Representation of a ROC curve. The reference line represents an outcome where results are randomly classified, in which cases the P_D would approach the P_{FA} values.**

the optimum γ that maximizes AD accuracy. For each optimum threshold γ, the accuracy is calculated from the sum of the samples classified as normal and outliers and then divided by the total of tested samples. The optimal trade-off for maximizing the accuracy can be visualized in a ROC curve as the shortest square distance in the upper left corner of the axes and the test result, as depicted in figure 3.

3.2.3 Model selection for anomaly detection

An anomaly detector can be built through conventional ML models such as the Supported Vector Machine (SVM). However, Deep Learning (DL), a branch of ML, has seen increased interest due to its capability of big data analysis, making it a powerful tool for data prediction and semantic interpretation, in addition to rapid improvements in computational power, fast data storage, and parallelization [75]. Additionally, DL has seen indisputable success in image processing and computer vision and has been

exploited for communications to handle several obstacles of traditional AI algorithms [56].

A DL model is built from a Neural Network (NN) composed of a stack of layers formed by artificial neurons, which mimic how the human brain operates. The layers in a NN can be configured in a way that gives them the flexibility to bring a wide variety of implementations for regression and classification problems.

The implementation of an anomaly detector for a NN depends primarily on the nature of the input data, which can be broadly classified into sequential (text, time series, speech, etc.) and non-sequential (such as images) [76].

Sequential data analysis often considers the long-term trends and variations of the individual (or groups of) parameters (such as financial or economic indicators). As such, time-series analysis is often used to predict future behavior. Usually, the pre-processing of the time-series parameters (predictors) is of secondary importance to building a predictive or explanatory model.

These sequential patterns can be learned by a Recurrent Neural Network (RNN). Which has seen applications in the fields of speech recognition [77], language modeling [78], and vehicle trajectory planning [79]. Anomaly detection is implemented in [80] for network intrusion detection through packet analysis based on cosine similarity. In [81], flight data analysis is performed to detect events that reduce safety margins.

For non-sequential data analysis, Autoencoders are widely explored in the literature due to the ease of implementation and straightforward intuitions [82]. They are an unsupervised neural network-based approach designed to learn an approximation of inputs using feature learning to capture an underlying distribution. They achieve this through an encoding and decoding process, where the encoding part learns to extract the most relevant data of a given input by imposing a bottleneck within the NN.

From here, the decoding part learns to use this compressed data to rebuild the original input. Thus anomaly detection consists of making the autoencoder more proficient at rebuilding the normal class, where the distinction between a normal and an anomaly is made by measuring the reconstruction error.

In [83], autoencoders achieved superior results compared to one-class SVM, isolation forest for anomaly detection through network traffic analysis for IoT-connected devices for intrusion detection. Furthermore, a convolution process can enhance the effectiveness of feature extraction for autoencoders.

Convolutional Neural Networks (convNets) are often applied for tasks such as image denoising and classification problems due to their capability to exploit their spatial characteristics and have seen great success in visual classification [60]. Moreover, convolution has been demonstrated to be able to learn features automatically with superior discriminatory power [84], [85]. Additionally, it serves as a solution to alleviate the significant limitation where training a NN with a large number of neurons can largely increase the computational complexity of this process [86].

ConvNets have also been applied for SS-related studies. In [60], a ConvNet is utilized in a cooperative sensing environment that learns to combine the individual sensing results to devise an occupancy decision of the channel. The viability of a ConvNet for Modulation recognition on time series radio signals is demonstrated in [87], where its performance is compared against more conventional expert feature-based methods. In [88], a ConvNet is employed for SS by classifying between user and noise using the energy and cyclostationary features learned from the occupied and unoccupied states of the channel. Compared to a high-layer network, AMC is also implemented using a ConvNet in [89]. The results demonstrated a higher performance of the ConvNet at low SNR conditions.

A convolutional autoencoder (CAE) is implemented in [90] for Acoustic Anomaly Detection (AAD) in which the CAE is trained with audio signal spectrograms, and it is compared to a One-Class Support Vector Machine. The detection is designed to maximize the model's accuracy by utilizing Youden's index, and the experiments are measured in terms of accuracy. The results demonstrated that CAE is more successful in detecting anomalies in lower SNR conditions.

Although designed for two-dimensional data (such as width x height), in [91], it is stated that 1D ConvNets have achieved state-of-the-art performance levels in several signal processing applications. Some of these include electrocardiogram monitoring for early arrhythmia detection [92] and real-time monitoring of high-power circuitry [93], among other studies. Anomaly detection for motor fault detection through vibration signal analysis is performed in [94]. The work presented in [95] uses a 1D CAE to restore the original message symbols of a signal below its symbol rate.

In this thesis, vacant channel detection is approached as an AD problem by using two implementations, the first being a 1D convolutional autoencoder for a non-sequential data approach, where signal samples are being treated as independent of one another, and a Long-Short Term Memory RNN for a sequential data approach. The following sections explain the architecture and the training process of these models.

Chapter 4

Neural Networks and Autoencoder Convolutional for Anomaly Detection

This section explains the generic architecture and functioning of a neural network (NN), as well as the training process in section 4.2. Section 4.3 explains the architecture of Convolutional Autoencoder. An introduction of the implemented Autoencoder followed by an explanation of the convolution process and its integration into the encoding and decoding process is described in Sections 4.3.1 and 4.3.2, respectively. Additionally, Section 4.3.3 presents the reconstruction error measurement metrics utilized for the convolutional autoencoder, which play a crucial role in anomaly detection. In addition to the CAE-based Anomaly Detection, a Long-Short Term Memory Recurrent Neural Network model based on entropy features in time series is described in Section 4.4. First, the concept of RNNs is introduced in section 4.4.1, as well as the general training process. The LSTM architecture is described in Section 4.4.2. The concept of entropy and the process of noise and signal conversion to entropy in time series is explained in section 4.4.3.

4.1 Neural Network Architecture

NNs are composed of an arrangement of neurons in the form of layers. The input layer is where the raw information is introduced to the NN, the hidden layer, which can either be absent or be conformed of multiple neuron layers, is usually where most of the computation takes place as the input data is transformed and is sent to the output

layer. Each neuron is connected as seen in Figure 4.

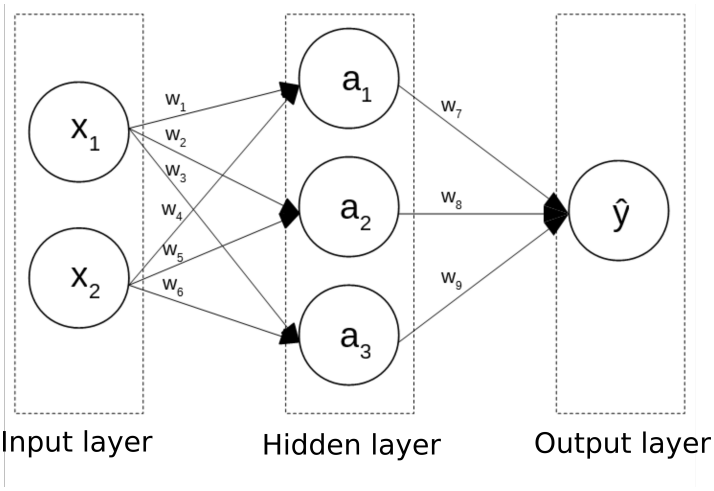


Figure 4: **Three layered neural networks, formed by an input layer, a hidden layer, and an output layer..**

The information sent between layers is transformed as such:

$$a = \rho(\theta^{\intercal} x + b) \tag{4}$$

Where an activation function ρ is applied to a transposed weight matrix $w^{\intercal}$ and a bias b which in some cases is omitted. The activation function transforms the output values of the layers adding non-linearity and allowing the model to learn complex patterns in the data. The activation functions used for the models presented in this work are represented in Figure 5, which include the sigmoid function, the Rectified Linear Unit (ReLU), leaky ReLU, and the tanh function, utilized in the second anomaly detection scheme.

Sigmoid is a widely used activation function for classification models defined as:

$$h_{\theta}(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

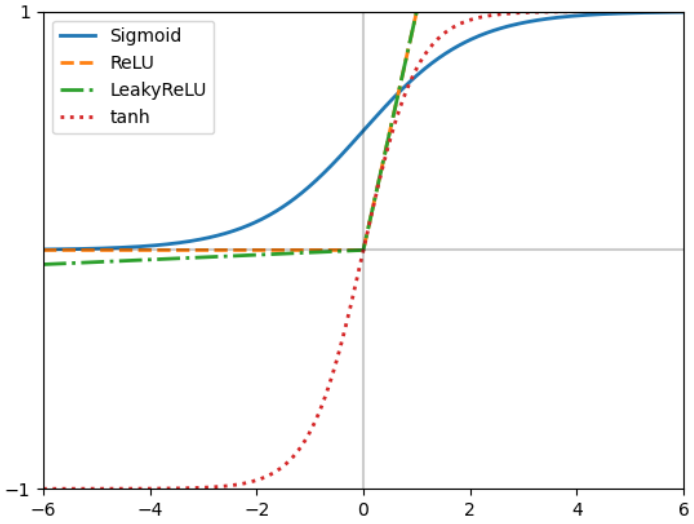


Figure 5: **Sigmoid, ReLU, LeakyReLU, and tanh activation functions.**

Which translates an input into a value between 0 and 1 in a non-linear manner, thus guaranteeing that the output of a neuron falls within this range. On the other hand, the Rectified Linear Unit (ReLU) is a linear function that converts any negative value as 0 and leaves positive values unmodified. A leaky ReLU function introduces a small slope on the negative side which converts any negative number to a value close to 0 instead while leaving any positive value unaltered as well. Finally, while having a shape similar to the sigmoid function, tanh translates the input into a value between -1 and 1 in a non-linear manner as well.

Depending on the application of the NN model the output $\hat{y}$ of the model can consist of single or multiple prediction values, usually ranging from 0 to 1. For classification, this value can represent the confidence that a sample belongs to a certain class. This confidence interval is given by an evaluation metric from which the output of the model is measured, where a value close to 0 could represent the confidence that a given sample belongs to class A and a value close to 1 for class B. A threshold is introduced to make

a final decision, i.e. a threshold set at the confidence interval 0.5 means that any value below or above it belongs to class A or B respectively.

4.2 Neural Network Training process

The data utilized during the training process contains the most relevant features in relation to the outcome and are either manually selected according to human criteria or automatically extracted during the process.

The training process of a neural network model is represented in figure 7. Before initializing the training process, data is divided into a training subset and a validation subset. The training subset is introduced into the model in batches, the batch size of the hyperparameter defines the quantity of data that is introduced.

The w and b of the model are commonly adjusted through the backpropagation algorithm by using a gradient descent optimization in response to a loss function, which measures the error of the model prediction in relation to an expected result. The gradient descent is used to adjust the model parameters in an iterative manner in order to minimize a loss function. It can be represented by a curve as seen in Figure 6.

As mentioned in [96], it measures the local gradient of the error function with regard to the parameter vector θ in the direction of descending gradient until it converges to a minimum. An important parameter in gradient descent is the size of the learning steps, which is determined by the learning rate hyperparameter.

The learning hyperparameter needs to be adjusted so it's neither too high so θ goes to the other side of the gradient, and potentially increases the error or too low to reach the minimum loss.

A training epoch is completed when all training data is used, however, it can then

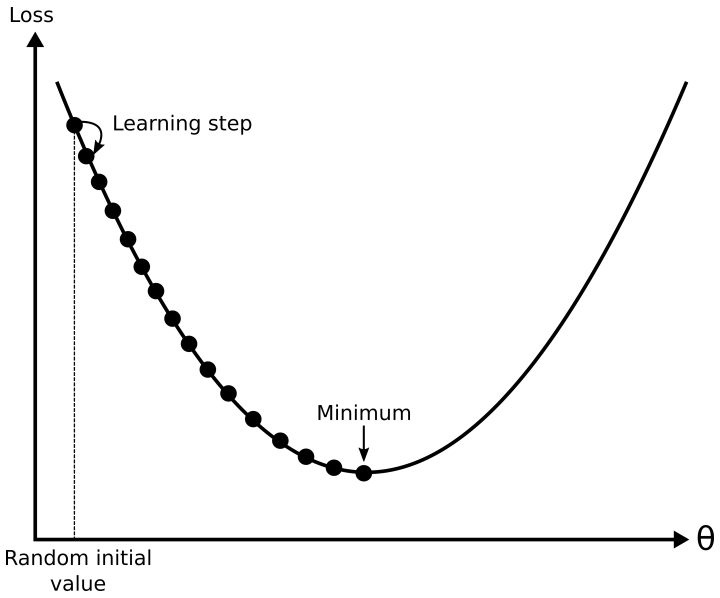


Figure 6: **Generic representation of gradient descent. The model parameters θ are initialized randomly. Through training iterations, these parameters lower the loss of the model proportionally to the learning step size.**

be reutilized to begin another training epoch and continue the training process. The number of epochs a model trains for is another hyperparameter in this process.

The generalization capability of the model is monitored during the training process after each batch with the validation subset in order to test the model without bias. Observing a higher performance in the training data when compared to the validation data means that the model is overfitting, which can occur after too many epochs. This leads to a model too used to the training data, degrading its generalization capability as it negatively impacts its performance when confronted with any data outside the training subset, therefore the training process should end when the loss function reaches the lowest point when tested with the validation dataset.

A test dataset is used to evaluate the trained model in a more rigorous manner to make a more precise estimate on the evaluation of a model. These are composed of samples outside the training and validation datasets and are formed by a larger number

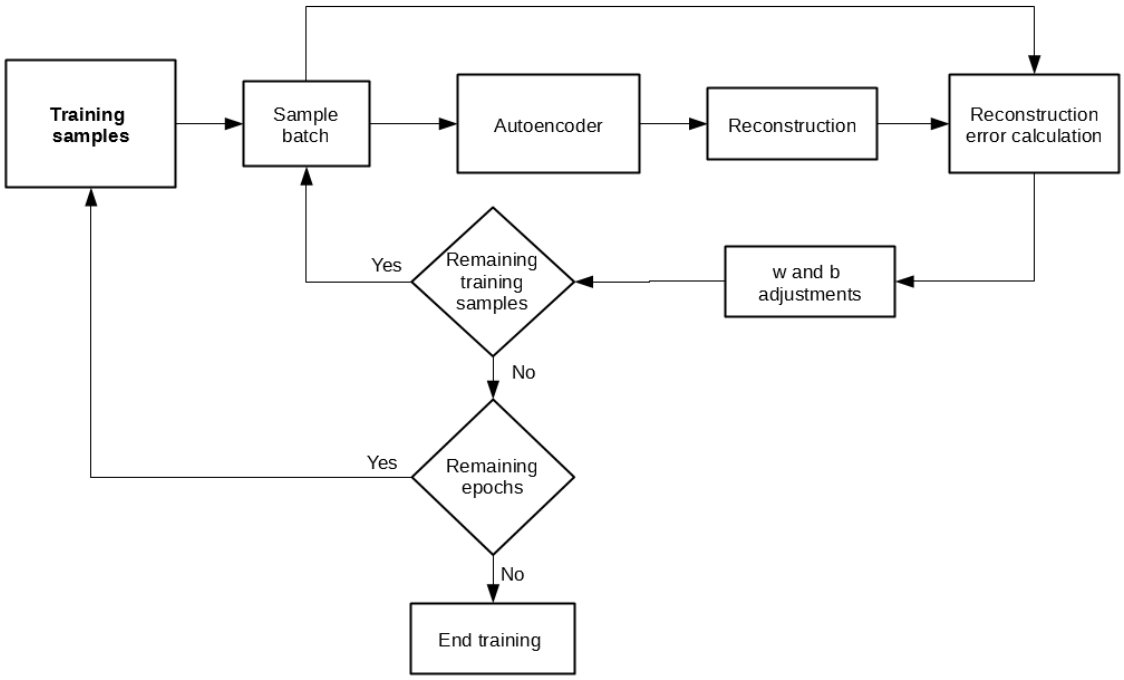


Figure 7: **Neural network training.**

of samples than the validation dataset.

This training and testing process is shared among the implemented algorithms in this work.

4.3 Convolutional Autoencoders Architecture

4.3.1 Autoencoders

Autoencoders are a type of neural network architecture that is primarily used for unsupervised learning tasks. Their purpose is to efficiently encode and decode data, aiming to reconstruct the original input from a compressed representation or encoding within

the network's hidden layers.

They consist of two connected networks: an encoder $f_\theta()$ and a decoder g_ψ. The encoder learns to map the input data $X \in \mathbb{R}^d$ to a low dimensional representation (latent or hidden layer representation) $Z \in \mathbb{R}^{d'}$. The model then attempts to reconstruct the original input X using the features extracted from the given input and converts them into Z. The decoder maps back Z to generate an output $\hat{X}_i$ as similar as possible to the original input using the reverse mapping function

This encoding and decoding process is done conventionally by reducing the number of neurons found in the hidden layers, then the reconstruction or decoding process is attempted by reverting the number of neurons per hidden layer until it matches the input.

