FEDERAL UNIVERSITY OF AMAZONAS
FACULTY OF TECHNOLOGY
POSTGRADUATE PROGRAM IN ELECTRICAL ENGINEERING

SENSING, ESTIMATION, AND SECURITY OF THE FREQUENCY SPECTRUM USING SHALLOW AND DEEP LEARNING TECHNIQUES

Myke Douglas de Medeiros Valadão

Doctoral Thesis presented to Postgraduate Program in Electrical Engineering at the Federal University of Amazonas, as part of the requirements necessary to obtain the degree of Doctor in Electrical Engineering.

Advisor: Dr. Waldir Sabino da Silva Júnior

Manaus
October of 2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Valadão, Myke Douglas de Medeiros
V136s Sensing, Estimation, and Security of the Frequency Spectrum Using Shallow and Deep Learning Techniques / Myke Douglas de Medeiros Valadão . 2024
107 f.: il. color; 31 cm.

Orientador: Waldir Sabino da Silva Júnior
Tese (Doutorado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Spectrum Sensing. 2. Machine Learning. 3. Deep Learning. 4. Generative Adversarial Network. 5. Transformer. I. Silva Júnior, Waldir Sabino da. II. Universidade Federal do Amazonas III. Título

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

MYKE DOUGLAS DE MEDEIROS VALADÃO

SENSING, ESTIMATION AND SECURITY OF THE SPECTRUM OF FREQUENCY BASED ON DEEP AND MACHINE LEARNING TECHNIQUES

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 07 de outubro de 2024.

BANCA EXAMINADORA
Prof. Dr. Waldir Sabino da Silva Júnior- Presidente
Profª. Dra. Eulanda Miranda dos Santos - Membro Titular 1 - Externo
Prof. Dr. Celso Barbosa Carvalho - Membro Titular 2 - Interno
Prof. Dr. Carlos Augusto de Moraes Cruz - Membro Titular 3 - Interno
Prof. Dr. Tadeu Nagashima Ferreira- Membro Titular 4 - Externo
Documento assinado eletronicamente

Manaus, 03 de outubro de 2024.

Documento assinado eletronicamente por **Tadeu Nagashima Ferreira**, **Usuário Externo**, em 11/10/2024, às 08:08, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Eulanda Miranda dos Santos**, **Professor do Magistério Superior**, em 11/10/2024, às 10:57, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Waldir Sabino da Silva Júnior**, **Professor do Magistério Superior**, em 11/11/2024, às 22:00, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Celso Barbosa Carvalho**, **Professor do Magistério Superior**, em 12/11/2024, às 10:08, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Carlos Augusto de Moraes Cruz**, **Coordenador**, em 25/11/2024, às 20:55, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2265536** e o código CRC **0184773B**.

Av. Octávio Hamilton Botelho Mourão - Bairro Coroado 1 Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181
CEP 69080-900 Manaus/AM - mestrado_engeletrica@ufam.edu.br

Referência: Processo nº 23105.039199/2024-79 SEI nº 2265536

Criado por 31183646291, versão 2 por 31183646291 em 03/10/2024 09:55:11.

Acknowledgment

I would like to express my sincere gratitude to everyone who contributed to the completion of this research work and to the realization of this doctoral thesis.

First and foremost, I am deeply thankful to my supervisor, Dr. Waldir Sabino Junior, for their wise guidance, unwavering support, and dedication throughout this process. Their critical insights and guidance were instrumental in shaping this work.

I extend my heartfelt thanks to Dr. André Costa and the Federal University of Uberlândia for providing me with the opportunity to pursue my doctoral studies. Their support and resources have been invaluable in the development of this research.

To my family, Rose, Matheus and Thiago, for their unconditional love, constant encouragement, and emotional support during the highs and lows of this academic journey. Without their support, this achievement would not have been possible.

To my friends, Maurício, Áurea and Camila who shared their experiences and knowledge, making this journey more enriching and rewarding. Your discussions and collaborations were essential to my academic and personal growth.

To Federal University of Amazonas, for providing the research environment and necessary resources for conducting this study. I also thank the funding agency, FAPEAM and SiDi Manaus, for their financial support that enabled this research.

A special thank you goes to my colleagues at SiDi Manaus. Your collaborative spirit, expertise, and dedication have significantly enriched this project. Working alongside such talented individuals has been both inspiring and rewarding.

Lastly, I am grateful to all professors, researchers, and professionals whose work and contributions were crucial to the development of this thesis. Your discoveries and insights have been a constant source of inspiration.

This work is dedicated to all of you, whose support and encouragement have

been indispensable to this achievement.

Abstract of Thesis presented to UFAM as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

SENSING, ESTIMATION, AND SECURITY OF THE FREQUENCY SPECTRUM USING SHALLOW AND DEEP LEARNING TECHNIQUES

Myke Douglas de Medeiros Valadão

Advisor: Dr. Waldir Sabino da Silva Júnior

The frequency spectrum is a limited resource that has experienced a growing demand in recent years, especially with the advent of 5G and 6G technologies. Spectrum sensing, estimation, and security are essential factors for increasing efficiency and flexibility in spectrum utilization, enabling its usage optimization and ensuring security for a larger number of users. Spectrum sensing is crucial for filling spectral holes, relieving congested frequency bands. Estimating spectrum conditions also plays an extremely important role in designing and proposing specific solutions and services for different conditions. Last but not least, with the development of generative artificial intelligence, spectrum security becomes essential for developing measures to mitigate malicious user activities. In this context, this thesis presents research related to these approaches. Experimental results suggest promising prospects for these approaches, implying improvements in efficiency, robustness, and low latency in current communication systems. For instance, in the spectrum sensing approach, the proposed simplified ResNet achieved 98% accuracy under a noise level of -134 dBm/Hz, with a response time below 0.05 seconds, ensuring low latency. For spectrum estimation, the XGBoost and Transformer models achieved the best correlation coefficients for identifying the noise level and the distance between users in a spectrum sensing environment, with values of 0.98 and 0.84, respectively. Lastly, in spectrum security, the proposed generative adversarial network was able to deceive deep cooperative spectrum sensing models in over 98% of cases.

Keywords: Spectrum sensing, Machine Learning, Deep Learning, Generative Adversarial Network, Transformer.

Resumo da Tese apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Doctor em Engenharia Elétrica.

SENSORIAMENTO, ESTIMATIVA E SEGURANÇA DO ESPECTRO DE FREQUÊNCIA UTILIZANDO TÉCNICAS DE APRENDIZADO RASO E PROFUNDO

Myke Douglas de Medeiros Valadão

Orientador: Dr. Waldir Sabino da Silva Júnior

O espectro de frequência é um recurso limitado que tem enfrentado uma crescente demanda nos últimos anos, especialmente com o advento das tecnologias 5G e 6G. Sensoriamento, estimativa e segurança do espectro são fatores essenciais para aumentar a eficiência e a flexibilidade na utilização do espectro, permitindo sua otimização e garantindo a segurança para um maior número de usuários. O sensoriamento do espectro é crucial para preencher lacunas espectrais, aliviando bandas de frequência congestionadas. A estimativa das condições do espectro também desempenha um papel extremamente importante no desenvolvimento e proposição de soluções e serviços específicos para diferentes condições. Por fim, com o desenvolvimento da inteligência artificial generativa, a segurança do espectro torna-se essencial para a criação de medidas que mitiguem atividades maliciosas de usuários. Nesse contexto, esta tese apresenta pesquisas relacionadas a essas abordagens. Os resultados experimentais sugerem perspectivas promissoras para essas abordagens, implicando melhorias na eficiência, robustez e baixa latência nos sistemas de comunicação atuais. Por exemplo, na abordagem de sensoriamento do espectro, a ResNet simplificada proposta alcançou 98% de acurácia em um nível de ruído de -134 dBm/Hz, com um tempo de resposta inferior a 0,05 segundos, garantindo baixa latência. Para estimativa do espectro, os modelos XGBoost e Transformer alcançaram os melhores coeficientes de correlação para identificação do nível de ruído e da distância entre usuários em um ambiente de sensoriamento do espectro, com valores de 0,98 e 0,84, respectivamente. Por fim, na segurança do espectro, a rede adversária generativa proposta foi capaz de enganar modelos de detecção cooperativa profunda em mais de 98% dos casos.

Palavras-chave: Sensoriamento do espectro, Aprendizado de Máquina, Aprendizado Profundo, Rede Generativa Adversária, Transformer.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The frequency spectrum is a finite resource that has become increasingly contested recently due to the rise in demand and new technologies such as 5G/6G [1–7]. The spectrum allocation policy, suggested by the International Telecommunication Union (ITU), has become inefficient, leading to underutilization of some bands and overutilization of others. Cognitive radio senses the frequency spectrum and identifies spectral holes to dynamically allocate secondary users (SUs) in bands that are partially or entirely unoccupied by licensed users, primary users (PUs) [8, 9]. There are two types of spectrum sensing approaches, narrowband [7, 10] and wideband [7, 11], and among these approaches, there are various techniques, such as energy detector [12–14], similarity filter detector [15–17], cyclostationary feature detector [18–20], and more recently, machine learning and deep learning-based detectors [21–23].

In the short-band sensing approach, only one channel is sensed, and the SU can be allocated to the frequency channel of interest if there is no presence of PUs [7]. The most used techniques in this approach are: energy detector [12], similarity filter detector [15], cyclostationary feature detector [18], waveform detector [10], detector based on Wavelet transform [10], and some approaches using artificial neural networks as in [24]. The energy detector is one of the simplest methods to evaluate the channel, where a threshold is set, and if the received signal energy is higher than the threshold, then there is PU presence in the channel; otherwise, there is no PU presence in the channel [10, 12]. In the similarity filter technique, pilot samples captured from the same signal transmitter are compared with the received

signals, and an adaptive threshold is applied to detect the condition of the evaluated channel [7, 15].

In cyclostationary feature detection, cyclostationary information is extracted from the received signals, such as hopping sequence, periodicity, and pulse train [10]. This method is considered robust due to the random nature of the noise signal, hence the absence of cyclical characteristics, which enhances performance even at low signal-to-noise ratios (SNR). Subsequently, a correlation approach determines the presence of PUs in the evaluated channel [18]. In the waveform detector, a priori information of the PU signal is required to correlate this information with the received signal [10]. Finally, the Wavelet detector calculates the power spectrum density (PSD) of the received signal, and the Wavelet transform is then applied to extract unique information from the evaluated channel [7, 10].

In the next generation of communication systems, such as 5G and 6G, high data rates are required. For this purpose, broadband spectrum sensing techniques have been necessary to sense a wide range of bands in the frequency spectrum [7, 11, 25–28]. Among the most popular methods for broadband sensing are Nyquist and sub-Nyquist [7]. In the Nyquist approach, the received signal is sampled by a traditional analog-to-digital converter with a Nyquist sampling rate ($f_s \geq 2f_m$) [28]. However, despite its straightforward structure, this approach demonstrates high sampling rates and high energy costs, making it difficult to apply in experimental scenarios. The sub-Nyquist approach overcomes the challenges presented in the Nyquist method by reducing the sampling rate and then detecting the presence of PUs, or not, with the remaining partial data [7, 28].

Another widely used technique for broadband sensing is compressive spectrum sensing, which is divided into two categories: multi-bit compressive sensing and one-bit compressive sensing [7, 27]. Multi-bit compressive sensing is divided into three stages: sparse representation, measurement, and sparse recovery. The sparse representation process is carried out by some transformations such as Fourier, discrete Fourier, and discrete cosine [29]. Measurement is performed by multiplying the sparse signal by a measurement matrix. Recovery can be classified into three categories: convex relaxation, greedy, and Bayesian [7, 30]. To address the issue of multi-level quantization error, one-bit compressive sensing uses only one quantiza-

tion bit, reducing the need for robust hardware while preserving the measurement signal information.

More recently, machine learning and deep learning techniques have been demonstrating increasing robustness and performance in spectrum sensing [1, 4, 21, 31,32]. Among these techniques, convolutional neural network (CNN) has stood out and remains one of the most used [33]. CNN is a feedback neural network that can extract features from data with convolutional structures [33, 34]. Some CNN-based networks have become popular, such as ResNet, DenseNet, SqueezeNet, Inception, MobileNet, ConvNeXt [33–35], AutoEncoder [36], Transformers, and generative adversarial networks (GAN) [37]. With this arsenal of machine and deep learning algorithms, some approaches to spectrum sensing have become promising [1,4,32,38–40], such as cooperative sensing and the use of GAN for data augmentation and new research in spectrum security [41–48].

Cooperative sensing is a method for broadband sensing where multiple user SUs share their sensing information with a fusion center. The fusion center is where the sensing information from all SUs in the system is sent and analyzed, and then decision-making takes place [49]. With this collaboration, the chance of correctly identifying the PU increases considerably [1, 4, 32]. Due to various variables such as power, distance, noise, and shadowing effects, the SU may mistakenly identify the channel condition, compromising the dynamism of the cognitive radio [40]. Cooperative sensing deals with this by making decisions based on information from multiple SUs, increasing the diversity of the received data. With this approach, even if one SU fails to identify the presence of PUs in the channels, another SU in the cooperation with better conditions will be able to identify it, providing greater reliability to the system.

The proposal for this initial research, as published in [4], is for a cooperative deep spectrum sensing based on a simplified ResNet. In this approach, the SUs present in the system share the received channel condition information with a fusion center, where a pre-trained ResNet makes the decision whether there is presence of PUs or not. However, due to the large amount of data, the model proved to be slow both for training and inference, which hinders potential experimental applications. Therefore, to reduce computational cost, a joint approach with feature extraction

and a Random Forest classifier is proposed [1]. In this new proposal, various features are extracted from the signals received by the SUs, and a pre-trained Random Forest model makes individual decisions on whether there is presence of PUs in each channel. These new pieces of information are then shared with the fusion center, where a simplified ResNet makes the final decision on channel conditions based on the cooperation matrix.

Due to the nature of the experimental process carried out in the initial research, another avenue has become promising. Noise is a component present in virtually all types of transmitted signals, directly impacting signal quality, system performance, spectrum sensing, and security [50, 51]. The level of noise in a communication system can be influenced by the distance between users; thus, the noise level can vary depending on user proximity [52]. Therefore, having a priori information about the noise level and user distances can increase efficiency in spectrum usage, as well as in data analysis and decision-making. In this perspective, the use of prediction algorithms (regressors) is proposed to estimate the noise level based on the signals received by SUs sensing the spectrum. Similarly, the use of regression algorithms is also proposed to estimate the initial and final distances between users based on the received signals.

In parallel with these research proposals, a third front has shown promise. GANs are networks that can create and alter data, such as wireless signals [43,46,53]. With the use of these networks, it is possible for a malicious user (MU) to mimic characteristics of different users and thus compromise the functionalities of the cognitive radio [48, 54, 55]. In this context, a new research was conceptualized, with an initial proposal evaluated on the premise that the cognitive radio can more easily identify channel conditions with prior information about the received signal. The proposal involved the use of a semi-supervised generative adversarial network (SGAN) for generating fake signals with the intention of deceiving state-of-the-art modulation recognition models [48]. Based on this initial research, a new approach was conceived. Also, by using GANs, it would be possible for a MU simulate fake signals to access the fusion center and simulate a fake cooperation matrix, with the intention of deceiving the final decision model of the cooperative sensing system. Experimental results have shown that this is possible, and the MU could deceive these

decision-making models in the fusion center, drastically impacting the efficiency and dynamic of the spectrum.

1.1 General objectives

The frequency spectrum is an increasingly contested resource, therefore, methods for spectrum estimation, sensing, and security are essential to increase its efficiency, as well as its scalability. Thus, three main objectives are presented in this thesis: (1) the aim is to present a method for cooperative spectrum sensing in order to meet the high transmission rate requirements of the new generations of communication systems; (2) the goal is to present methods for noise estimation and distance estimation between users based on regression algorithms; and (3) to explore the latest deep learning networks for spectrum security purposes.

1.1.1 Specific objectives

The specific objectives are described below:

1. Investigate the use of feature extractor combined with a Random Forest model for complexity reduction.

2. Application of a simplified ResNet model for the channel condition classification stage based on user cooperation.

3. Investigate the use of regression algorithms for estimating noise and distances between users based on spectrum sensing signals.

4. Investigate the use of GAN for generating cooperation matrices with the intention of deceiving PU identification models in fusion centers.

1.2 Thesis contributions

The contributions related to cooperative spectrum sensing are: (1) the use of feature extractor combined with a Random Forest model to reduce system complexity; (2) application of a simplified ResNet also aiming to reduce latency; and (3) high accuracy even at high noise levels with low latency.

The contributions related to noise and distance estimation between users are: (1) the use of regression algorithms to estimate the noise of a signal at reception; (2) the use of regression algorithms to estimate the initial and final distance of the SU in relation to the PU; and (3) in the use of metrics related to regression problems, promising results were achieved in the correct prediction of noise and distances.

Contributions related to spectrum security include: (1) The use of SGAN for generating fake modulated signals to simulate attacks on automatic modulation recognition models; (2) The use of GANs to generate fake cooperative matrices to deceive the final decision models of channel conditions at the fusion center; and (3) The experiments presented show that the proposed method successfully deceived the fusion center, sparking promising debates about security.