A generic diagram of an autoencoder is represented in Figure 8. The encoding process is achieved conventionally by lowering the number of neurons within each layer to obtain a latent representation of the original sample, whereas the decoding process consists of increasing the number of neurons to match the length of the input layer.

4.3.2 Convolutional Autoencoders

Convolutional Autoencoders (CAE) differentiate from conventional autoencoders in that the CAE weights are shared among all locations in the input, preserving the spatial locality similar to ConvNets [97].

In autoencoders, convolutional layers are typically incorporated to capture spatial dependencies and local patterns in the input data. These layers produce feature maps, which are the aggregated results of multiple filters or kernels that slide across the input performing element-wise multiplications, these being the weights that the model adjusts

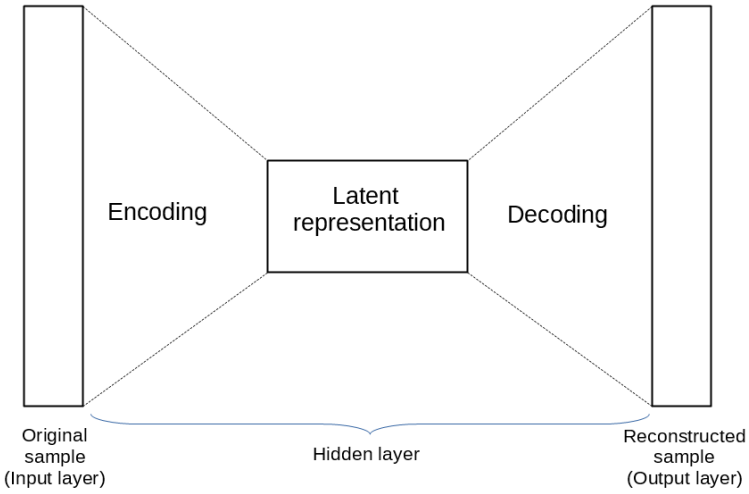


Figure 8: **Generic representation of the encoding and decoding process.**

during training:

$$a = \rho(\theta^{\mathsf{T}} * x + b) \tag{6}$$

In the context of convolutional operations in neural networks, kernel size refers to the dimensions of the sliding window.

Through this process, the filter can extract the features within the data with a property known as invariance. As mentioned in [98] invariance is a useful property for when the presence of a feature itself is the relevant part of the application, instead of knowing exactly where it is. In other words, being able to recognize an element independently of its location.

The kernel size has a direct impact on the behavior of the convolution operation. A larger kernel size captures more extensive spatial information and can detect larger

patterns or features in the input. However, it also increases the number of parameters and computational complexity of the model.

Conversely, a smaller kernel size captures more localized and fine-grained details in the input. It is useful for detecting smaller patterns or features. Smaller kernel sizes reduce the number of parameters and computational requirements.

The choice of kernel size is often determined through experimentation and consideration of trade-offs between model complexity, and the amount of information to be captured.

The encoding process as implemented in this thesis is achieved through the convolution process by utilizing a 2-stride kernel which reduces the data by half. The decoding process consists of upsampling the latent representation of the data and filtering it through a convolution kernel to restore it back to its original values, as presented in Figure 9.

The output reconstruction of the model is then evaluated through an error or determination function that compares it to the original sample.

4.3.3 Reconstruction error measurement

The error functions used with the CAE model to analyze their efficiency on anomaly discrimination include two of the most commonly used for model evaluation, which are the Mean Squared Error (MSE) and the Mean Absolute Error (MAE), as well as the R^2 coefficient of determination.

MAE measures the mean absolute distance between every value that composes both

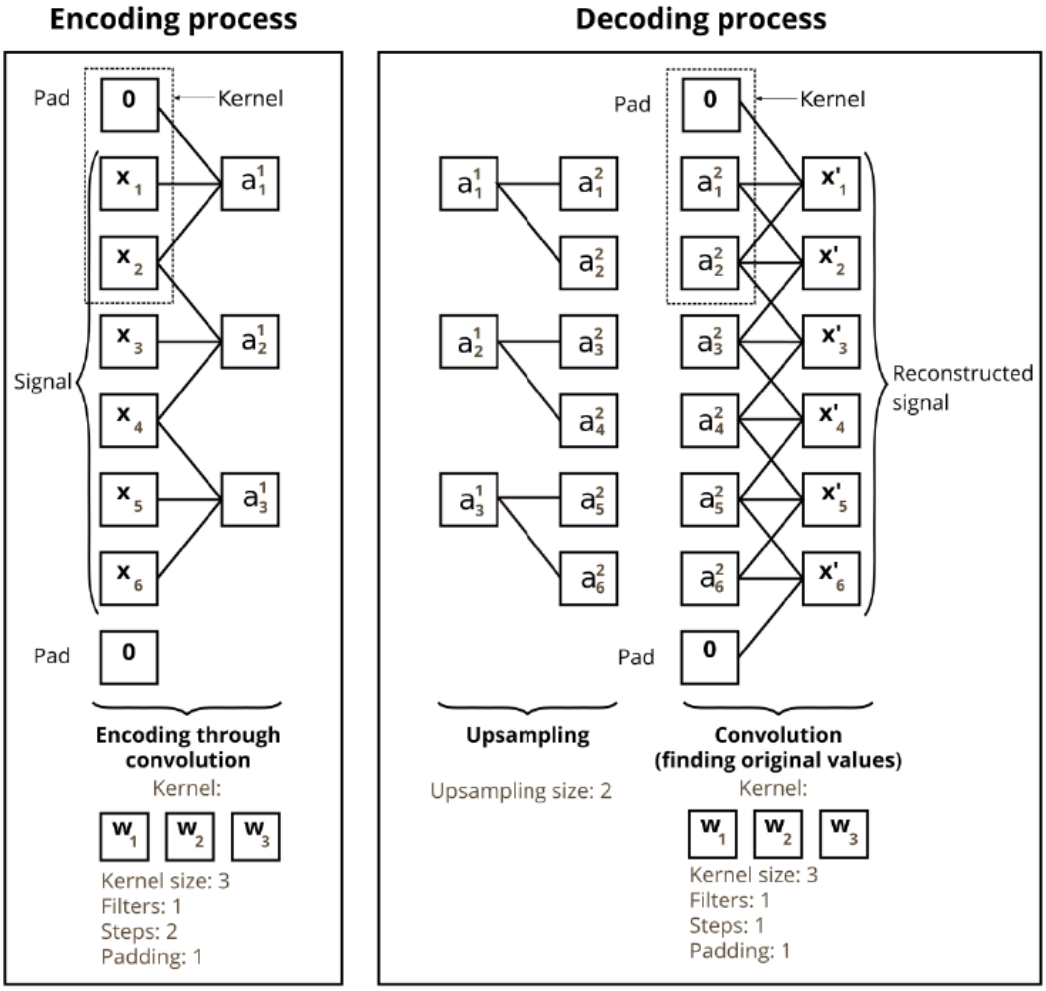


Figure 9: **Encoding and decoding process of a convolutional autoencoder. A filter kernel of size 3 is passed through the original two-step sample, compressing the original sample by half. In the decoding process, each value is repeated and filtered by a convolution kernel to reconstruct the original information.**

the original and the reconstructed sample, and is given by:

$$\frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - x_i| \tag{7}$$

For MSE the distance between each value is squared instead, giving more weight to larger differences:

$$\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - x_i)^2 \tag{8}$$

The R^2 determination coefficient is a statistical measure that determines the strength of the relationship between the variance of the reconstructed sample and the variance of the original. It is given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(x_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(x_i - \bar{y}_i)^2} \tag{9}$$

For MSE and MAE an error of 0 means that both samples are identical, whereas for the coefficient of determination R^2 this is represented by a value of 1. As such, the reconstructed sample of a signal for an anomaly detector trained with noise samples is expected to have an error closer to 1 and an R^2 closer to 0.

A reconstruction error serves as the anomaly score or test statistic $\gamma(x)$ to determine if the introduced sample is considered an anomaly and therefore decides on the occupied state of a channel. This decision is dependent on the selection of a threshold. However, reconstruction error measurement is not only necessary for the classification process but also plays an important role during the training process, which in turn affects the development of the model. Hence the implementation of three CAE models trained with different loss functions for performance comparison

4.4 Long-Short Term Memory Recurrent Neural Networks

4.4.1 Recurrent Neural Networks

An introduction to Recurrent Neural Networks is presented in the next section, and later in section 4.3.2 the convolution process and its inclusion in the encoding and de-

coding process are explained. Section 4.3.3 introduces the utilized reconstruction error measurement metrics for the convolutional autoencoder which serves as the decisive factor for anomaly detection.

Unlike a feed-forward NN such as the CAE, where the information is sent towards a single direction, Recurrent Neural Networks (RNN) send the information back to retain previous information. While feed-forward NN assumes that the outputs of the model are independent of the inputs, in an RNN the previous data influences the decision of the current output, thus RNNs specialize in time series and other sequential data.

The left side of Figure 10 depicts a recurrent neuron. The right side of the picture represents an unrolled version of a recurrent neuron, developing into a 3-layer network for a sequence of three elements starting from time $t - 2$.

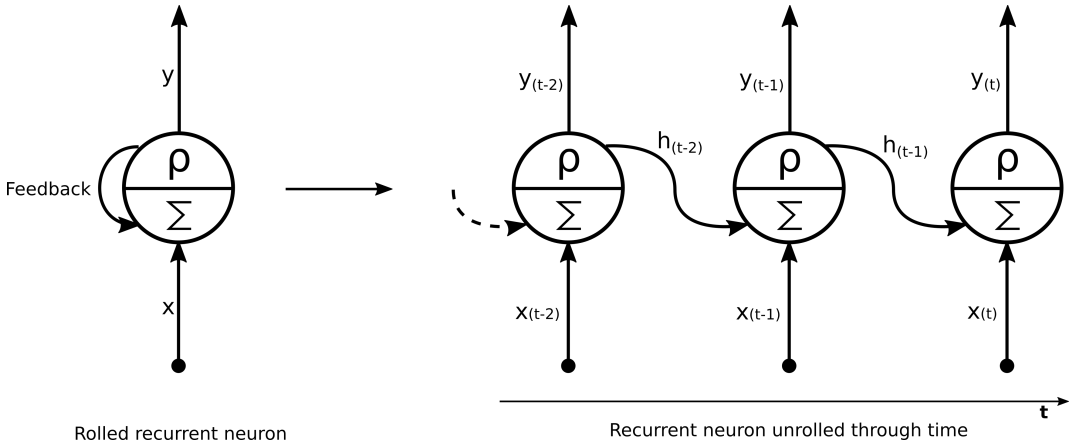


Figure 10: **Recurrent neural network.**

A hidden state $h(t)$ represents a state at time t retained or "memorized" by the network. $y(t)$ is calculated from the current input x and $h_{(t-1)}$, as such, a recurrent neuron utilizes two weights, being (θ_x) and θ_y. The output vector of a recurrent layer is computed as follows:

$$y_{(t)} = h_{(t)} = \rho(\theta_x{}^{\intercal} x_{(t)} + \theta_y{}^{\intercal} h_{(t-1)} + b) \tag{10}$$

An RNN can be configured to map one input to one output (one-to-one) as shown in the previous figure. However, the length of their simultaneous inputs and outputs can vary. One-to-Many refers to a single input being mapped to multiple outputs, many-to-one refers to many inputs being mapped to one output and many-to-many stands for multiple inputs to multiple outputs, as visualized in Figure 11.

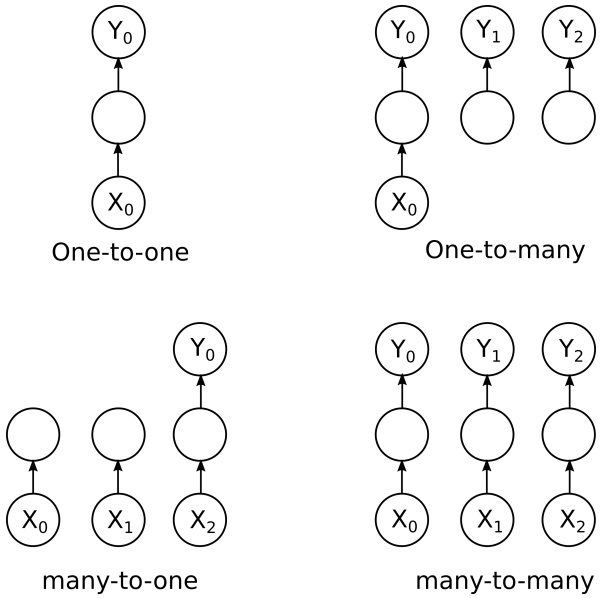


Figure 11: **Input and output sequences of a Recurrent Neural Network.**

A conventional RNN has a short-term memory, where they only take into account short-term dependencies within the data, making them settle in sub-optimal solutions [99] thus limiting their application. Long-Short Term Memory addresses this problem [100]. LSTM serves as an extension to a recurrent neuron, giving them the capacity to learn which data in a sequence is relevant to store long-term and when to forget it. The next section gives an introduction to LSTM cells.

4.4.2 Long-Short Term Memory

An LSTM cell consists of three gates, their architecture is shown in Figure 12..

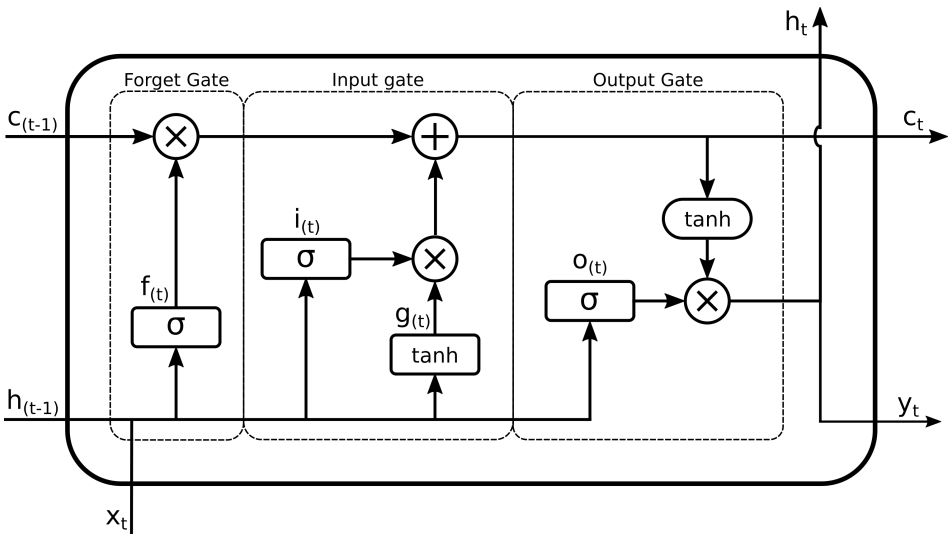


Figure 12: **LSTM cell architecture.**

It can be seen that the LSTM has an additional input, being the long-term state ($c_{(t-1)}$) that goes first into the forget gate. This part of the LSTM cell decides which parts, if any, of the long-state information are going to be erased and is given by:

$$f_t = \sigma(\theta_{xf}{}^{\intercal} x_{(t)} + \theta_{hf}{}^{\intercal} h_{(t-1)} + b_f) \tag{11}$$

The long-term state then reaches the input gate, where its added by the product of $i_(t)$ and $g(t)$. This part of the cell decides whether some parts of new information are added to the long-term memory.

$$i_t = \sigma(\theta_{xi}{}^{\intercal} x_{(t)} + \theta_{hi}{}^{\intercal} h_{(t-1)} + b_f) \tag{12}$$

$$g_t = tanh(\theta_{xg}{}^{\intercal} x_{(t)} + \theta_{hg}{}^{\intercal} h_{(t-1)} + b_f) \tag{13}$$

The output gate then decides which parts of the long-term state are going to be output at the present time step.

$$o_t = tanh(\theta_{xo}{}^{\intercal} x_{(t)} + \theta_{ho}{}^{\intercal} h_{(t-1)} + b_f) \tag{14}$$

$$y_{(t)} = h_{(t)} = o_{(t)} * tanh(c_{(t)}) \tag{15}$$

Finally, the long-term state is output from the LSTM cell as

$$c_{(t)} = f_{(t)} * c_{(t-1)} + i_{(t)} * g_{(t)} \tag{16}$$

Signal entropy is measured based on conventional entropy spectrum sensing works [101] [102]. However, the implementation for the RNN in this work consists of utilizing a sequence of entropy values taken from sub-samples of the signals involved in this work. The next section introduces the concept of entropy, furthermore, the conversion process is explained in 4.4.3

4.4.3 Entropy in time series

Entropy measures the amount of uncertainty involved in the value of a random variable, for a signal this is represented by the power magnitude, in this case, $\gamma(x)$ is defined by the level of uncertainty within a given frequency range. A random discrete variable is represented by a probability space, given by a sample space Γ, divided by an L quantity of sub-states. From here a probability P is calculated for each sub-state. Since p_i represents the probability for a given state and L the probability dimension space, we have that $\sum_{i=1}^{L} pi = 1(i = 1, 2...L)$ Shannon's information entropy contained within a message is defined as:

$$H(Y) = -\sum_{i=1}^{L} p_i * log_2 * pi \tag{17}$$

A histogram method is utilized where L is equal to the number of bins. k_i represents the total of occurrences in bin i, such as $\sum_{i=1}^{L} ki = N$. The probability of each state (p_i) is the rate of occurrences in bin i, thus, $p_i = \frac{k_i}{N}$. Replacing (17), the estimated entropy H_L is expressed as:

$$H(Y) = -\sum_{i=1}^{L} \frac{k_i}{N} * log_2 * \frac{k_i}{N} \tag{18}$$

A signal sample consisting of N amplitude values in the time domain is divided into n sub-sets with a length m and frequency domain conversion through FFT is then computed for each. A histogram of 15 bins is created for each sub-set of now amplitude values in the frequency domain, and the probability of occurrences is calculated for each bin, meaning each sub-set now consists of 15 probability values. Finally, Shannon's entropy is computed for each sub-set of amplitude values as depicted in Figure 13, leading to a vector of entropy values of size k equal to the number of sub-sets ($k = n$). This process, starting from a matrix of frequency domain amplitude values is depicted in Figure 13

4.4.4 Sensing process

User signal is expected to have a lower entropy value, while for AWGN this is maximized, therefore the predicted entropy for a user signal should be overestimated by a model trained to predict the noise entropy behavior, therefore this entropy prediction serves as the anomaly score or test statistic $\gamma(x)$ to determine whether the input sample is an anomaly if the measured entropy falls below a given threshold λ.

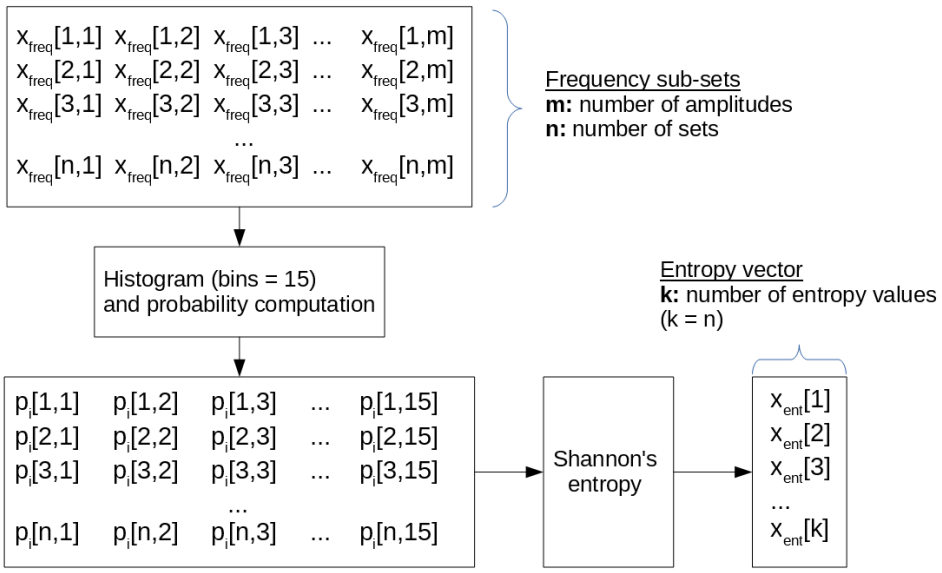


Figure 13: **Conversion process, from a matrix of frequency domain amplitude values to a sequential entropy vector.**

Chapter 5

Anomaly Detection Model Implementation

This section begins by presenting the utilized hardware and the dataset generation procedure for training and evaluation of the models. The following sections 5.2 and 5.3 explain the architecture and the parameters for the implementation and training of the CAE and the LSTM models, respectively. Finally, the threshold selection process for CFAR and accuracy optimization is explained in Section 5.4.

5.1 Dataset generation and training process

The dataset utilized for training, testing, and threshold selection is displayed in Table II. Each signal type within the dataset contains a single array of amplitudes, where the training/evaluation noise was captured over a lapse of 20 minutes, and the rest of the signals were captured over a lapse of 5 minutes. A window of 1024 samples is equivalent to 0.002048 seconds of observation.

Most datasets were generated through a vectorial signal generator with a data rate of 1 MHz and a symbol rate of 100 KHz. These signals were sent in a wired manner to an RTL-SDR USB dongle with an R820T tuner. GNU Radio was used as the software tool for signal processing. The incoming samples were decimated by a factor of two before being saved to the hard drive. The GNU radio simulated AWGN was generated within the software, and the vacant channel noise was captured from an antenna connected directly to the Nooelec R820T. Finally, The NVIDIA Jetson AGX Xavier Developer Kit was used to train the CAE models used in this work, as seen in Figure 14

Generated Dataset				
Signal	**Purpose**	**Source**	**Total amplitudes**	**Sample rate (Hz)**
AWGN	Training/Evaluation	SMU200A	600,000,000	500,000
AWGN	Threshold selection	SMU200A	150,000,000	500,000
Simulated noise	Model testing	GNU Radio	150,000,000	500,000
Vacant channel noise	Model testing	Nooelec R820T	150,000,000	48,000
QPSK (-15 to 5 dBs)	Model testing	SMU200A	150,000,000 (per dB)	500,000
16QAM (-15 to 5 dBs)	Model testing	SMU200A	150,000,000 (per dB)	500,000
64QAM (-15 to 5 dBs)	Model testing	SMU200A	150,000,000 (per dB)	500,000

Table II: **Generated dataset for model training and testing.**

Figure 14: **Diagram of the hardware used for signal generation, capturing, and model training.**

Both models were trained and tested with the same datasets. Several samples (or subsets) were created out of the total amplitudes for each signal type through a windowing process, where a sample was created for every n quantity of amplitudes, i.e., a subset of 1000 samples with an amplitude length of 1024 will utilize $1000 \times 1024 = 1,024,000$ amplitude values from the given dataset.

5.2 Convolutional Autoencoder implementation for non-Sequential Anomaly Detection

5.2.1 Model Architecture

The CAE implemented in this work as an anomaly detector has a deep architecture organized in multiple encoder-decoder convolutional layers, where the encoder and decoder structure consists of three convolutional layers each, as shown in Figure 15.

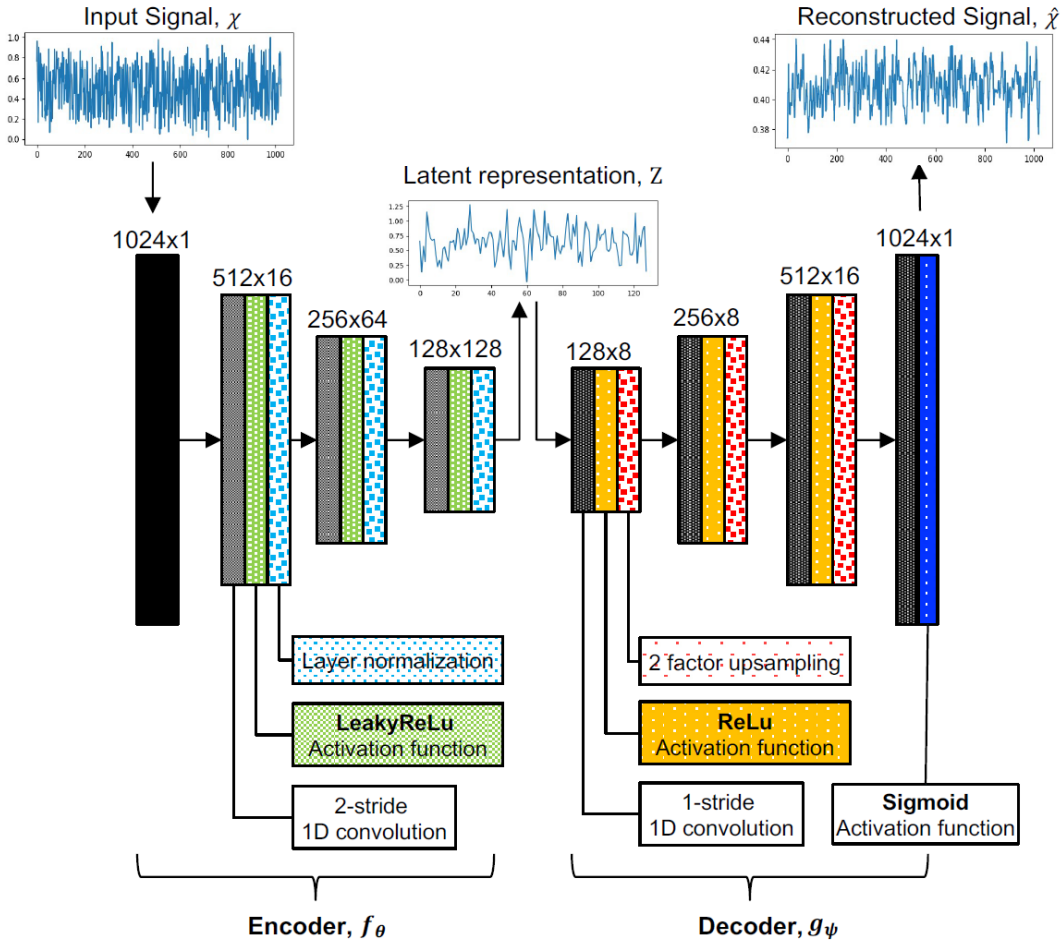


Figure 15: **Architecture of the implemented CAE model.**

A sample represented by 1024 amplitude values is introduced to the input layer. At

the start of each layer, the signal is filtered through a convolution process with a kernel of five by making steps of two, reducing the sample size at the output of each convolutional layer by half. A leaky ReLU function transforms any negative values output from the convolution process, multiplying them by a factor of 0.01. Furthermore, a layer normalization is applied that converts the output values of the layer to a minimum of 0 and a maximum of 1. This process is repeated for each layer in the encoding part of the model, where the number of convolution filters is increased, being 16, 64, and 128, respectively, with a kernel size of 5. The encoder output will be a latent representation of 128 amplitude values.

The decoding process consists of three layers where a convolution process is also applied to the introduced values, this time by a step of one. A ReLU activation function changes the negative values to 0; then, a 2-factor upsampling repeats each element of the compressed sample. The filters in the convolution process within each layer attempt to reconstruct the original sample out of the latent representation, where the number of filters used consists of 8, 8, and 16, respectively, with a kernel size of 5. At the output layer, one last convolution is performed using one filter and a sigmoid activation function which performs a non-linear transformation of values to a range between 0 and 1.

Six CAE models were trained for this work, each sharing the same architecture mentioned above. In the time domain, three models were trained for each loss function (MSE, AME, and R^2). In the next section, the training process and parameters for these models are explained.

5.2.2 Model Training

The CAEs in this work were trained using unlabeled min-max normalized AWGN samples as the normal state of a channel.

The training dataset consists of a total of 500,000 generated noise samples, where each sample is formed by 1024 amplitude values. They were split into a training data subset of 450,000 samples and a validation data subset of 50,000 samples. Before beginning the training process, the training subset is shuffled to break any continuity between the samples and avoid model overfitting.

The dataset is introduced into the model in batches of 256 samples to optimize the model's parameters to minimize the reconstruction error, where an A Stochastic Gradient Descent (SGD), a variant of the gradient descent that optimizes the learning process for large datasets, is utilized with a learning rate of 0.02.

A new training epoch begins after the 450,000 samples are processed, and the previous training data subset is re-utilized. the validation data is used to monitor the training process by observing the unbiased classification performance of the model. The CAE is trained after finalizing 50 epochs.

5.3 Recurrent Neural Network For Sequential Data Anomaly Detection

5.3.1 RNN Model architecture

The implemented LSTM RNN anomaly detector model consists of two stacked LSTM layers. The diagram of the model architecture is displayed in figure 16. The first layer consists of 32 LSTM cells, outputting a sequence of 32 predicted entropy values in a

many-to-many configuration. The second layer consists of 16 LSTM cells that deliver a single output (many-to-one). Thus, the output layer consists of a single neuron with no activation function and outputting the predicted entropy value.

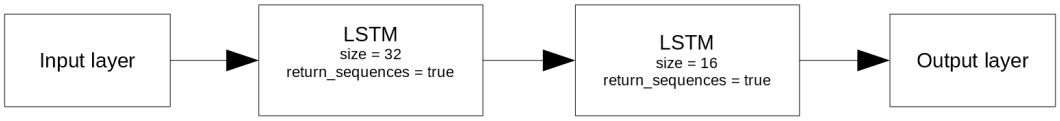


Figure 16: **Model architecture of the implemented LSTM.**

5.3.2 RNN Model training

The training for the LSTM model was supervised using entropy sequence samples computed from a continuous noise sample of 112,640,000 amplitude values extracted from the training dataset. Each sub-set of noise samples had a length m of 1,024 amplitudes. Shannon's entropy was calculated for each subset, resulting in $112,640,000/1,024 = 110,000$ sequential entropy samples. The entropy samples were normalized by the highest possible entropy, where a histogram of 15 bins where utilized for its calculation as:

$$Hnorm = \frac{H}{\log_2(15)} \tag{19}$$

A window with a step of 1 and a length of 65 was slid across the entropy samples. For each step, a sub-set was created until a matrix of 110000 elements was formed as shown in figure 17

Each entropy sub-sample consists of 65 sequential entropy values; from here, 64 are used for model training, and the last value in the sequence is used to compare the model output to calculate the prediction error. The training and validation dataset comprised 100,000 and 10,000 labeled entropy samples.

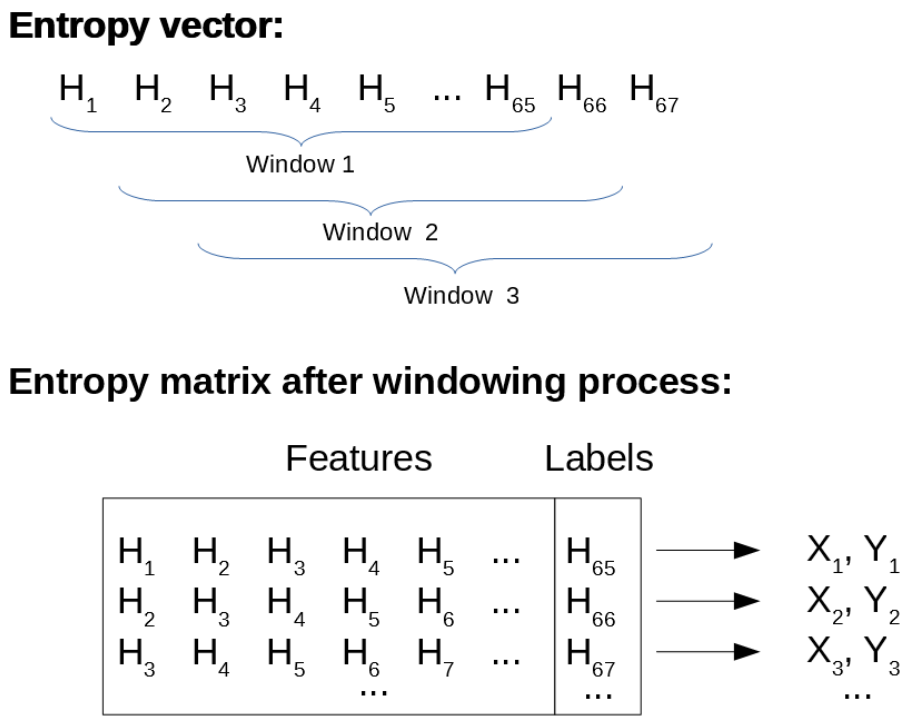


Figure 17: **Entropy vector to the matrix through the windowing process.**

Like the implemented CAE, the model uses an SGD learning algorithm and an MAE loss function for error measuring. A learning scheduler was included, where instead of using a fixed learning rate for the whole training process, the learning rate diminishes gradually as the training progresses in epochs. At the beginning of the training process, the learning rate hyperparameter starts at 0.0001. With every epoch, this value goes down by $1e-8 \times 10^{(1/20)}$, allowing the model to make a finer weight adjustment as it approaches the optimal solution. This training process was configured for 20 epochs and a batch size of 64.

5.4 Threshold Selection Process

CAE and LSTM models use a Cumulative density function (CDF) threshold-finding method. A CDF gives the probability that a random variable takes on a value less than or equal to a given value. For threshold selection, a decision is made regarding the design of the detector in terms of the trade-off between PD and PFA. For CFAR detection, a threshold is set based on a given PFA.