1.3 Publications

Spectrum sensing:

- VALADÃO, Myke DM et al. Deep Cooperative Spectrum Sensing Based on Residual Neural Network Using Feature Extraction and Random Forest Classifier. Sensors, v. 21, n. 21, p. 7146, 2021.

  DOI: 10.3390/s21217146

- VALADÃO, Myke DM et al. Cooperative Spectrum Sensing System using Residual Convolutional Neural Network. In: 2022 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2022. p. 1-5.

  DOI: 10.1109/ICCE53296.2022.9730218

- VALADÃO, Myke DM et al. Trends and Challenges for the Spectrum Efficiency in NOMA and MIMO based Cognitive Radio in 5G Networks. In: 2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2021. p. 1-4.

  DOI: 10.1109/ICCE50685.2021.9427695

- VALADÄO, Myke DM; CARVALHO, Celso B.; JÚNIOR, Waldir SS. Trends and challenges for the spectrum sensing in the next generation of commu-

nication systems. In: 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2020. p. 1-2.

DOI: 10.1109/ICCE-Taiwan49838.2020.9258205

- VALADÃO, Myke DM et al. Classificação Automática de Modulações utilizando Redes Neurais Artificiais com Regularização Bayesiana e Algoritmo de Retropropagação de Levenberg-Marquardt. In: XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2020). SBrT 2020. p. 1-5.

  DOI: 10.14209/SBRT.2020.1570649633

- VALADÃO, Myke et al. MobileNetV3-based Automatic Modulation Recognition for Low-Latency Spectrum Sensing. In: 2023 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2023. p. 1-5.

  DOI: 10.1109/ICCE56470.2023.10043380

Spectrum estimation:

- VALADÃO, Myke DM et al. Noise Power Density Estimation Based on Deep Learning Using Spectrograms Extracted from Wireless Signals. In: XLII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2024). SBrT 2024. p. 1-5.

  DOI: 10.14209/sbrt.2024.1571036528

- (Submetted) VALADÃO, Myke DM et al. Predicting Noise and User Distances from Spectrum Sensing Signals Using Transformer and Regression Models. Digital Signal Processing, 2024.

Spectrum security:

- VALADÃO, Myke DM et al. Rede Adversária Generativa Semi Superversionada para Falsificação de Sinais Modulados Utilizados em Simulação de Ataque a Modelos de Reconhecimento Automático de Modulações. In: XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2022). SBrT 2022. p. 1-5. DOI: 10.14209/sbrt.2022.1570822071

- (Submetted) VALADÃO, Myke DM et al. Spoofing Deep Cooperative Spectrum Sensing Using Generative Adversarial Network. Engineering Applications of Artificial Intelligence, 2024.

Colaborations:

- LINHARES, José EBS et al. Asset Administration Shell in Manufacturing: Case Study in Manaus Industrial Complex. In: 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2024. p. 269-270.

  DOI: 10.1109/ICCE-Taiwan62264.2024.10674616

- VALADÃO, Myke DM et al. Forecast of Anomalies in Vacuum Pump based on HEX@ Sensor using TSMixer Model. In: 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2024. p. 339-340.

  DOI: 10.1109/ICCE-Taiwan62264.2024.10674528

- TORRES, Gustavo M. et al. A Designing Databases Framework for AI Training in Industrial Predictive Maintenance. In: 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2024. p. 263-264.

  DOI: 10.1109/ICCE-Taiwan62264.2024.10674079

- VASQUES, Adriel et al. Integrating NVIDIA Jetson Nano as a Data Server in IIoT Ecosystems: A Case Study. In: 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2024. p. 677-678.

  DOI: 10.1109/ICCE-Taiwan62264.2024.10674471

- CASTRO, Lucas GM et al. Middleware Ginga: Evolution, Challenges, and Future Perspectives - A Systematic Review. In: XLII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2024). SBrT 2024. p. 1-5.

  DOI: 10.14209/sbrt.2024.1571036781

- VALADÃO, Myke DM et al. Automatic Video Labeling with Assembly Actions of Workers on a Production Line Using ResNet. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2022. p. 323-324.

  DOI: 10.1109/ICCE-Taiwan55306.2022.9869008

- SILVA, Mateus O. et al. Action and Assembly Time Measurement System of Industry Workers using Jetson Nano. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2022. p. 319-320.

  DOI: 10.1109/ICCE-Taiwan55306.2022.9869028

- SILVA, Mateus O. et al. Action Recognition of Industrial Workers using Detectron2 and AutoML Algorithms. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2022. p. 321-322.

  DOI: 10.1109/ICCE-Taiwan55306.2022.9869197

- BESSA, Andrey RR et al. Design, Implementation and Evaluation of a Private LoRaWan Network for Industrial Internet of Things (IIoT) Applications. In: 2022 IEEE International Conference on Consumer Electronics-Taiwan. IEEE, 2022. p. 513-514.

  DOI: 10.1109/ICCE-Taiwan55306.2022.9869158

- PEREIRA, Antônio MC et al. Classificação Automática de Modulações DP m-PSK e DP m-QAM em Receptores Ópticos Coerentes Flexíveis. In: XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT). At: Brasil. SBrT 2021. p. 1-5.

  DOI: 10.14209/sbrt.2021.1570724020

- FURTADO, Rafael S. et al. Automatic Modulation Classification in Real Tx/Rx Environment using Machine Learning and SDR. In: 2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2021. p. 1-4.

  DOI: 10.1109/ICCE50685.2021.9427693

- TORRES, Gustavo M. et al. Automated Mura Defect Detection System on LCD Displays using Random Forest Classifier. In: 2021 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2021. p. 1-4.

  DOI: 10.1109/ICCE50685.2021.9427579

- FERREIRA, David AO et al. Dead pixel detection on liquid crystal displays using random forest, SVM, and harris detector. In: 2020 IEEE International

Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2020. p. 1-2.

DOI: 10.1109/ICCE-Taiwan49838.2020.9258171

- FERREIRA, David AO et al. Dead pixel detection on liquid crystal displays using random forest, SVM, and harris detector. In: 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2020. p. 1-2.

  DOI: 10.1109/ICCE-Taiwan49838.2020.9258207

1.4 Thesis Organization

This thesis is structured as follows:

- **Chapter 1 - Introduction:** Introduces the background, motivation, objectives, contributions, and outlines the thesis structure.
- **Chapter 2 - Deep Cooperative Spectrum Sensing Based on Residual Neural Network Using Feature Extraction and Random Forest Classifier:** Covers the proposed deep cooperative spectrum sensing method using a residual neural network combined with a random forest classifier.
- **Chapter 3 - Predicting Noise and User Distances from Spectrum Sensing Signals Using Transformer and Regression Models:** Presents prediction models for estimating noise levels and distances between users based on spectrum sensing signals using transformer and regression methods.
- **Chapter 4 - Spoofing Deep Cooperative Spectrum Sensing Using Generative Adversarial Network:** Discusses the use of GAN to spoof cooperative spectrum sensing systems and deceive decision models.
- **Chapter 5 - Conclusions:** Summarizes the research findings, implications, and provides potential directions for future work in spectrum sensing, estimation, and security.

Each chapter is designed to be self-contained, allowing them to be read independently. However, reading the chapters in sequence provides a more comprehensive understanding of the overall research. This structure allows readers to focus on specific areas of interest without needing to reference earlier chapters, making the thesis adaptable to different reading approaches.

Chapter 2

Deep Cooperative Spectrum Sensing Based on Residual Neural Network Using Feature Extraction and Random Forest Classifier

2.1 Introduction

With the advent of new generations of communication systems, such as 5G and 6G, high transfer rates have been required, leading to the necessity of wideband spectrum sensing. Various bands are sensed by the secondary users (SUs) participating in collaborative spectrum sensing. These sensing data are shared with a fusion center. The fusion center, whether centralized or decentralized, receives and analyzes all this information, making decisions on the presence or absence of primary users (PUs) in the evaluated bands [56]. The goal is to have a diversity of information and thus achieve a high probability of PU detection. In deep cooperative spectrum sensing, deep learning techniques are used to make these decisions.

A deep spectrum sensing is proposed using a simplified version of ResNet [1,4]. ResNet is an artificial deep neural network based on convolutional layers. The distinctive feature of ResNet is that its architecture was designed to address the vanishing gradient problem, allowing for the training of much deeper networks. In the ResNet architecture, there are residual blocks, which ensure that during learning,

the model only learns what is necessary, discarding residual information that the model has seemingly already learned in previous layers. There are several ResNet architectures; the most famous ones are ResNet-50 and ResNet-101, with 50 and 101 layers, respectively. A simplified version of ResNet is proposed precisely to reduce the inference time, thus reducing system latency.