This process is exemplified in Figure 18 for an anomaly score based on MAE. A thousand AWGN test samples representing vacant channels are introduced to the trained model. A CDF is then created from the calculated anomaly scores of each AWGN sample.

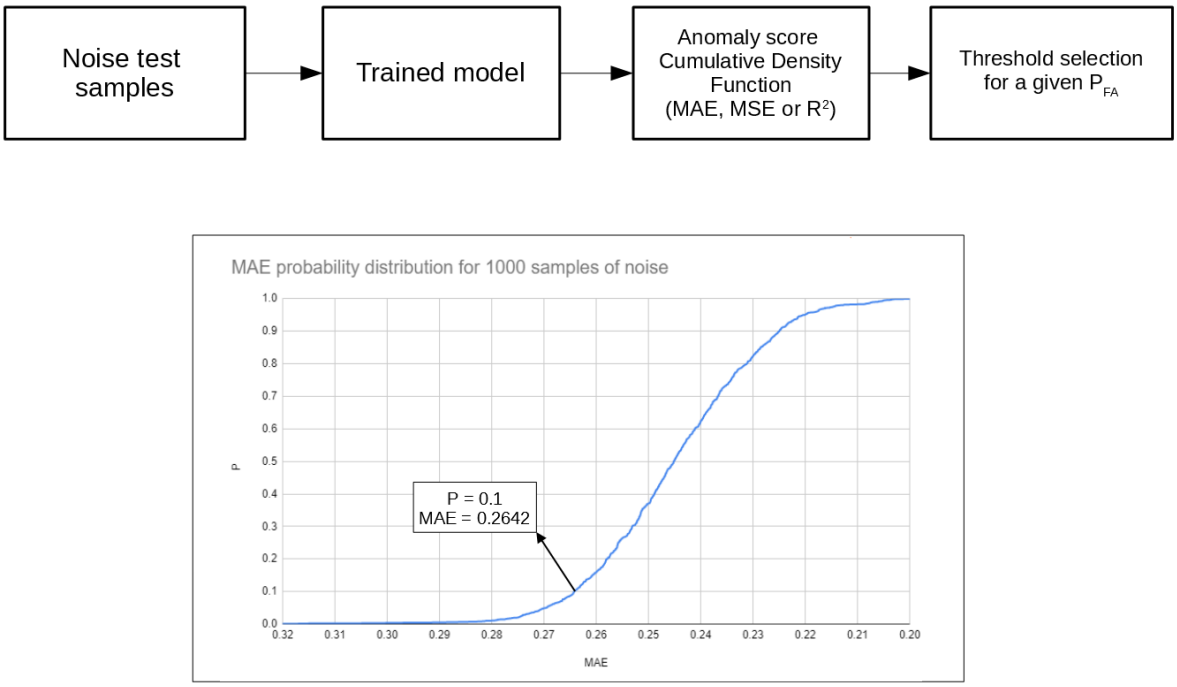


Figure 18: **Threshold selection process for a given P_{FA}. The example shows the CDF for the anomaly scores using MAE, where a threshold for a constant $P_{FA} = 0.1$ is indicated. Any sample with an anomaly score above 0.2642 would therefore be considered an anomaly.**

The value of the y axis of a given point in the CDF curve represents the probability

of obtaining the anomaly score below or equal to the value of the x axis. In the example, we see that the indicated point in the curve is composed of $y = 0.1$ and $x = 0.264$, meaning that while measuring noise, an anomaly score equal to or below MAE $= 0.2642$ would have a $P = 0.1$ chance of occurring.

If the anomaly score is expected to be lower in the presence of user signals, the values in x would be arranged in descending order (the tail of the curve would be on the opposite side). However, a lower MAE is expected when pure noise is perceived by the model.

This detection method inverts the hypothesis testing from what is presented in 2, where a higher MAE than the threshold would indicate that the channel is occupied, while a vacant channel detection would occur with lower MAE values.

Although once defined, the threshold remains constant for every SNR condition, the previous process for threshold identification is repeated for MSE, R^2, or entropy.

Chapter 6

Evaluation of the Anomaly Detection-Based Spectrum Sensing

In this chapter, the spectrum sensing capabilities of the CAE and LSTM algorithms are assessed. Initially, the performance of the CAE implementation is evaluated by utilizing a CFAR detection model for each signal modulation and the results are then compared to the conventional energy detector. Subsequently, the CAE model is optimized for accuracy under varying SNR conditions in Section 6.1.2. Section 6.2.1 centers on the evaluation of the LSTM algorithm using a CFAR detection model, followed by a comparison of its detection performance with the CAE. Subsequently, Section 6.2.2 focuses on optimizing the LSTM model for improved accuracy.

A total of 65 test subsets were collected for model evaluation, where 63 of them consist of modulated-signal samples and two AWGN, where one is used for P_{FA} evaluation purposes and one for the threshold selection process. All test subsets comprise 1000 samples where signal subsets contain modulated-plus-noise signals at a fixed SNR level. Three modulation schemes are considered for the performance evaluation (QPSK, 16QAM, and 32QAM) at sixteen SNR levels ranging from -15 to 5 dB. Due to receptor hardware limitations, a constant noise power was fixed at -50 dBm for all samples.

6.1 Performance evaluation of CAE-based spectrum sensing

The experiments in this section consist of numerical simulations to analyze the performance of the proposed Anomaly detection-based Spectrum Sensing approach.

It is important to note that during the experimentation, it was found that the CAE models performed significantly better at discriminating the anomaly samples when their output was inverted, such as $\hat{y}_{inv} = 1 - \hat{y}$ before doing any error measurement. However, this applies only to anomaly scores calculated through MSE and MAE, as R^2 being a statistical measurement of the variance, doesn't consider the distance between each of the original samples and its reconstructed version. Therefore, each result presented involving MSE and MAE for error calculation was done with an inverted version of the signal reconstruction.

The plot in 19 shows the MSE measurement for a QPSK signal for -15 to 5 dB SNRs using an inverted version of the reconstruction regarding a non-inverted version. It shows a more defined error curve as the SNR becomes greater, allowing a threshold to make an easier detection at higher SNR values. However, when using a non-inverted version, the error remains almost constant, thus complicating the classification process.

Additionally, it was found that although the CAE model was trained for noise signal recognition, it could more accurately reconstruct signals at higher SNR in terms of R^2. For this case, the decision for AD was so that reconstruction errors below a given threshold were considered anomalies.

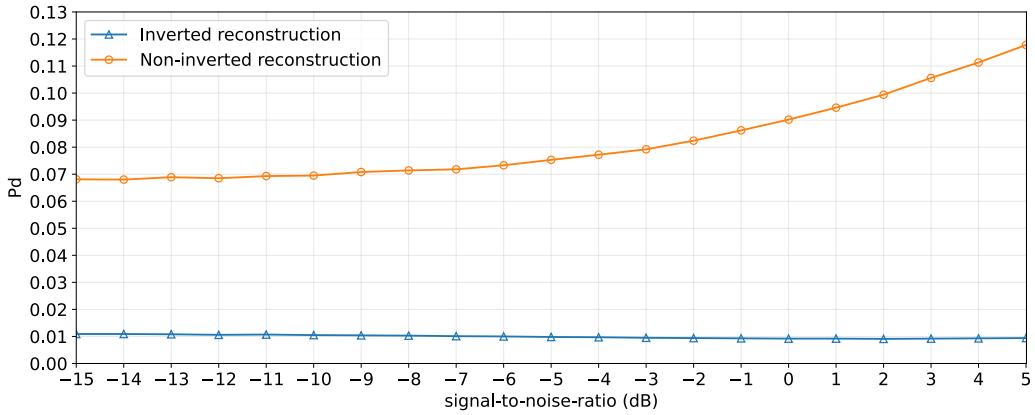
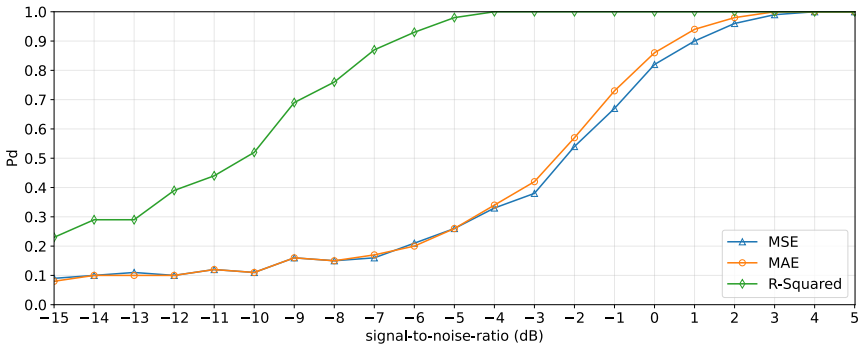


Figure 19: **Architecture of the implemented CAE model.**

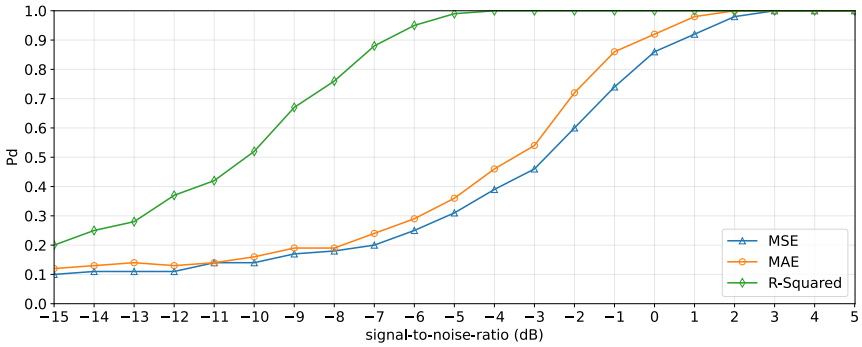
6.1.1 Constant false alarm sensing evaluation

Detection performance for QPSK modulated signals under noise conditions ranging from -15 to 5 dB are shown in Figure 20 for MSE, MAE, and R^2 loss trained noise CAE models. These results show the influence of the loss functions for training and the model's behavior for AD using each as an anomaly score. The detection performance for each of the tested anomaly scores shows similar behavior for each tested model, where MAE is slightly above MSE in terms of detection rate performance, where both reach an SNR-wall at 0 dB and around 1 dB, respectively. Using R^2 as an anomaly score offers the best results in the three models reaching the SNR-wall at -7 dB.

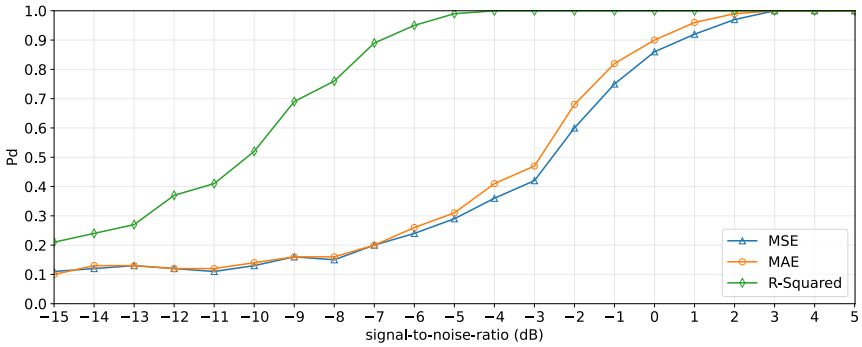
The detection rate of the R^2 trained model was tested using amplitude lengths 8192, 4096, 2048, 1024, 512, and 256 for each sample. A comparison for each anomaly score is depicted in figure 21. These results show that the model can increase its P_D for lower dB values if the amplitude length increases for any tested anomaly scores. However, R^2 (Figure 20.III) shows a further improvement in lowering its SNR wall to -12 dB by using a length of 8192 (a 0.016384-second detection window), while for MSE (Figure 20.I) and MAE (Figure 20.II), it is reached at -7 dB.



(I)



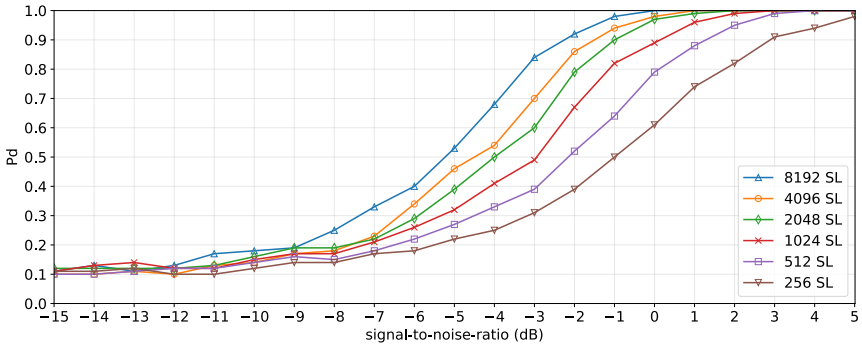
(II)



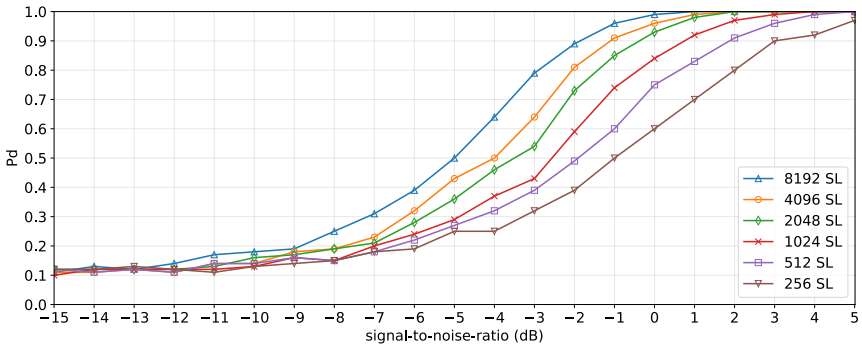
(III)

Figure 20: **SNR performance using models trained with MSE (I), MAE (II), and R^2 (III) loss functions.**

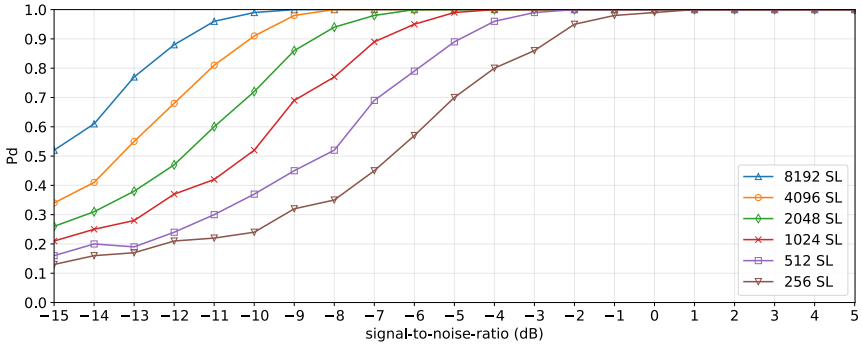
Similar to the results shown in Figure 20, the detection rate for MSE, MAE, and R^2 for the R^2 trained CAE model is illustrated in Figure 22. An improvement for MSE and MAE-based detection is seen, where the SNR wall is reduced by around 3 dB in MSE and CAE-trained models (Figure 22.I and Figure 22.II) and by 4 dB for the R^2



(I)

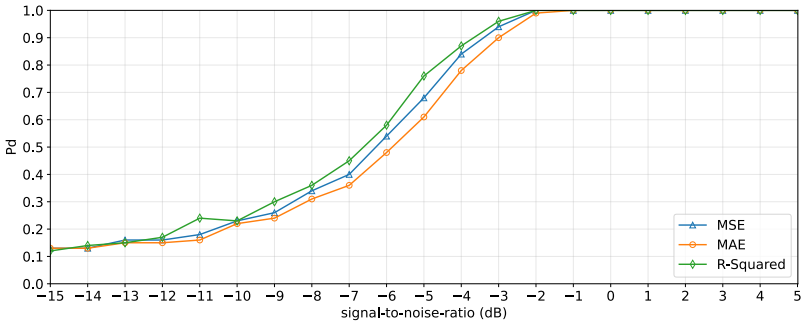


(II)

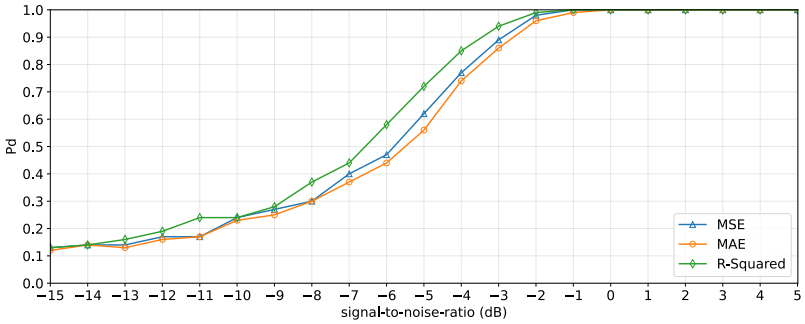


(III)

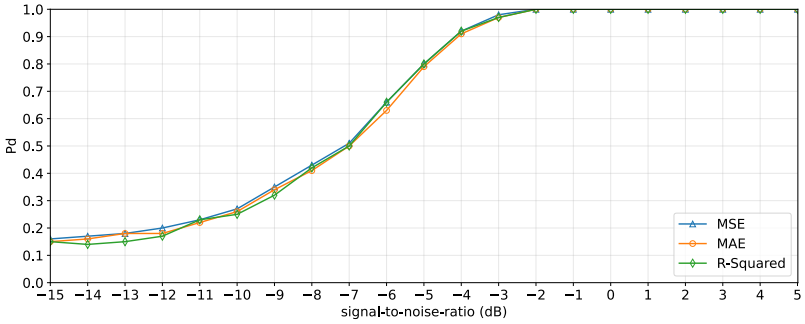
Figure 21: P_D **performance comparison for diverse amplitude lengths.**



(I)



(II)



(III)

Figure 22: **Dection performance using models trained with MSE (I), MAE (II), and R^2 (III) loss functions in the frequency domain.**

trained model (Figure 22.III). However, the detection rate for R^2 based detection is degraded in all three models by around 3 dB, reaching a performance similar to MSE and MAE, where in the R^2 trained model, they all share the same detection behavior.

The following results presented in Figure 23 evaluate an R^2 time domain noise trained CAE using MSE, MAE and R^2 for detection in terms of $P_{FA\_obtained}$ against

$P_{FA\_expected}$ from different noise sources. The reference line represents the expected behavior of any noise source, "Vacant Channel noise" represents noise samples taken from a real environment, "Simulated noise" are noise samples generated in GNU Radio, and "SMU200A generated noise" corresponds to noise samples generated from the vectorial generator.

The obtained P_{FA} while using MSE and MAE remains close to the expected outcome for time domain models. However, the results for vacant channel noise in frequency domain models tend to output a much higher P_{FA} than expected, while simulated noise tends to be lower. R^2 detection shows a more unpredictable behavior for simulated and vacant channel noise in the time domain, where their curves tend to be in the lower right corner. In the frequency domain, vacant channel noise behaves similarly to MSE and MAE, where the obtained P_{FA} is higher than expected while remaining closer to the reference line. For simulated noise, the P_{FA} remains nearly 0 for every expected P_{FA}.

Figure 24 illustrates the generalization ability of the proposed detection algorithm. The results show that R^2 based detection offers nearly identical performance for QPSK, 16QAM, and 32QAM modulated signals, while MAE also offers a similar behavior, where the detection performance difference of 16QAM and 32QAM is relatively insignificant, their SNR-wall is at a higher SNR compared to QPSK.

Based on the previous result, in Figure 25, the performance of the AD-based SS with other similar works is reported [53], [103]. These reported proposals also consider a non-cooperative Cognitive Radio Network. The detection performance results reported in [53] and [103] achieve better detection than the AD-based SS proposal. Its superior performance relies on sufficient training on a massive amount of labeled training data. Additionally, the signal samples used for the model training (BPSK, QPSK, 2FSK,

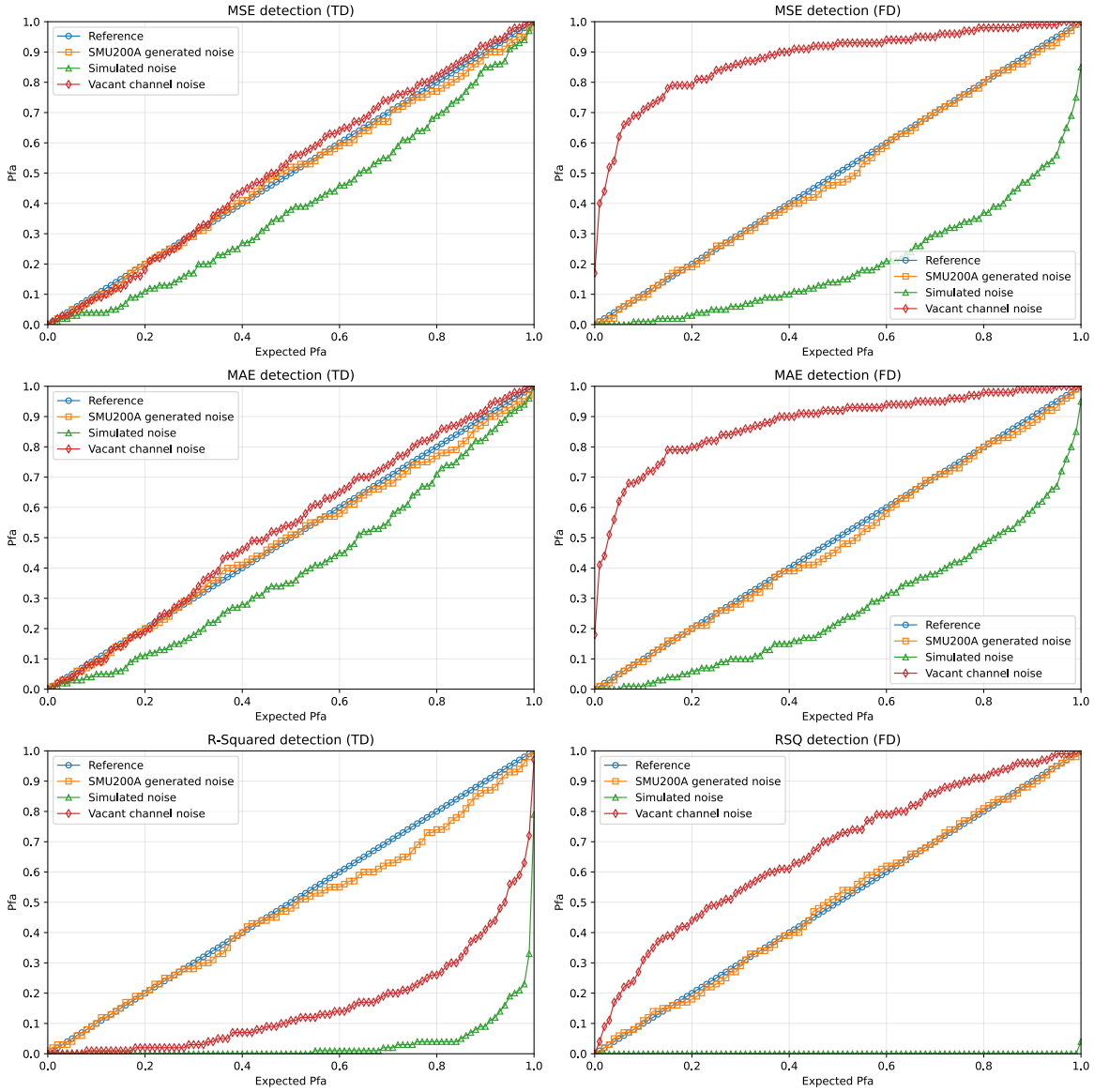


Figure 23: **Estimated P_{FA} vs expected P_{FA} for MSE, MAE and R^2 based detection in time and frequency domains.**

4FSK, 16QAM, 32QAM, 4PAM, and 8PAM) are the same family as the signals used in the detection phase (8PSK and 64QAM). Although the signal samples are not the same, the trained ConvNet-detector is insensitive to the magnitude order of the modulation. Regarding the results reported in [103], the Unsupervised Deep Learning Spectrum Sensing (UDSS) used only unsupervised data (raw and noise sensing samples) during

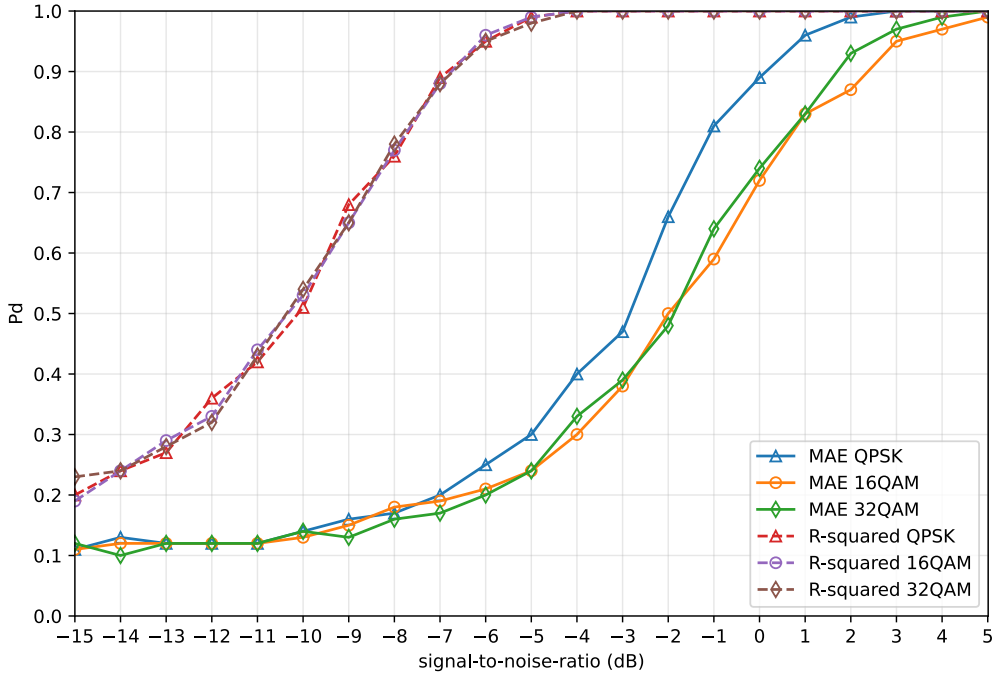


Figure 24: P_D **vs SNR curves for QPSK, 16QAM and 32QAM modulated signals for MAE and** R^2 **based detection.**

the training process. But after the learning process, it applies a small amount of labeled noise data for cluster classification and threshold estimation. IEEE 802.22 standard real-time raw sensing samples are used for signal detection performance. The results of UDSS shown in Figure 25 correspond to an uncorrelated channel scenario for a user equipped with eight receiver antennas.

To illustrate the advantage of the anomaly detector over the conventional energy detector, the SNR wall of the energy detector is -3.75 dB for a QPSK-modulated signal. The SNR wall of the proposed method is -7.45 dB for R^2 based detection for a QPSK signal, which is 3.7 dB lower.

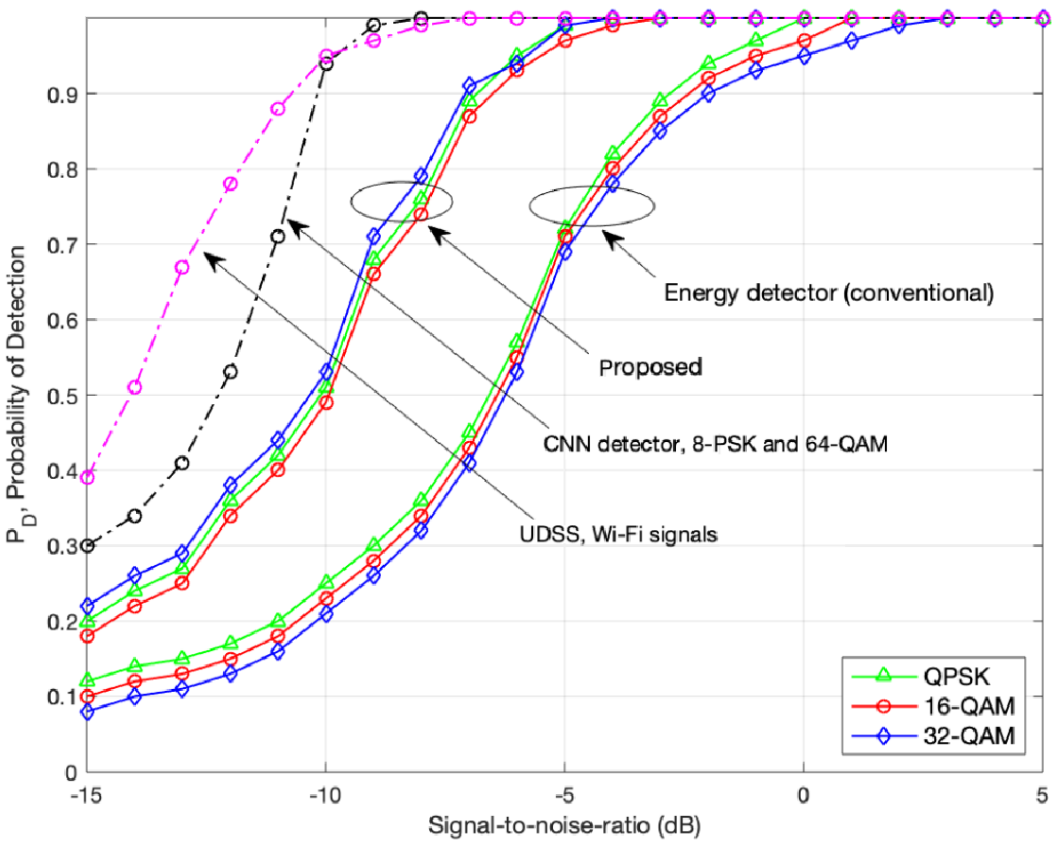


Figure 25: P_D **vs SNR curves for QPSK, 16QAM and 32QAM modulated signals for MAE and** R^2 **based detection.**

The AD model was tested against the deepsig.ai dataset RADIOML 2018.01A, as generated in [104], which is composed of a diversity of modulated signal sub-samples with a sample length of 1024 amplitude values. Each sample is impaired to a different random degree in terms such as Rayleigh fading, which affects the received energy of the signal due to obstacles that may be presented in a real environment. In order to compare the previous results involving the SMU200A generated signals, the following results include the same modulations QPSK, 16QAM, and 32QAM as found in the deep signs dataset, and additionally, the modulations GMSK, OQPSK, BPSK, and FM are included. The utilized model for this test was a CAE model trained with the R^2 loss function. In regards to MSE and MAE detection, we can observe in Figure 26 that the

SNR-wall in the time domain is found at around -1.5 dB for nearly all signals, however FM and more remarkably BPSK show a different behavior, where the SNR-wall for the FM signal reaches -2 dB, while for the BPSK signal, it is reached at 2 dB. We see a general improvement in terms of detection in the frequency domain, where the SNR-wall is lowered by 2 dB in nearly all cases, whereas for FM it is improved by more than 4 dB.

R^2 detection shows a different behavior, where in the time domain, the SNR-wall is reached at around -6 dB, however, BPSK still shows a lower SNR-wall, being at -2 dBs. R^2 detection in the frequency domain shows a lower performance, while the SNR-wall is raised by 2 dB for nearly all signals, where FM is the most affected, being raised by 6 dB.