Each SU performs sensing of multiple channels and applies a feature extraction step to reduce data complexity, making it suitable for low-latency scenarios. A Random Forest classifier for local PU detection follows this feature extraction. The decisions of individual SUs are then sent to the fusion center, where a simplified ResNet model aggregates this information to make a final decision, ensuring higher accuracy and robustness under noisy conditions [1, 57, 58]. This combination of feature extraction and hierarchical decision-making reduces system latency while maintaining high detection performance, achieving up to 98% accuracy in complex scenarios. In the application of spectrum sensing, the challenge is to select unique features that can distinguish a PU signal from a noise signal under various conditions. So, instead of processing large quantities of signal samples, the extractor drastically reduces this amount of information, providing a reduction in processing time.

With the use of the feature extractor, the complexity of the data to be applied to a classifier has been reduced, which also conditions the use of simpler classifiers for the task of unitary decision regarding the presence of PU in the channels sensed by each SU. In this case, due to the size of the data, classical machine learning models are suggested, mainly for their low inference time [1, 58]. The Random Forest classifier is proposed for the task of recognizing PU in the channels sensed by each SU. It is an algorithm used for both classification and regression, and it was developed precisely to overcome the problem of overfitting. It creates several decision trees randomly using bootstrap aggregating or bagging, which are data segregation techniques.

Thus, the use of feature extractor and Random Forest classifier is proposed for each SU in the cooperation system in spectrum sensing of various bands. The information from these SUs is shared with a fusion center, centralized or decentralized, where a simplified pre-trained ResNet model will make the final decision on whether

there is presence of PU in the channels sensed by the SUs or not. It is expected, in the proposed method, that with the plurality of information, the system can identify the presence of PU with higher accuracy. In addition, it should have low latency, which is highly recommended for the new generations of communication systems, such as 5G and 6G.

Each component of the cooperative spectrum sensing system is presented in the remainder of this chapter. It begins with the feature extraction stage in the SUs, which reduces data complexity and prepares the signal for local classification using Random Forests. Next, integrating local decisions at the fusion center using the simplified ResNet architecture is discussed, aggregating the diversity of information to achieve high accuracy even under high noise levels while maintaining system latency within 5G and 6G requirements.

2.2 Related works

This section reviews the main contributions in cooperative spectrum sensing, focusing on techniques such as energy detectors, deep learning methods, and cooperative strategies. The aim is to highlight the strengths and limitations of existing approaches compared to the proposed methodology.

2.2.1 Deep learning methods

In [39], the authors proposed an energy detector along with a convolutional neural network (CNN) for cooperative spectrum sensing. In this work, a single PU and several SUs move randomly in a certain area with a certain velocity over a period of time. An energy detector was used in two approaches, hard and soft decision, for identifying the PU in the evaluated bands. Then, the information from multiple SUs is shared with a fusion center, where a pre-trained model is used to identify whether there is a presence of PU in the evaluated frequency range. This pre-trained model was generated by a CNN with convolutional and fully connected layers. They achieved improved performance with an increase in the number of SUs in cooperation. However, the level of noise used in the experiments is low.

An approach using deep learning with a DLSenseNet (spectrum detection

network based on deep learning) is proposed by [59] for identifying channel conditions. The RadioML2016.10b database was used in the experimental process. The proposed DLSenseNet was composed of three blocks: an 'inc' block, an long short term memory (LSTM) block, and a dense block. The 'inc' block consists of three parallel paths with different filter sizes. The LSTM block contains 128 cells. Finally, the 'dense' block is composed of fully connected layers.

Finally, in [40], the authors proposed cooperative spectrum sensing with the generation of a database with 25 SUs and used a CNN in two scenarios. The data were generated considering the quadrature component, in-phase component, and frame size. Thus, the data have a format of $[25 \times 128 \times 2]$ for all scenarios. In the first scenario, the proposed CNN consisted of two convolutional layers followed by a flatten layer and two dense layers. For the second scenario, a more robust CNN is proposed, which was built with two convolutional layers with dropout in each, followed by a flatten layer and two dense layers.

2.2.2 Other methods

An energy detector associated with a classifier model for identifying spectral holes is also proposed by [3]. The energy detector is based on a threshold; if the signal energy is above this threshold, it indicates the presence of PU in the band; otherwise, there is no presence. The output information of the energy detector undergoes the SMOTE data augmentation algorithm (synthetic minority over-sampling technique) to not only increase the database but also balance it. Accuracy above 90% was achieved by applying the proposed methodology; however, few details were provided regarding the generation of the database.

The authors in [60] proposed a cooperative spectrum sensing approach to overcome the difficulty in classifying channel conditions under unfavorable received signal conditions. The relationship between interference range and detection sensitivity was described. An energy detection function and cooperative sensing with a regularization trade-off were applied to avoid time consumption and increase detection probability. With four cooperating SUs, a reduction in false alarm identification was already possible. And with an increase in the number of cooperating SUs, the detection probability increased considerably. Intuitively, it is reasoned that some

SUs will have significantly better channel conditions, resulting in a higher detection probability.

A methodology to optimize cooperative spectrum sensing is presented in [61]. The authors also used an energy detector but proposed a half-voting rule as an optimizer for the fusion rule. The error probability was evaluated with variations in the energy detector threshold and the number of cooperating SUs. It was concluded that with the half-voting rule, there was an improvement in performance, and depending on the threshold, either the AND or OR rule is more efficient.

In summary, the existing works present various approaches for spectrum sensing, ranging from traditional energy detection techniques to more complex deep learning-based methods. While energy detectors are computationally efficient, they often suffer from low performance under high noise and interference levels. On the other hand, deep learning models offer improved detection accuracy but can be computationally intensive, limiting their deployment in low-cost devices. Cooperative sensing strategies attempt to leverage the strengths of multiple techniques by aggregating decisions from different SUs; however, they may introduce additional latency and complexity in real-time scenarios. Therefore, a hybrid approach that combines the robustness of deep learning with the efficiency of shallow learning models, as proposed in this thesis, is necessary to achieve high detection accuracy and low latency in challenging communication environments

2.3 Methodology

This section presents the proposed methods for deep cooperative spectrum sensing.

2.3.1 System model

A cooperative spectrum sensing is proposed where each SU shares its detection information with a fusion center, either centralized or decentralized, where a model will decide whether there is presence of PU in the sensed channels or not. The proposed methodology is divided into two steps: (1) database generation, where signal generation [1, 4], feature extraction [1], training of a model that classifies the

presence or absence of PU in each individual sensed channel, and finally, generation of the cooperation matrix [1]; (2) training of a simplified ResNet network for more precise identification of PU in broadband, in the channels sensed by all cooperating SUs [1, 4].

In signal generation, two hypotheses are assumed: the noise signal and the signal with PU presence. These signals are generated taking into account several variables. It is assumed that the SUs and the PU move at the same speed, and their initial locations are random in a certain area, so their locations change over time. Bandwidth, noise power density, and multipath fading are variables also taken into consideration. Spectral and transform features are extracted from the generated signals to highlight their unique properties, reducing computational costs and facilitating the classification step. Now, the signals are represented by feature vectors, which are used to train a classifier that decides whether there is PU presence in each channel sensed by the SUs. Finally, all this information is shared, and cooperation matrices are generated. These matrices have the format [N_{SU} x N_B], where N_{SU} is the number of cooperating SUs and N_B is the number of sensed bands, which is fixed.

In step (2), the training of the ResNet model for cooperative sensing is performed. The cooperation matrices consist of 1s and 0s, where 1 indicates the presence of PU and 0 indicates the absence of PU. These matrices are used to train a simplified ResNet model, which will be applied to a fusion center. After generating the cooperation matrices, they serve as inputs for the ResNet model, which aggregates information from multiple SUs to refine the final decision-making process in the fusion center. The hypothesis is that the more SUs in cooperation, the higher the probability of identifying PU in the sensed channels, as there will be greater diversity of information. This results in greater efficiency of the cognitive radio and spectrum utilization.

2.3.2 Database

Signal generation

In the spectrum sensing process, the decision about the channel condition is binary, and two hypotheses are considered, H_1 and H_0, where H_1 represents the

signal with PU presence and H_0 without PU presence. For signal generation, it is assumed that N_{SU} and a single PU move at a velocity v, and their initial locations are random in a certain area, so their locations change over a time period Δt. It is also considered N_B bands, with B_W being the bandwidth. Additionally, it is assumed that the SU has no information about which bands are used by the PU, and the PU can use N_{B_P} consecutive bands. Then, the signal received by SU i in band j at time n can be described as:

$$y_i^j(n) = \begin{cases} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_P \\ \sqrt{\eta} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_A \\ w_i^j(n), & \text{for } H_0 \end{cases} \tag{2.1}$$

where $s_i^j(n) = \kappa_i(n) g_i^j(n) x(n)$ and $w_i^j(n)$ is the Gaussian white noise (AWGN) with zero mean, and standard deviation $\sigma = \sqrt{B_W 10^{\frac{N_0}{10}}}$. And N_0 is the noise power density in dbm/Hz. Let η be the proportion of power leaked to adjacent bands, then B_P are the bands occupied by the PU, and B_A are the bands affected by power leakage from the PU.