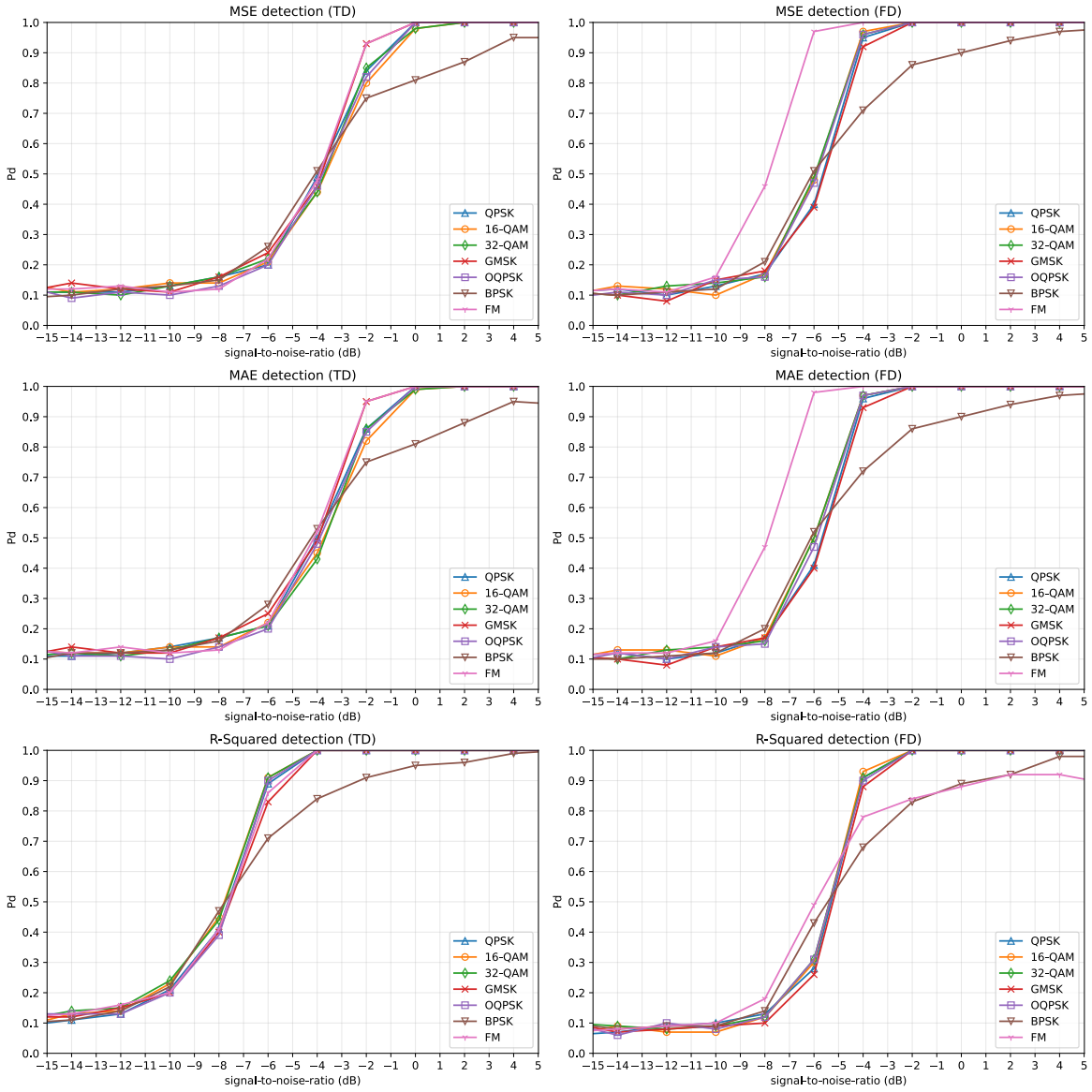


Figure 26: P_D **vs SNR curves for deepsig's QPSK, 16QAM, 32QAM, GMSK, OQPSK, BPSK, and FM for each anomaly score in time and frequency domain.**

6.1.2 Model Accuracy Evaluation and optimization

For the results of this section, the best case of the previous results was considered, being an R^2 trained CAE, as well as the use of R^2 for classification decisions.

Each point in the ROC space shown in Figure 27 compares the result of the P_D

and P_{FA} as the threshold setting γ changes. For each test dataset, 100 different γ values are evaluated, consistent with a P_{FA} ranging from 0.01 to 1 in steps of 0.01. The performance evaluation of 4 test datasets corresponding to QPSK signals at SNR = -6, -8. -10, and -12 dB, respectively, are shown. It can be observed that as the SNR level increases, the detector's performance improves (higher sensitivity, P_D and higher specificity $1 - P_{FA}$).

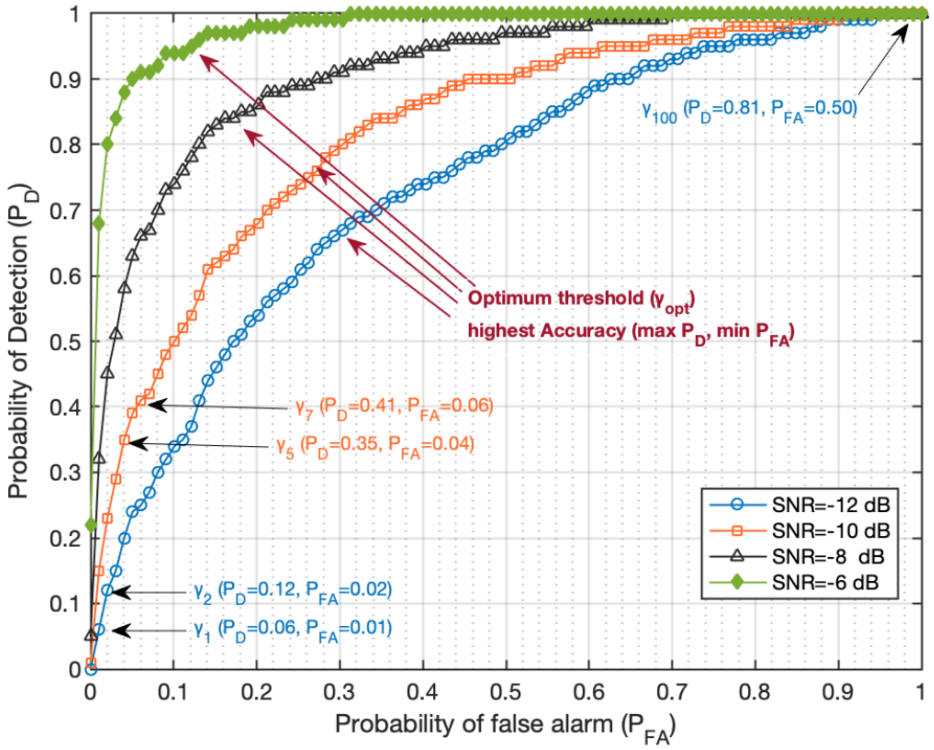


Figure 27: **ROC Curves for QPSK under diverse SNR conditions.**

In Figure 28, it can be observed that for each SNR level, a specific γ value for R^2 is identified. As the SNR level increases, the threshold γ tends to be greater, allowing the model to use a more discriminatory decision threshold which lowers the P_{FA} while keeping a high P_D and thus increasing the overall accuracy of the algorithm.

In contrast, for low SNR values, a low accuracy is obtained. Accuracy values close to 0.5 represent a random guessing decision. It is difficult for the detection scheme to

distinguish between noise and modulated signal samples. As a numerical example, for the test dataset corresponding to the QPSK signal and SNR= -10 dB, the SS algorithm reports an accuracy of 0.74 for $\gamma = 0.7463$. Out of 1000 noise-signal samples, all of them are classified as normal. On the other hand, out of 1000 modulated signal samples, 740 are classified as outliers (or user signals), whereas the remaining 260 are classified as normal (false negatives). Also, it is observed little difference among the accuracy performance results obtained for the three modulated signals (QPSK, 16QAM, and 32QAM).

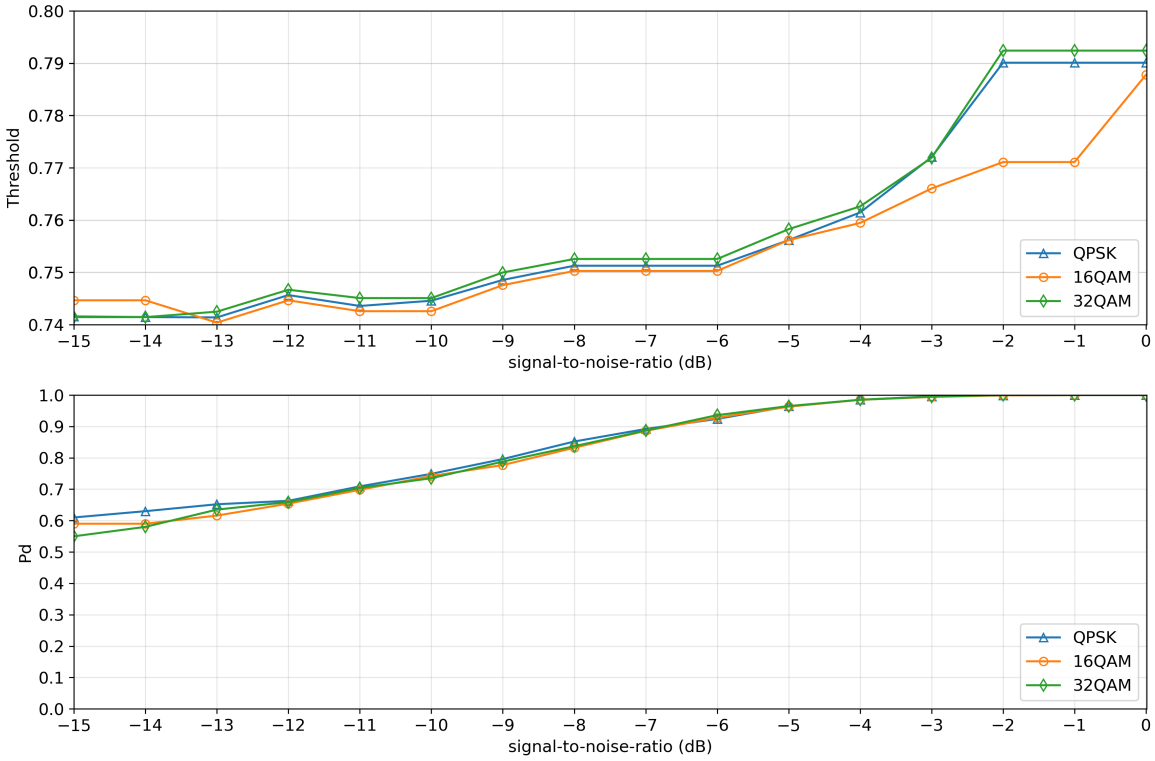


Figure 28: **Accuracy and optimal discrimination threshold vs SNR for R^2 based detection.**

The ROC Area Under the Curve (AUC) shown in Figure 29 is an evaluation metric to determine the degree of separability between P_D and P_{FA}. Unlike the ROC curve, in which both P_D and P_{FA} are threshold functions, the AUC is not directly related to

the threshold as it is averaged out over an entire range of thresholds. Therefore, the AUC metric depends on SNR for all possible decision thresholds. The higher the AUC score is, the better the model is.

Similar to SS based on energy detection, CAE AD-based sensing does not have prior knowledge of the primary signals' features, so it could be categorized as a blind detector. In Figure 29, a comparison of the performance (measured in terms of AUC) of the CAE-AD-based SS and the energy detector is shown. It is observed that the implemented scheme outperforms energy detection for any SNR level and type of modulated signal evaluated. For propagation conditions at SNR=-5 dB, the CAE-AD-based SS proposal can ideally separate normal and anomaly samples (i.e. absence/presence of primary signal for AUC = 1). In contrast, the ED can only achieve this for propagation environments at SNR levels greater than 0 dB.

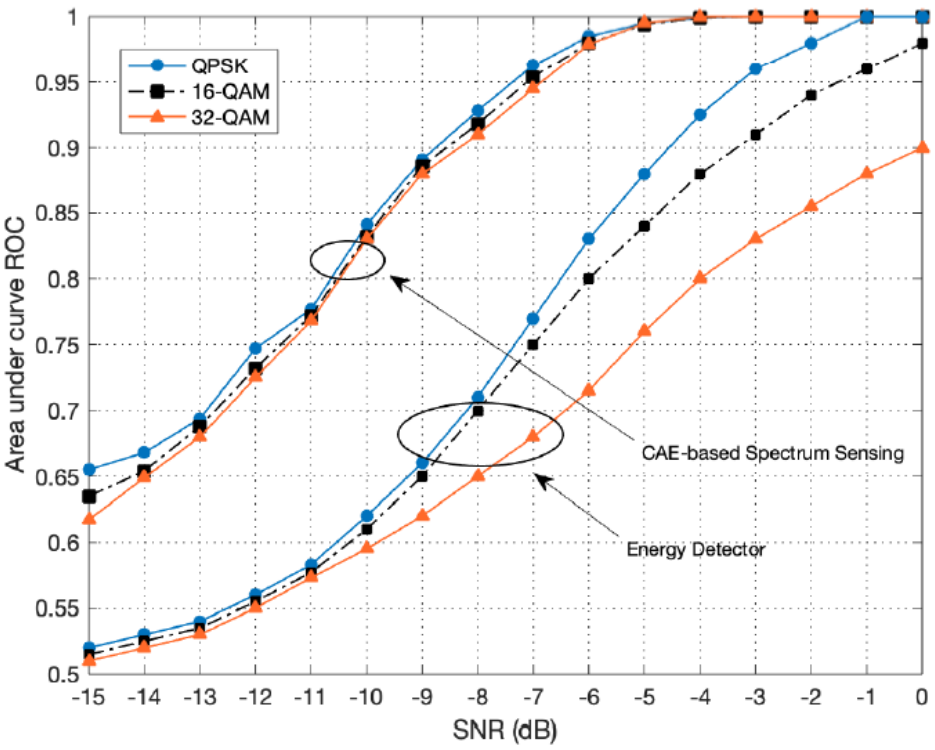


Figure 29: **AUC vs SNR for the conventional energy and CAE-AD based SS algorithms.**

Figure 30 depicts the AUC for each signal within the Deepsig dataset, utilizing MSE, MAE, and R^2 anomaly scores. The ideal detection conditions are reached at 0 dB for MSE and MAE detection for all signals except BPSK which is only reached after the 5 dBs. In the frequency domain, the ideal conditions are reached at -4 dB, except for FM signals where it is reached at around -6 dB, while BPSK detection shows a similar underperformance seen in time domain sensing. R^2 detection in the time domain offers the best overall results where all curves tend to be closer to the upper-left corner of the plot. At -6 dB nearly all signals reach an AUC close to 1, until reaching an ideal detection potentially below -4 dB, however similar to MSE and MAE detection, we also see a lower detection rate for BPSK signal. R^2 detection in the frequency domain is degraded for all modulations, where the FM signal remains below 0.9 for all SNR conditions.

6.2 RNN-based anomaly detection evaluation

This section evaluates the LSTM-AD model for SS. Due to the length of the required samples for the detection process, the number of samples utilized for each result was lowered to 500. Moreover, the Deepsig dataset consists of fixed samples of 1024 amplitudes, which are not long enough to make the conversion process obtain a sequence of entropy values. However, the obtained results are enough to get an understanding of the model behavior for spectrum sensing.

6.2.1 Constant false alarm sensing evaluation

The SNR curve of the LSTM-based AD model for signal modulations QPSK, 16QAM, and 32QAM is presented in Figure 31. The LSTM-based AD offers a similar per-

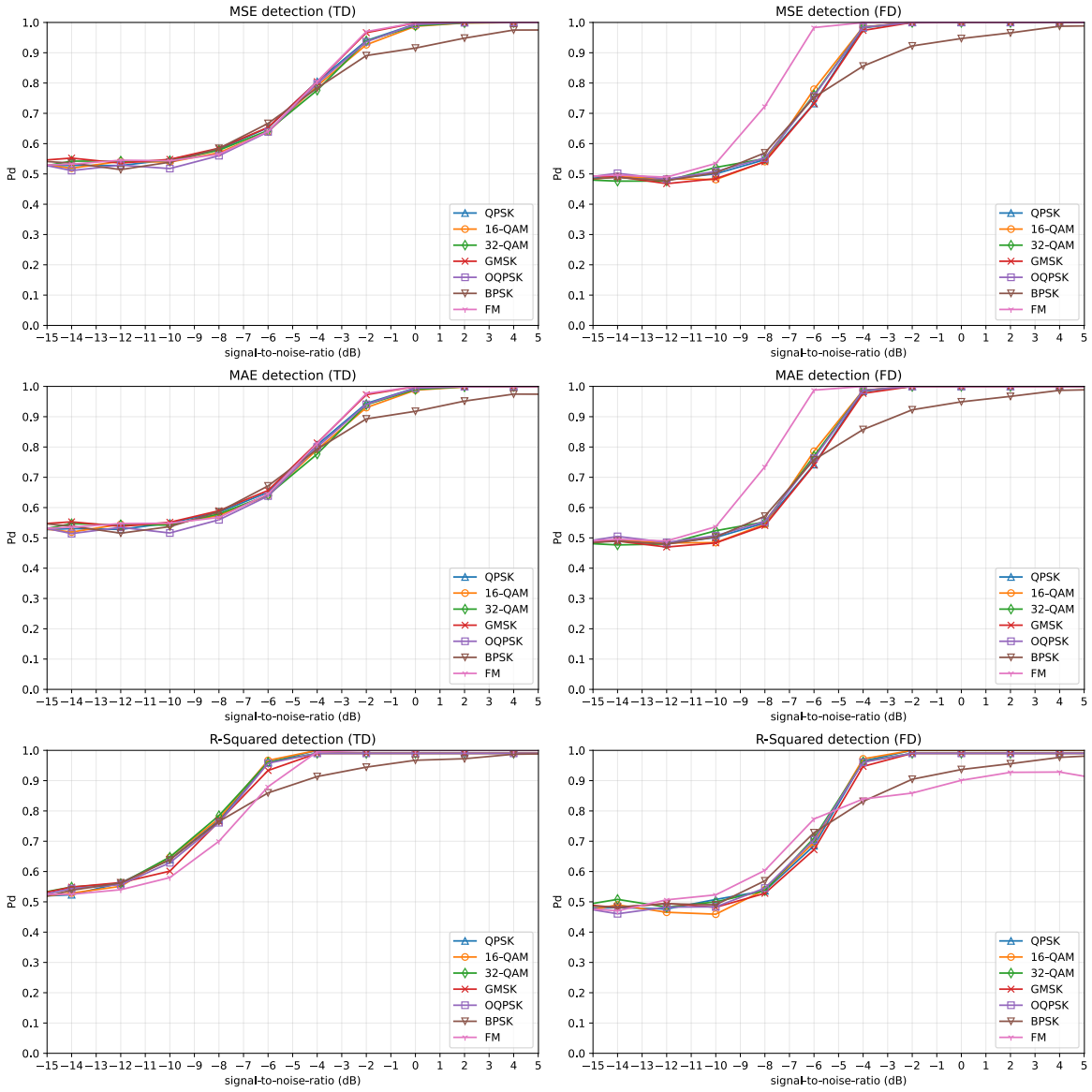


Figure 30: **AUC per SNR conditions for the Deepsig dataset for each utilized anomaly score considering time and frequency domain detection.**

formance as the CAE-based AD for detection based on R^2 anomaly scores, however, the LSTM has a different behavior for each of the modulations, displaying a weaker generalization capability.