In $s_i^j(n)$, we have the simplified model and path, which can be written as:

$$\kappa_i(n) = \sqrt{\frac{P}{\beta (d_i(n))^{\alpha} 10^{\frac{h_i(n)}{10}}}} \tag{2.2}$$

where α and β are the path loss exponent and path loss constant, respectively. $d_i(n)$ is the Euclidean distance between the PU and the SU i at time n. The shadow fading of the channel, $h_i(n)$, between the PU and the SU i at time n in dB, can be described as a normal distribution with zero mean and variance σ^2, and P is the power transmitted by the PU in a specific band. Also, the multipath fading, $g_i^j(n)$, is modeled as an independent circularly symmetric complex Gaussian random variable (CSCG) with zero mean. Finally, $x(n)$ is the data transmitted by the PU at time n with an expected value of $x(n)$ equal to 1.

Feature extraction

For feature extraction, a set of spectral and transform features such as maximum power spectral density (indicating signal strength), standard deviation of normalized amplitude (capturing amplitude variability and noise level), and Walsh-Hadamard coefficients (representing frequency-domain characteristics) are computed

to highlight unique signal properties, including power concentration, frequency stability, and phase nonlinearity (see Appendix A for detailed mathematical formulations).

Random Forest classifier

The Random Forest classifier is a supervised machine learning algorithm initially proposed in 2001, and has been used in classification and regression tasks [62]. The "forest" is built with multiple decision trees, trained using the bagging or boosting method. With these approaches, the aim is to increase the model's performance and reduce the impact of issues like overfitting. For the proposed method, the default settings of the scikit-learn library are assumed, where the boosting method is the default. Weighted voting is used in the boosting method [3], so instances with higher weights have a greater probability of being selected for tree construction.

Cooperation matrix

Two groups of cooperation matrices will be generated:

1. In this group, some bands will have feature vectors representing the PU signal, while the remaining bands will have feature vectors representing the noise signal.

2. In this group, all bands will have feature vectors representing noise signals only.

The format of the matrices is given by $[N_{SU} \times N_B]$, where N_{SU} is variable and N_B is fixed. Thus, each SU in the cooperation senses the bands, extracts features, and uses the trained Random Forest model. The model returns 1 if there is a presence of PU in the channel and 0 if there is no presence of PU. Therefore, when this information is shared, the cooperation matrix consists of 1s and 0s in a 2D format.

In (1), since the SU doesn't know in which bands the PU may be present, the generation of the matrices is done randomly. For each generated matrix, a band is randomly chosen, and a different random output result of the classifier, which had a PU signal as input, is selected for each cooperating SU. The number of bands affected by power leakage is also randomly chosen, respecting adjacency to the initial position of the PU. In Algorithm 2, the logic for creating the cooperation matrix is presented. The number of occupied bands (N_{OB}) is randomly chosen between

1 and N_B. The notation $Vector \rightarrow B_P$ represents a random PU signal, while $Vector \rightarrow B_A$ represents a signal leaked to adjacent bands of the chosen PU signal. Each $Vector$ is randomly created with a signal of unknown N_0 and distances. The variable *prediction* denotes the output of the model for individual spectrum sensing, which is binary, consisting of 0s and 1s. The other bands are filled with output results of the classifier, which had a noise signal as input. In (2), all bands are filled with output results of the classifier, which had a noise signal as input only.

Algorithm 1 Cooperation Matrix Algorithm for (1)

function COOPERATION MATRIX (CM)
    $CM \leftarrow matrix(N_{SU}, N_B)$
    $N_{BO} \leftarrow random(1, N_B)$
    $positions \leftarrow choice(N_{SU}N_B, size = N_{BO})$
    for *pos* in *positions* **do**
        $row, col \leftarrow divmod(pos, N_B)$
        $CM[row, col] \leftarrow predict(Vector \rightarrow B_P)$
        $direction \leftarrow choice([left, right, both, none])$
        if *direction* is $left$ **then**
            if $col > 0$ **then**
                $CM[row, col - 1] \leftarrow predict(Vector \rightarrow B_A)$
            end if
        else if *direction* is *right* **then**
            if $col < N_B - 1$ **then**
                $CM[row, col + 1] \leftarrow predict(Vector \rightarrow B_A)$
            end if
        else if *direction* is *both* **then**
            if $col > 0$ **then**
                $CM[row, col - 1] \leftarrow predict(Vector \rightarrow B_A)$
            end if
            if $col < N_B - 1$ **then**
                $CM[row, col + 1] \leftarrow predict(Vector \rightarrow B_A)$
            end if
        else if *direction* is *none* **then** $None$
        end if
    end for
    return CM
end function

2.3.3 Proposed ResNet

A simplified ResNet was selected for its ability to address the vanishing gradient problem and its low computational complexity, making it suitable for real-time

applications in cooperative sensing. The input data for the proposed ResNet are 2D matrices, $[N_{SU} \times N_B]$, and the proposed architecture is a simplified version of ResNet, structured as follows: (I) initially, we have a residual block composed of a 2D convolutional layer (conv2D) for feature extraction, followed by a batch normalization layer, aiming to speed up and stabilize the network during normalization process, and then the rectified linear unit ($ReLU$) activation function. Next, another conv2D layer followed by a batch normalization layer. An Add operator is added to compute the residual, $H(x) = F(x) + x \rightarrow F(x) = H(x) - x$, where $F(x)$ is the map of learnable layers and x is the input data. Concluding this first residual block with $ReLU$ activation function; (II) the next layer is a max pooling 2D layer, which reduces the dimensionality of the output from the residual block; (III) a second residual block is applied, where a conv2D layer is followed by a batch normalization and $ReLU$ as activation function, sequentially followed by another conv2D and batch normalization. All of this in parallel with a conv2D and also a batch normalization. The Add operator computes the residual between the last two batch normalization layers in parallel, then $ReLU$ activation function is applied; (IV) the next layer is an average pooling 2D layer, which is a dimensionality reduction operator that computes the mean [2]; (V) a flatten layer is used to vectorize the data; (VI) followed by a dropout layer; (VII) and finally, a dense layer, which is fully connected, ending with softmax activation function, $y = \frac{e^x}{\sum e^x}$. In Figure. 2.1 is shown the architecture of the proposed simplified ResNet.

2.3.4 Metric

For model evaluation, the accuracy metric was applied, described as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.3}$$

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. An evaluation is also performed on the processing time of the entire process, including feature extraction, Random Forest model inference, creation of the cooperation matrix, and ResNet model inference.

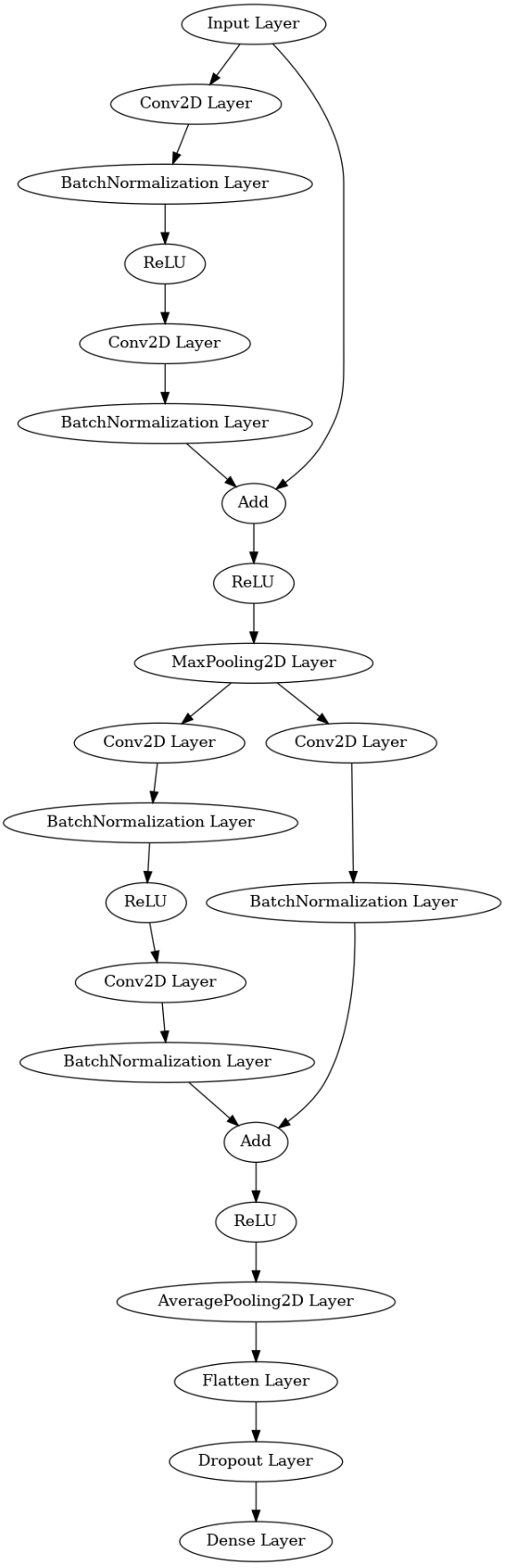


Figure 2.1: Architecture of the proposed ResNet.

2.4 Experiments and results

In this section, the results and analysis of the proposed cooperative spectrum sensing using feature extraction, Random Forest classifier for individual sensing, and ResNet for broadband cooperative sensing are presented. The proposed deep learning and machine learning techniques were compared with other techniques for method validation.

2.4.1 Database generation

The first step of the experiments is the signal generation and feature extraction. In signal generation, it is assumed that multiple SU and a single PU move at a speed of $v = 3$ km/h and their initial positions are randomly chosen in an area of 250 meters (m) $\times$ 250 meters (m), so the users' positions change over a period of time $\Delta t = 2$ seconds. The 3 km/h speed was chosen to simulate pedestrian mobility, a common scenario in urban cognitive radio networks. The area size of 250 m $\times$ 250 m ensures sufficient signal propagation variability while maintaining manageable computational complexity. N_B is 16, with B_W being 10 MHz, and the PU can use from 1 to 3 bands simultaneously. Additionally, $P = 23$ dBm, $\beta = 10^{3.453}$, $\alpha = 3.8$, $\sigma = 7.9$ dB, and N_0 is randomly chosen between -114 and -174 dBm/Hz. The proportion of power leaked to adjacent bands, η, is 10 dBm, so the power leaked to adjacent bands is half the power of the PU signal. For the experiments, $150,000$ instances were generated, divided into 75% for training and 25% for testing. N_{SU} varies between [1, 5, 10, 15, 20], totaling $750,000$ instances for all experiments, equally balanced between the two classes. $1,024$ samples per second of the signal were generated.

After signal generation, aiming to reduce system complexity, feature extraction is proposed as described by Equations (1) to (10) in Appendix A. The Hilbert transform and modulation are performed before feature extraction. The carrier frequency used is 2.412 GHz, which is widely employed in various wireless communication standards, including Wi-Fi and Bluetooth. In Figure 2.2 (left), the representation of the maximum value of the PSD of the normalized and centered instantaneous amplitude, $\gamma_{\max}$, for the two evaluated classes can be seen. It is noted

that the noise signal and the PU signal are correlated, as for all N_0 values, the two classes present similar values of $\gamma_{\max}$. In Figure 2.2 (right), the representation of the standard deviation of the normalized and centered instantaneous amplitude, σ_{aa}, for the two classes is presented. In this feature, it is noted that the noise signal and the PU signal values show a low level of correlation, especially at lower levels of N_0. It can be stated that the σ_{aa} feature can more easily differentiate the two classes.

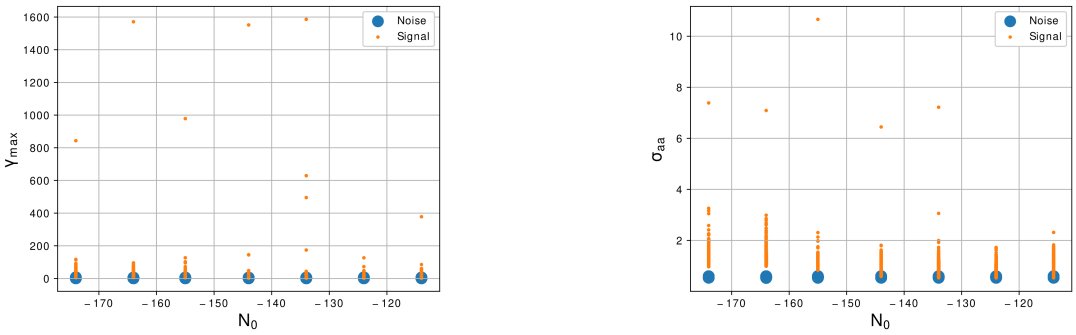


Figure 2.2: Graph of the features γ_{max}, Equation (1) (left), and σ_{aa}, Equation (2) (right), with variation of N_0 from -114 dBm/Hz to -174 dBm/Hz.

In Figure 2.3 (left), the representation of the standard deviation of the centered nonlinear absolute phase, σ_{ap}, for the two classes can be seen. In this case, it is noted that the difference between the noise signal and the PU signal is significant, especially at lower levels of N_0, where there is practically no intersection between the classes. The representation of the standard deviation of the centered nonlinear direct phase, σ_{dp}, in Figure 2.3 (right), is similar to the σ_{ap} feature. At higher levels of N_0, there is only a small area of overlap between the classes, which also facilitates the classifier's task.

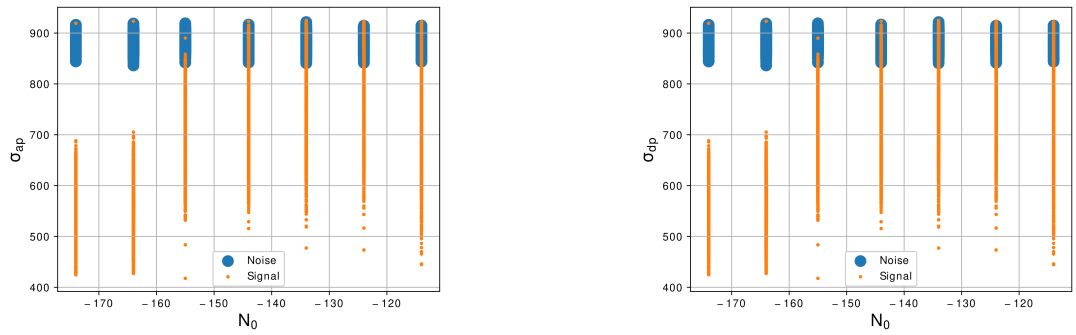


Figure 2.3: Graph of the features σ_{ap}, Equation (3), and σ_{dp}, Equation (4), with variation of N_0 from -114 dBm/Hz to -174 dBm/Hz.

The standard deviation of the normalized and centered instantaneous frequency, σ_{af}, is presented in Figure 2.4 (left). It can be observed that there is no overlap of values between the two classes at lower levels of N_0. Even at higher levels, there is only a small area of intersection, demonstrating that the σ_{af} feature is a promising resource for facilitating the classification process. Very similar to σ_{af}, the standard deviation of the absolute value of the normalized and centered instantaneous frequency, σ_f, in Figure 2.4 (right), shows a considerable difference between the classes, with a small area of correlation at higher levels of N_0. This feature also proves to be useful for distinguishing the noise signal from the PU signal.

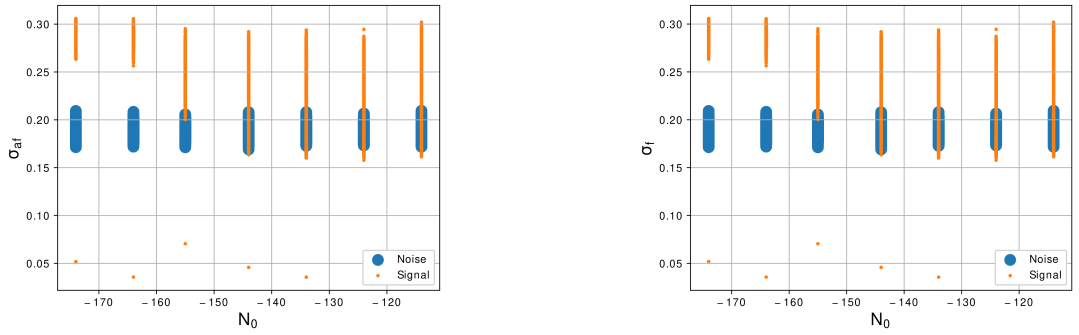


Figure 2.4: Graph of the characteristics σ_{af}, Equation (5), and σ_f, Equation (6), with variation of N_0 from -114 dBm/Hz to -174 dBm/Hz.

Figure 2.5 (left) presents the graph of the maximum value of PSD of the normalized and centered instantaneous frequency, γ_{maxf}, for noise signal and PU signal. It can be observed that there is some level of correlation between the classes in all levels of N_0. However, at lower levels of N_0, there is less overlap in the amplitude values. In Figure 2.5 (right), the maximum value of the discrete cosine transform, max_{dct}, is shown. It can be observed that the amplitude values are very similar for both classes, which complicates the classification process.