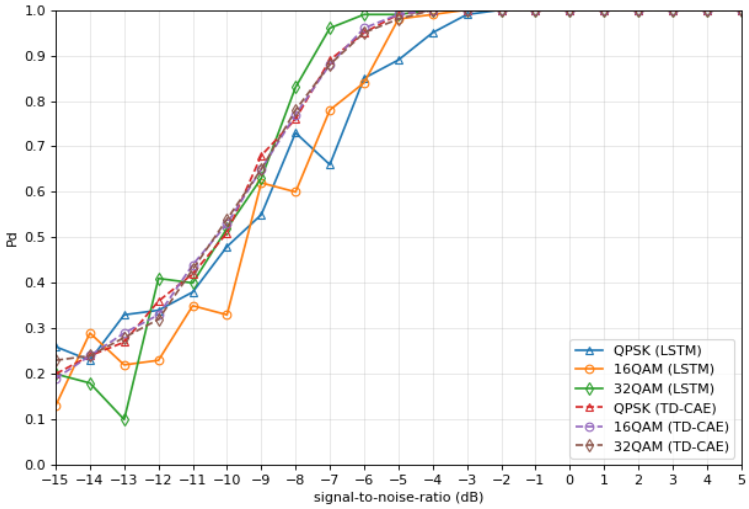


Figure 31: **Comparison between the detection performance of the LSTM-based vs the CAE-based anomaly detectors for spectrum sensing.**

The P_{FA} curves for diverse noise sources are presented in Figure 32. The P_{FA}s for the vacant channel noise tend to be higher than expected, while the simulated and SMU200A generated noise tend to be closer to the reference line, although when compared to the CAE results (Figure 23) we see that R^2 detection also presents an unpredictable behavior for this test.

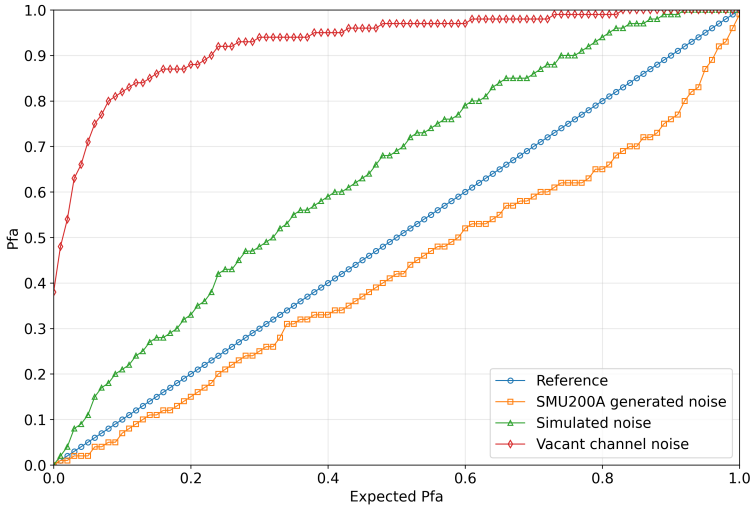


Figure 32: P_{FA} **curves for diverse noise sources.**

6.2.2 Model Accuracy Evaluation

Similar to the ROC curves presented for the CAE-based AD model, the ROC curves in 33 compare the result of the P_D and P_{FA} of the LSTM-based AD model for a diversity of thresholds γ. A higher SNR improves the performance of the detector, showing a higher P_D while preserving a lower P_{FA}. When comparing these results to the CAE, it can be seen that the LSTM ROC curves tend to be further away from the ideal point (where $= P_D = 1.0$ and $P_{FA} = 0$), offering a lower accuracy.

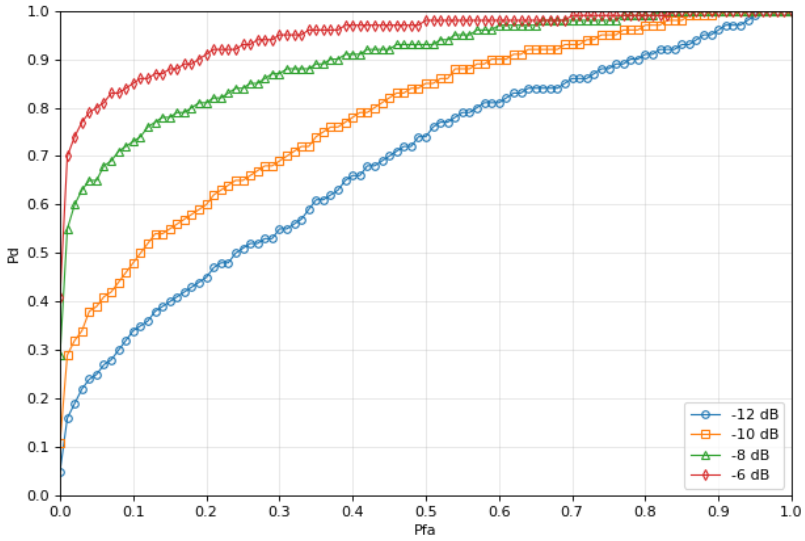
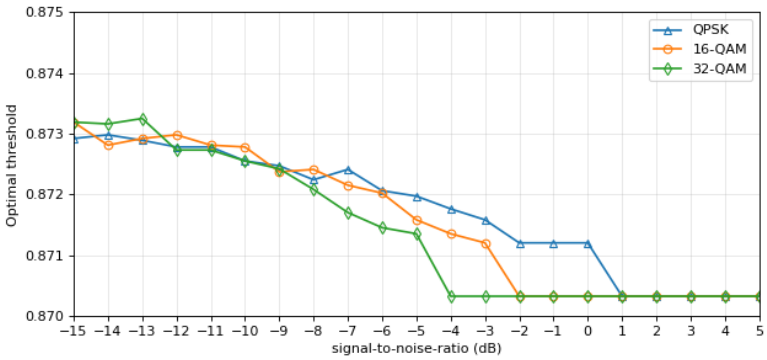
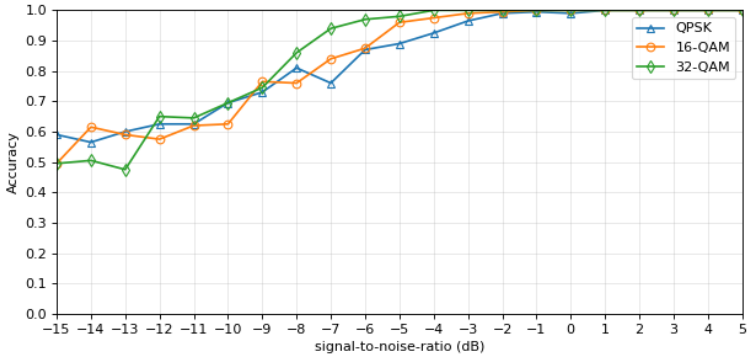


Figure 33: **LSTM AD ROC curves for a QPSK modulated signal under diverse SNR conditions.**

Accuracy and the optimal threshold are presented in figure 34. From here it can be seen in (I) that the γ value for entropy decreases as the SNR conditions increase, allowing the model to become more discriminatory towards the user signals. However, the measured entropy values present a greater variance, offering a more unpredictable detection for the model, as was reflected on the SNR curve in Figure 31, as well as in the optimum accuracy curves (II)



(I)



(II)

Figure 34: **Optimal entropy threshold and accuracy per SNR condition.**

The AUC of the LSTM AD model is presented in figure 35 where a comparison is made with the CAE-AD model. The CAE presents a more constant performance for all modulations while offering a better AUC than the LSTM model.

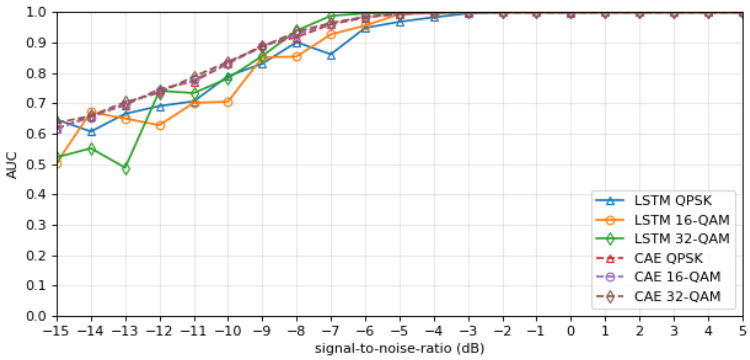


Figure 35: **AUC for modulations QPSK, 16QAM and 32QAM.**

Chapter 7

Conclusion and future work

7.1 Thesis summary

The evolution of cellular networks is becoming one of the major drivers in the information and communications technology (ICT) industry. The Beyond Fifth-Generation (B5G) mobile networks are expected to support extremely high data rates and ultra-low latency. To efficiently provide high-quality services, the B5G wireless networks must adopt more new technologies, including densification of cells and massive multiple-input multiple-output (MIMO), cognitive radio (CR), and spectrum-sharing. However, the static allocation of the radio spectrum obstacles the flexibility to attend to the demand of emerging technology and new use cases in the field of telecommunications. Dynamic spectrum allocation enabled by CR stands as a promising solution to make more efficient use of the spectrum, where users can transmit without band restrictions while at the same time preserving the quality of service of CR technology has been highlighted as a possible spectrum-management technique that successfully tackles the bandwidth dilemma in legacy licensed wireless communication networks by offering opportunistic, on-demand connection. In CR-based networks, there are two types of users, namely the primary (licensed) users (PU) and the unlicensed users, namely the secondary users (SU). While PUs have the priority to access the spectrum, the SUs aim to enable opportunistic use of the licensed spectrum. CR is not only configurable but also aware of and adaptable to its operating radio environment to maximize spectrum utilization

while protecting PU's performance. For this to be possible, reliable spectrum sensing is required, meaning it needs to be able to assert the state of the observed channel accurately. Spectrum sensing means gaining awareness of real-time spectrum utilization and the presence of PU. Spectrum sensing technology is a requirement of CR technology. For unknown spectrum situations, users in the mobile network need to sense the spectrum and find free frequency bands for utilization. Reducing interference to higher priority users, such as in the typical CR scenario, SU opportunistically accesses the licensed spectrum currently not used by the PU. An error in the spectrum sensing result represents false alarms or miss-detections, causing SU to lose access to the spectrum and reduce the network capacity. Miss-detection refers to SU missing the PU presence due to noise and channel fading, causing interference with PU. Therefore, avoiding miss detection and improving the sensing accuracy are the primary purposes of spectrum sensing technology. Several spectrum sensing schemes have been proposed in the literature. Each sensing scheme provides a different trade-off between required sensing time as well as computational and memory complexities.

In recent years, academia has also begun to use artificial intelligence (AI) technology for wireless signal recognition. Also, we can find various proposals that aim to improve the reliability of the prediction of channel occupancy. For example, Neural network-based spectrum sensing is applied for classifying communication signals. This close-to-optimal strategy to classify data is learned from a large amount of sample data via the back-propagation algorithm. Unsupervised learning method like reinforcement learning (RL) has been applied to most cognitive radio schemes, such as dynamic channel selection, channel sensing, the channel features prediction, channel impulse response modeling, channel parameter estimation, and scenario classification. The arrival of the

era of big data is driving the development of AI-based cognitive technology. It will better solve the spectrum shortage problem. Based on the large amount of raw data, such as the real-time spectrum utilization and devices' geographic locations, the transmitted data content can be analyzed to improve the spectrum sensing capability of network users to reduce the interference caused by the spectrum competition while increasing the opportunities for data sharing.

Model-free approaches, like deep learning (DL), have been applied to solve complicated problems in wireless communication systems, including the occupancy of frequency bands. Signal detection is modeled as an anomaly detection problem. Anomaly detection (also known as outlier detection or one-class classification) is finding patterns in data that do not conform to a model of normal behavior. Errors in the data can cause anomalies but sometimes indicate a new, previously unknown, underlying process. They allow the detection of financial fraud, network intrusions, unusual traffic situations, and other rare events. Unsupervised anomaly detection using deep learning is proven very effective. The approaches for training the neural network for one-class classification are classified into generative and discriminative approaches. Generative approaches use generative frameworks such as autoencoders (as feature extractors) or Generative Adversarial Networks (GAN) for one-class classification. The general structure of an autoencoder is based on mapping an input vector to an output vector (a reconstruction) through an internal representation. It can be trained using only the normal class. However, the absence of data from the negative class(es) makes the one-class classification problem difficult because the abnormal class data is unavailable in most situations. Discriminative approaches for one-class classification utilize an external reference dataset as the negative class to train a deep network using a novel loss function. This approach for one-class classification has not been well explored in the

literature.

This thesis utilizes an anomaly detection approach where an AWGN-trained model deems any user signal as an anomaly. In contrast, in the context of signal detection, this is analogous to user presence. Moreover, accuracy, a common ML evaluation that describes the model assertion capabilities, is utilized for analyzing and optimizing the performance of the implemented models, in addition to the more traditional CFAR threshold selection. A convolutional autoencoder (CAE) is put forward to deal with signal detection (or anomaly detection) of the spectrum in wireless communication. AWGN raw data are used to train the auto-encoders network, which acts as a one-class classifier. Our approach relies on the reconstruction error of the network to judge whether the signal is present in the frequency band of interest.

7.2 Findings

Spectrum sensing has been characterized in the literature as a classification problem.

Both CAE and LSTM models demonstrated that anomaly detection could be utilized for signal detection through sequential and non-sequential data, where both models showed an improved detection rate over the conventional energy detector. Although both performed similarly in accuracy and detection rate per SNR conditions, the LSTM model has a higher computational complexity due to the entropy measurement steps. In contrast, the CAE model only needs the amplitude values and significantly fewer data to reach these results. The larger data requirement of the LSTM model also prevented the Deepsig dataset from being tested due to the samples consisting of only 1024 amplitude values. The LSTM required $1024 \times 65 = 66,560$ amplitudes in this

work. Although the variance in the results could be reduced and potentially increase the detection rate of the LSTM model if the signal length is extended for each entropy calculation, this would only further increase the already larger data required for this model. However, the experimentation was more emphasized in implementing the CAE to test the proposal of approaching spectrum sensing as an anomaly detection problem. This led to the model having a more exhaustive optimization phase.

While CFAR restricts the model to a fixed specificity at the expense of sensitivity, accuracy optimization maximizes the model's performance by minimizing the classification error through a more balanced P_D and P_{FA} for each SNR condition. Additionally, the AUC helps us measure the detection performance of the models in a summarized manner by including the entire range of thresholds under the ROC curves.

7.3 Future Work

Low computational complexity is essential in the spectrum sensing task within a cognitive radio network for vacant channel reallocation to minimize any possible interference to the primary user and interrupt any ongoing secondary user communication. Therefore the AD model must still be analyzed in terms of complexity and compared against the conventional energy detector. Additionally, the CAE-AD demonstrated an improvement in terms of detection when longer sample lengths were utilized. Therefore, the trade-off between detection rate against complexity is another important aspect yet to be studied for this strategy.

The AD-based SS could be improved through other promising NN architectures in the literature for sequential and non-sequential data, such as utilizing Variational Autoencoders and Generative Adversarial Networks. For non-sequential data, various

LSTM architectures can be implemented, including the addition of convolution and the combination of both CAE and LSTM, as well as Gated Recurrent Units (GRU). Moreover, an ML-calculated anomaly score can also be explored.

References

[1] W. Cheng, X. You, X. Gao, and X. Zhu, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[2] "Cisco annual internet report (2018–2023) white paper," Jan 2022. [online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executiveperspectives/annual-internet-report/white-paper-c11-741490.html.

[3] M. Ali, M. N. Yasir, D. M. S. Bhatti, and H. Nam, "Optimization of spectrum utilization efficiency in cognitive radio networks," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 426–430, 2023.

[4] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[5] M. M. Buddhikot, "Understanding dynamic spectrum access: Models,taxonomy and challenges," in *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 649–663, 2007.

[6] R. K. Saha, "Nationwide spectrum sharing of mobile network operators with indoor small cells," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–2, 2019.