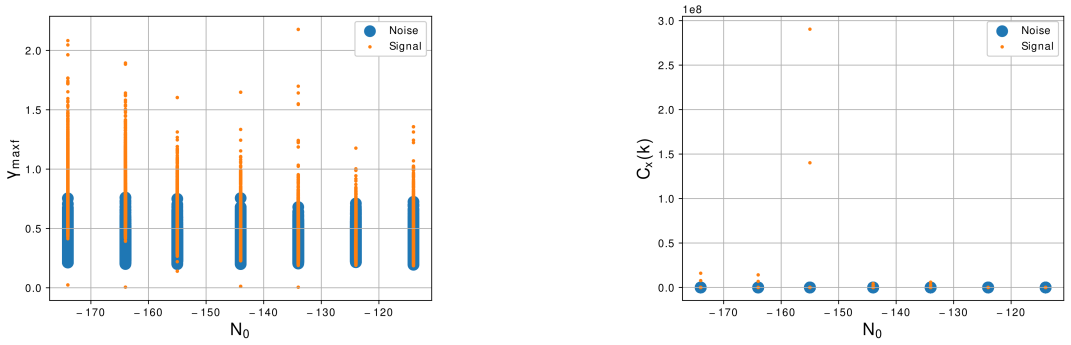


Figure 2.5: Graph of the features γ_{maxf}, Equation (7), and $C_x(k)$, Equation (8), with variation of N_0 from -114 dBm/Hz to -174 dBm/Hz.

The maximum value of the Walsh-Hadamard transform, σ_{wht}, is shown in Figure 2.6 (left). It can be observed that the behavior of this characteristic is similar to that of max_{dct}; the amplitude values of both classes in this characteristic exhibit a high level of correlation. Finally, the standard deviation of the discrete Wavelet transform, σ_{dwt}, is presented in Figure 2.6 (right). However, similar to the last two characteristics presented, max_{dct} and σ_{wht}, σ_{dwt} demonstrates high levels of correlation between the two classes.

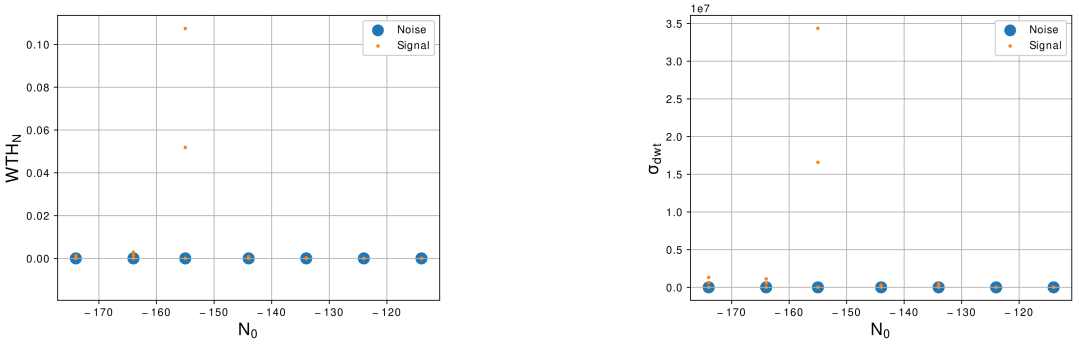


Figure 2.6: Graph of the characteristics WTH_N, Equation (9), and σ_{dwt}, Equation (10), with variations of N_0 from -114 dBm/Hz to -174 dBm/Hz.

After the feature extraction process, the signal representation is achieved through a vector of size 10, corresponding to the number of extracted features. With these operations, a reduction in the complexity of the model required to classify the two classes is expected, along with improved response time and computational cost, especially compared to [4]. The proposed model for the classification task is the Random Forest classifier, which is compared with other classical machine learning

techniques. The output of the Random Forest is a binary response: 1 if the input vector represents a PU signal, and 0 otherwise. Each SU then transmits to the fusion center a vector of size 16, which corresponds to the N_B channels sensed by each SU, along with the conditions of each of these sensed channels, i.e., 1s and 0s. The hyperparameters of the Random Forest classifier were configured according to the default specifications of the scikit-learn library.

In Figure 2.7, the accuracy of the proposed Random Forest classifier is presented in comparison with other classical machine learning techniques. It can be observed that the Random Forest achieves the best results for higher levels of N_0 and reaches 100% accuracy for lower levels of N_0. Compared to other methods, it is evident that k-nearest neighbors (KNN) and support vector machine (SVM) also exhibit good results, especially for lower levels of N_0. The Naive Bayes classifier is the method with the lowest accuracy, achieving good levels of accuracy only at low N_0 values. The Random Forest achieves an accuracy of over 80% for high noise levels (-114 dBm/Hz) and over 95% for N_0 values below -130 dBm/Hz.

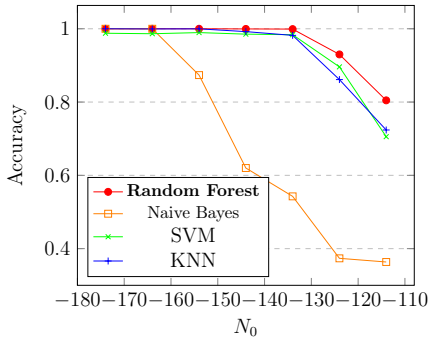


Figure 2.7: Graph of the accuracy of the Random Forest compared to classical machine learning techniques with N_0 ranging between -114 dBm/Hz and -174 dBm/Hz.

In Figure 2.8, the confusion matrix of the proposed Random Forest classifier for the range of N_0 between -114 dBm/Hz and -174 dBm/Hz is presented. It can be observed that the Random Forest model is able to recognize 97% of true positives for PU signals and 95% for noise signals. Additionally, the model misclassifies only a small percentage of instances, as shown in the confusion matrix.

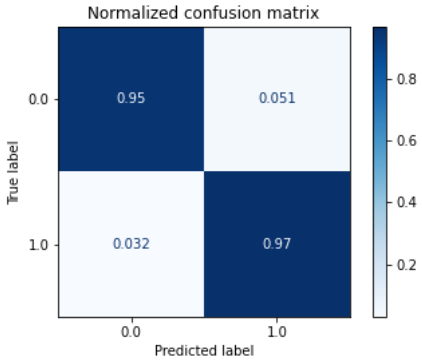


Figure 2.8: Confusion matrix of the Random Forest classifier with N_0 ranging from -114 dBm/Hz to -174 dBm/Hz.

2.4.2 Deep cooperative spectrum sensing

For the cooperative sensing approach, the input matrix of the proposed ResNet consists of N_{SU} and N_B $[N_{SU} \times N_B]$, thus, we have a 2D matrix containing information about the conditions of 16 channels and the cooperation of N_{SU} users. The proposed method is compared with other deep learning approaches, such as CNN and recurrent neural network (RNN), to demonstrate the benefits of ResNet in cooperative sensing. CNN and RNN were chosen as comparative models due to their widespread use in time-series analysis and signal processing. However, ResNet is expected to outperform these models due to its residual learning capabilities. These capabilities help mitigate the vanishing gradient problem, which occurs when gradients become too small during backpropagation. This leads to minimal updates in the initial layers and hinders effective training in deep networks. In these subsections, the system's response is also presented with the variation of N_{SU} compared to CNN and RNN networks, considering that N_0 varies between -114 dBm/Hz and -174 dBm/Hz. Another analysis is performed regarding the processing time of the entire system with the proposed ResNet and the other methods.

The parameters of the proposed ResNet are described as follows: in the residual conv2D layers, we have a kernel size of $[3, 3]$ and strides of [1, 1], with the padding parameter set to same. The first and second residual layers extract 16 and 32 filters, respectively. After the first residual layer, there is a max pooling 2D layer with a pooling size of [2, 2], which reduces the size of the output matrix from the first residual layer by half. After the second residual layer, there is an average

pooling 2D layer with a pooling size of [4, 4], reducing the size of the output matrix from the second residual layer by four times. Next, there is a flatten layer, which vectorizes the data, followed by a dropout layer with a dropout rate of 0.6. Finally, there is a fully connected (dense) layer with a softmax activation function, where the outputs are binary, representing whether or not there is a presence of PU in the evaluated channels. The optimization function used is Adam with a learning rate of 0.001. The number of epochs is 100, and the batch size is 32.

The parameters of the CNN and RNN are described as follows: the designed CNN has two conv2D layers with 8 and 16 filters, respectively. *ReLU* activation function is applied to these two layers. Following this, a 2D max pooling is applied, followed by a flatten layer and a dense layer with 64 neurons and softmax activation function. The RNN has 3 fully connected layers with 100, 50, and 2 neurons, respectively. *ReLU* activation function is applied to the first two layers, and the softmax function is used as the activation function for the last layer. The optimization function, learning rate, number of epochs, and batch size are the same as those applied in the proposed ResNet.