[7] T. Sheng, W. Zhang, W. Ding, J. Sun, and C.-X. Wang, "Dynamic spectrum sharing and aggregation scheme based on deep reinforcement learning," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 290–294, 2022.

[8] M. Parvini, A. H. Zarif, A. Nouruzi, N. Mokari, M. R. Javan, B. Abbasi, A. Ghasemi, and H. Yanikomeroglu, "Spectrum sharing schemes from 4g to 5g and beyond: Protocol flow, regulation, ecosystem, economic," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 464–517, 2023.

[9] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201–220, Feb. 2005.

[10] V. Prithiviraj, B. Sarankumar, A. Kalaiyarasan, P. P. Chandru, and N. N. Singh, "Cyclostationary analysis method of spectrum sensing for cognitive radio," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference On*, pp. 1–5, IEEE, 2011.

[11] S. Kapoor, S. Rao, and G. Singh, "Opportunistic spectrum sensing by employing matched filter in cognitive radio network," in *Communication Systems and Network Technologies (CSNT), 2011 International Conference On*, pp. 580–583, IEEE, 2011.

[12] N. Pillay and H. Xu, "Blind eigenvalue-based spectrum sensing for cognitive radio networks," *IET communications*, vol. 6, no. 11, pp. 1388–1396, 2012.

[13] S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Eigenvalue-based sensing and snr estimation for cognitive radio in presence of noise correlation," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3671–3684, 2013.

[14] Y. Zeng and Y.-C. Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," *IEEE transactions on communications*, vol. 57, no. 6, 2009.

[15] P. Pawelczak, K. Nolan, L. Doyle, S. W. Oh, and D. Cabric, "Cognitive radio: Ten years of experimentation and development," *IEEE Communications Magazine*, vol. 49, pp. 90–100, Mar. 2011.

[16] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, pp. 523–531, Apr. 1967.

[17] A. Gorcin, K. A. Qaraqe, H. Celebi, and H. Arslan, "An adaptive threshold method for spectrum sensing in multi-channel cognitive radio networks," in *2010 17th International Conference on Telecommunications*, pp. 425–429, Apr. 2010.

[18] H. Sakran and M. Shokair, "Hard and softened combination for cooperative spectrum sensing over imperfect channels in cognitive radio networks," *Telecommunication Systems*, vol. 52, no. 1, pp. 61–71, 2013.

[19] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors*, vol. 19, no. 1, 2019.

[20] Z. Li, W. Wu, X. Liu, and P. Qi, "Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks," *IET Communications*, vol. 12, no. 19, pp. 2485–2492, 2018.

[21] H. Xiao, X. Zhou, and Y. Tian, "Research on wireless spectrum sensing technology based on machine learning," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 472–479, Springer, 2018.

[22] V. Raj, I. Dias, T. Tholeti, and S. Kalyani, "Spectrum access in cognitive radio using a two-stage reinforcement learning approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 20–34, 2018.

[23] X.-L. Huang, Y. Xu, and E. Liu, "Historical sensing data mining in cognitive radio networks," *Advances in Intelligent Systems and Computing*, vol. 277, pp. 549–557, 01 2014.

[24] P. Baltiiski, I. Iliev, B. Kehaiov, V. Poulkov, and T. Cooklev, "Long-term spectrum monitoring with big data analysis and machine learning for cloud-based radio access networks," *Wireless Personal Communications*, vol. 87, no. 3, pp. 815–835, 2016.

[25] B. Dey, A. Hossain, R. Dey, and R. Bera, "Integrated blind signal separation and neural network based energy detector architecture," *Wireless Personal Communications*, vol. 106, no. 4, pp. 2315–2333, 2019.

[26] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.

[27] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[28] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, 07 2009.

[29] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, "Novelty detection for the identification of masses in mammograms," 1995.

[30] J. A. Quinn and C. K. Williams, "Known unknowns: Novelty detection in condition monitoring," in *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 1–6, Springer, 2007.

[31] L. Clifton, D. A. Clifton, P. J. Watkinson, and L. Tarassenko, "Identification of patient deterioration in vital-sign data using one-class support vector machines," in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 125–131, 2011.

[32] C. Surace and K. Worden, "Novelty detection in a changing environment: A negative selection approach," *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 1114–1128, 2010.

[33] X. Zhou, J. Xiong, X. Zhang, X. Liu, and J. Wei, "A radio anomaly detection algorithm based on modified generative adversarial network," *IEEE Wireless Communications Letters*, 2021.

[34] S. Yin, S. Li, and J. Yin, "Temporal-spectral data mining in anomaly detection for spectrum monitoring," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–5, IEEE, 2009.

[35] J. Xu, Y. Tian, S. Yuan, and N. Liu, "Noise attention based spectrum anomaly detection method for unauthorized bands," *arXiv preprint arXiv:2104.08517*, 2021.

[36] L. Zhang, C. X. Huang, H. Tang, J. J. Yang, and M. Huang, "Dtv radio spectrum anomaly detection based on an improved gan," in *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, pp. 1–4, IEEE.

[37] A. Toma, A. Krayani, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Deep learning for spectrum anomaly detection in cognitive mmwave radios," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–7, IEEE, 2020.

[38] D. Sun, S. Lu, and W. Wang, "Caae: A novel wireless spectrum anomaly detection method with multiple scoring criterion," in *2021 28th International Conference on Telecommunications (ICT)*, pp. 1–5, IEEE, 2021.

[39] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[40] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2013.

[41] B. Bhavana, S. L. Sabat, S. Namburu, and T. Panigrahi, "Energy detector for spectrum sensing using robust statistics in non-gaussian noise environment," in *2023 15th International Conference on COMmunication Systems NETworkS (COMSNETS)*, pp. 414–418, 2023.

[42] J. Chen, A. Gibson, and J. Zafar, "Cyclostationary spectrum detection in cognitive radios," in *2008 IET Seminar on Cognitive Radio and Software Defined Radios: Technologies and Techniques*, pp. 1–5, 2008.

[43] A. Brito, P. Sebastião, and F. J. Velez, "Hybrid matched filter detection spectrum sensing," *IEEE Access*, vol. 9, pp. 165504–165516, 2021.

[44] A. Parvathy and G. Narayanan, "Comparative study of energy detection and matched filter based spectrum sensing techniques," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 147–153, 2020.

[45] G. Prieto, G. Andrade, D. M. Martínez, and G. Galaviz, "On the evaluation of an entropy-based spectrum sensing strategy applied to cognitive radio networks," *IEEE Access*, vol. 6, pp. 64828–64835, 2018.

[46] S. V. Nagaraj, "Entropy-based spectrum sensing in cognitive radio," *Signal Processing*, vol. 89, no. 2, pp. 174–180, 2009.

[47] J. Nikonowicz, P. Kubczak, and Matuszewski, "Hybrid detection based on energy and entropy analysis as a novel approach for spectrum sensing," in *2016 International Conference on Signals and Electronic Systems (ICSES)*, pp. 206–211, 2016.

[48] Y. Zeng and Y.-C. Liang, "Spectrum-sensing algorithms for cognitive radio based on statistical covariances," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1804–1815, 2009.

[49] C. Charan and R. Pandey, "Double threshold based spectrum sensing technique using sample covariance matrix for cognitive radio networks," in *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pp. 150–153, 2017.

[50] S. A. Jain and M. M. Deshmukh, "Performance analysis of energy and eigenvalue based detection for spectrum sensing in cognitive radio network," in *2015 International Conference on Pervasive Computing (ICPC)*, pp. 1–5, 2015.

[51] L. L. R. Centeno, F. C. C. De Castro, M. C. F. De Castro, C. Müller, and S. M. Ribeiro, "Cognitive radio signal classification based on subspace decomposition and rbf neural networks," *Wireless networks*, vol. 24, no. 3, pp. 821–831, 2018.

[52] R. B. Chaurasiya and R. Shrestha, "Hardware-efficient and fast sensing-time maximum-minimum-eigenvalue-based spectrum sensor for cognitive radio network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 11, pp. 4448–4461, 2019.

[53] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Communications*, vol. 17, no. 2, pp. 138–148, 2020.

[54] D. Chew and A. B. Cooper, "Spectrum sensing in interference and noise using deep learning," in *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, IEEE, 2020.

[55] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Communications Letters*, vol. 9, no. 1, pp. 38–41, 2020.

[56] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 718–727, 2019.

[57] D. Ke, Z. Huang, X. Wang, and X. Li, "Blind detection techniques for non-cooperative communication signals based on deep learning," *IEEE Access*, vol. 7, pp. 89218–89225, 2019.

[58] J. Xie, J. Fang, C. Liu, and X. Li, "Deep learning-based spectrum sensing in cognitive radio: A cnn-lstm approach," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2196–2200, 2020.

[59] D. Janu, S. Kumar, and K. Singh, "A graph convolution network based adaptive cooperative spectrum sensing in cognitive radio network," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2269–2279, 2023.

[60] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3005–3009, 2019.

[61] S. Kim, "Multi-agent learning and bargaining scheme for cooperative spectrum sharing process," *IEEE Access*, vol. 11, pp. 47863–47872, 2023.

[62] B. P. J. J. S. J. J. Thudumu, Srikanth, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, no. 1, 2020.

[63] A. Sinha, S. Padhi, and S. Shikalgar, "A survey and analysis of crowd anomaly detection techniques," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 1500–1504, 2021.

[64] C. Lee, J. Kim, and S.-j. Kang, "Semi-supervised anomaly detection with reinforcement learning," in *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 933–936, 2022.

[65] H.-S. Wu, "A survey of research on anomaly detection for time series," in *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 426–431, 2016.

[66] A. Chirayil, R. Maharjan, and C.-S. Wu, "Survey on anomaly detection in wireless sensor networks (wsns)," in *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 150–157, 2019.

[67] S. Chang, B. Du, and L. Zhang, "A sparse autoencoder based hyperspectral anomaly detection algorihtm using residual of reconstruction error," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5488–5491, 2019.

[68] D. Zhu, B. Du, Y. Dong, and L. Zhang, "Spatial–spectral joint reconstruction with interband correlation for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[69] D. Sun, S. Lu, and W. Wang, "Caae: A novel wireless spectrum anomaly detection method with multiple scoring criterion," in *2021 28th International Conference on Telecommunications (ICT)*, pp. 1–5, 2021.

[70] H. Zhang, J. Yang, J. Chen, and Y. Gao, "Spectrum usage anomaly detection from sub-sampled data stream via deep neural network," *Journal of Communications and Information Networks*, vol. 8, no. 1, pp. 13–23, 2023.

[71] A. H. Celdrán, P. M. S. Sánchez, G. Bovet, G. M. Pérez, and B. Stiller, "Cyberspec: Behavioral fingerprinting for intelligent attacks detection on crowdsensing spectrum sensors," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2023.

[72] X. Jin, B. Yan, N. Sun, F. Iturbide-Sanchez, and Y. Chen, "A novel detection algorithm for impulse spike interferences in cross-track infrared sounder short-wave spectral radiance," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9089–9105, 2021.

[73] R. Tandra and A. Sahai, "Snr walls for signal detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 4–17, 2008.

[74] D. Berrar, "Performance measures for binary classification," 2019.

[75] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 4–21, 2017.

[76] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[77] A. Gupta and A. Joshi, "Speech recognition using artificial neural network," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0068–0071, 2018.

[78] M. Sundermeyer, I. Oparin, J.-L. Gauvain, B. Freiberg, R. Schlüter, and H. Ney, "Comparison of feedforward and recurrent neural network language models," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8430–8434, 2013.

[79] Z. Wang, Q. Shao, C. Wang, and Q. Zhang, "Automatic parking trajectory planning based on recurrent neural network," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 1–4, 2018.

[80] S. H. Park, H. J. Park, and Y.-J. Choi, "Rnn-based prediction for network intrusion detection," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 572–574, 2020.

[81] A. Nanduri and L. Sherry, "Anomaly detection in aircraft data using recurrent neural networks (rnn)," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, pp. 5C2–1, Ieee, 2016.

[82] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[83] C.-W. Tien, T.-Y. Huang, P.-C. Chen, and J.-H. Wang, "Using autoencoders for anomaly detection and transfer learning in iot," *Computers*, vol. 10, no. 7, p. 88, 2021.

[84] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, vol. 105, pp. 13–22, 2018.

[85] H. A. Perlin and H. S. Lopes, "Extracting human attributes using a convolutional neural network approach," *Pattern Recognition Letters*, vol. 68, pp. 250–259, 2015. Special Issue on "Soft Biometrics".

[86] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[87] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016.

[88] D. Han, G. C. Sobabe, C. Zhang, X. Bai, Z. Wang, S. Liu, and B. Guo, "Spectrum sensing for cognitive radio based on convolution neural network," in *2017 10th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)*, pp. 1–6, IEEE, 2017.

[89] Y. Xu, D. Li, Z. Wang, Q. Guo, and W. Xiang, "A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals," *Wireless Networks*, vol. 25, no. 7, pp. 3735–3746, 2019.

[90] T. B. Duman, B. Bayram, and G. İnce, "Acoustic anomaly detection using convolutional autoencoders in industrial processes," in *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, pp. 432–442, Springer, 2019.

[91] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-d convolutional neural networks for signal processing applications," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8360–8364, IEEE, 2019.

[92] S. Kiranyaz, T. Ince, and M. Gabbouj, "Personalized monitoring and advance warning system for cardiac arrhythmias," *Scientific reports*, vol. 7, no. 1, pp. 1–8, 2017.

[93] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for mmc using 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8760–8771, 2018.

[94] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[95] J.-H. Park and Y. Kim, "Convolutional autoencoder for compressive symbol detection," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 986–988, IEEE, 2018.

[96] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media, 2019.

[97] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval.," in *ESANN*, vol. 1, p. 2, Citeseer, 2011.

[98] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Adaptive computation and machine learning, MIT Press, 2016.

[99] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[100] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[101] S. V. Nagaraj, "Entropy-based spectrum sensing in cognitive radio," *Signal Processing*, vol. 89, no. 2, pp. 174–180, 2009.

[102] Y. L. Zhang, Q. Y. Zhang, and T. Melodia, "A frequency-domain entropy-based detector for robust spectrum sensing in cognitive radio networks," *IEEE Communications Letters*, vol. 14, no. 6, pp. 533–535, 2010.

[103] 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.

[104] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[105] A. A. Khan, M. H. Rehmani, and A. Rachedi, "Cognitive-radio-based internet of things: Applications, architectures, spectrum related functionalities, and future research directions," *IEEE wireless communications*, vol. 24, no. 3, pp. 17–25, 2017.

[106] D. Tarek, A. Benslimane, M. Darwish, and A. M. Kotb, "Survey on spectrum sharing/allocation for cognitive radio networks internet of things," *Egyptian Informatics Journal*, 2020.

[107] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[108] N. Widiastuti, "Convolution neural network for text mining and natural language processing," in *IOP Conference Series: Materials Science and Engineering*, vol. 662, p. 052010, IOP Publishing, 2019.

[109] B. Agarwal, H. Ramampiaro, H. Langseth, and M. Ruocco, "A deep network model for paraphrase detection in short text messages," *Information Processing & Management*, vol. 54, no. 6, pp. 922–937, 2018.

[110] S. Zhang, E. Grave, E. Sklar, and N. Elhadad, "Longitudinal analysis of discussion topics in an online breast cancer community using convolutional neural networks," *Journal of biomedical informatics*, vol. 69, pp. 1–9, 2017.

[111] T. Tran and R. Kavuluru, "Predicting mental conditions based on "history of present illness" in psychiatric notes with deep neural networks," *Journal of biomedical informatics*, vol. 75, pp. S138–S148, 2017.

[112] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.

[113] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ecg classification by 1-d convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2015.

[114] J. T. Ruiz, J. D. B. Pérez, and J. R. B. Blázquez, "Arrhythmia detection using convolutional neural models," in *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 120–127, Springer, 2018.

[115] S. Nallagonda, S. K. Bandari, S. D. Roy, and S. Kundu, "Performance of cooperative spectrum sensing with soft data fusion schemes in fading channels," in *2013 Annual IEEE India Conference (INDICON)*, pp. 1–6, IEEE, 2013.

[116] Y. Bengio, R. De Mori, G. Flammia, and R. Kompe, "Global optimization of a neural network-hidden markov model hybrid," *IEEE transactions on Neural Networks*, vol. 3, no. 2, pp. 252–259, 1992.

[117] C. Xiaojun, L. Yin, and D. Jingya, "Approximate entropy detection spectrum sensing in cognitive radio," in *2014 IEEE International Conference on Computer and Information Technology*, pp. 865–868, 2014.

[118] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," vol. 2, pp. 1097–1105, 2012. cited By 54407.

[119] S. Liu, L. J. Greenstein, W. Trappe, and Y. Chen, "Detecting anomalous spectrum usage in dynamic spectrum access networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 831–844, 2012.