In Figure 2.9, the accuracy of the proposed ResNet network is compared with CNN and RNN networks. It can be observed that the ResNet approach shows better performance, especially at high levels of N_0, achieving over 90% accuracy when N_0 is below -124 dBm/Hz. With N_0 below -134 dBm/Hz, the accuracy exceeds 98%. These results consider the cooperation of 10 SU in the system. In Figure 2.10, the graph shows the response of the ResNet with the increase in the number of SU in the system. It can be seen that as N_{SU} increases, the system accuracy also increases, and with 5 SU, the proposed method achieves about 94% accuracy, while the other methods do not reach 92%. With 10 SU, the ResNet has an accuracy of over 96%, and with 20 SU, the accuracy is about 98% with N_0 ranging from -114 dBm/Hz to -174 dBm/Hz. Increasing the number of SUs improves accuracy and introduces additional complexity in the fusion center, potentially increasing the overall processing time. Depending on the target application and latency requirements, this trade-off must be carefully balanced.

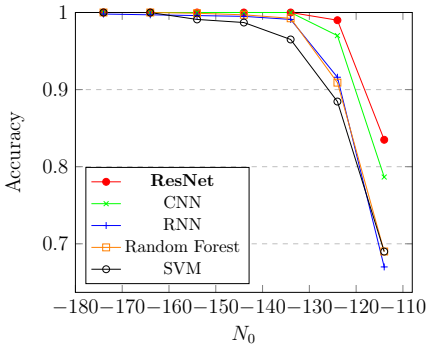


Figure 2.9: Graph of the proposed ResNet compared to other machine learning and deep learning methods with varying N_0.

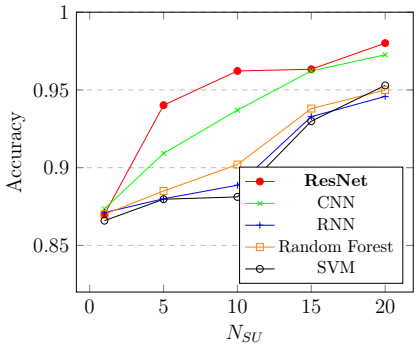


Figure 2.10: Graph of the proposed ResNet compared to other machine learning and deep learning methods with varying N_{SU}.

Another important analysis for the cooperative sensing approach is the system's response time. We can observe in Figure 2.11 that as the number of SU increases, the response time also increases. This is because the classification model needs to be more robust to classify more SU in the system. We can see that even when compared to less complex neural network architectures, the ResNet can still provide, in most cases, a shorter response time.

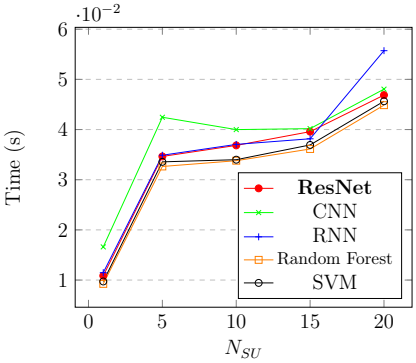


Figure 2.11: Response time of the entire system for the proposed ResNet with other machine learning and deep learning methods considering the variation of SU.

2.5 Discussions

In this chapter, we addressed cooperative spectrum sensing based on a ResNet using a feature extractor and a Random Forest classifier. In the first step of the proposed methodology, we generated signals based on Equations (4.1) and (4.2). Then, we extracted features, similar to the proposal in [58], but instead of using 29 features as in [58], we proposed using 10 features as described by Equations (1) to (10), which are spectral and transformation characteristics. This feature extraction aims to reduce complexity and computational cost. With this new signal representation, a Random Forest classifier is proposed to identify whether this representation is a PU signal or a noise signal. Then, a matrix with information from the channels of various SUs is used as input to a model that can identify, in broadband, the presence of UP in multiple channels.

The signal generation was done similarly to the proposal by [63] with some differences. The first, and most relevant, is the range of N_0 applied in their studies, which varies between -154 dBm/Hz and -174 dBm/Hz, while in this study, we applied a range of -114 dBm/Hz to -174 dBm/Hz. Our proposal considers signal conditions with higher noise influence, which provided us with a better perspective on how our methodology responds to more stressful signal propagation environments. Another change in signal generation compared to [63] is the mobility area of SU and PU. In [63], their users can move in an area of 200 m $\times$ 200 m, while we configured this area to be 250 m $\times$ 250 m. Signal degradation is related to the distance between SU and PU, so in a larger mobility area, it is expected that the signal will be more influenced by noise. Our signal generation also takes into account the low number of SU in the system. In this experiment, we simulated cooperation of up to 20 SU, which, due to the results obtained, further enhances the reliability of our methodology.

When it comes to the feature extractor, we extracted 10 features out of the 29 proposed by [58]. The first feature is γ_{max}, Equation (1), and we observed that in this feature, noise and PU signal are correlated, as we can see in the graph of Figure 2.2 (left), so this feature is not a priority for segregating the two classes. The second feature is σ_{aa}, Equation (2), and this feature presents a better response than the previous one and can differentiate noise and PU signal more easily, as shown

in Figure 2.2 (right). The next feature is σ_{ap}, Equation (3), and the differences between the two classes are significant, facilitating class segregation, as shown in Figure 2.3 (left). The fourth feature, shown in Figure 2.3 (right), is σ_{dp}, Equation (4), and, like the previous feature, there is a significant difference between the two classes. The fifth and sixth features are σ_{af} and σ_f, Equations (5) and (6), shown in Figures 2.4 (left) and (right), also showing a good difference between the two classes. The next feature is γ_{maxf}, Equation (7), and it shows some level of correlation at all levels of N_0, as we can see in Figure 2.5 (left), even at the lowest level of N_0, there is still some overlap between the classes. Furthermore, the last three features, Figures 2.5 (right) and 2.6 (left) and (right), show a high level of correlation and may be less relevant in the classification task. In this study, we concluded that the best features to represent the signals are σ_f, σ_{af}, σ_{dp}, σ_{ap}, and σ_{aa}, due to the lower level of overlap between the two classes at all levels of N_0. In [58], they obtained good results with the approach using 29 features; for our methodology, 10 of these 29 were sufficient to achieve excellent results. A study to further optimize these features may be a future work.

The Random Forest classifier is proposed to evaluate the individual conditions of each channel. We compared the Random Forest approach with three classical machine learning algorithms, and the proposed Random Forest achieved the best result, especially at higher levels of N_0. With N_0 at -114 dBm/Hz, the Random Forest achieved over 80% accuracy; at -134 dBm/Hz, accuracy reached about 98%, and at the lowest N_0, accuracy is 100% in correctly classifying noise and PU signal. In the graph of Figure 2.7, we can observe the superior performance of the Random Forest classifier at the highest noise level compared to other classical approaches. Naive Bayes performed poorly at the high N_0 level, while the other approaches, SVM and KNN, performed well but did not reach the performance level of the Random Forest classifier. In the confusion matrix, Figure 2.8, we can see that the Random Forest achieved an excellent result. In [3], the authors obtained 91% accuracy in identifying spectrum holes based on energy detection, compared to the 96% accuracy achieved in the methodology proposed in this study.

The proposed method for cooperative sensing based on ResNet outperforms the other compared methods. In Figure 2.9, it can be seen that the proposed ResNet

had better performance, especially at higher levels of N_0. With -114 dBm/Hz, for example, the ResNet accuracy was around 83%, while CNN and RNN achieved accuracy below 80%, with the cooperation of 10 SU. The proposed method achieves accuracy above 98% with N_0 below -130 dBm/Hz. In Figure 2.10, the increase in accuracy of the proposed method with the increase of N_{SU} is shown. We can observe that the ResNet performed better than the other methods, reaching about 98% with 20 SU in the system. In [63], they used a CNN together with an energy detector for cooperative sensing; their results were obtained considering a lower level of N_0 and a much larger number of N_{SU} in the system. We achieved high accuracy with a higher level of N_0 and fewer N_{SU} in the system. Additionally, in [64], for example, they achieved about 95% accuracy with the cooperation of 20 SU, while in this study, about 98% was obtained. The system response time was also taken into consideration. In Figure 2.11, we show the computation time in seconds. We can see that the proposed method was faster in most points and kept the time below 0.05 seconds even with 20 SU in the system. In [63], they obtained a faster response with 20 SU than the proposed method, while in [64], they obtained a longer response time in some tests. For future work, we propose to increase the number of N_{SU} to evaluate the proposed system.

When comparing the proposed method with classical approaches such as Random Forest and SVM, we can observe that the ResNet achieved better accuracy at all levels of N_0 with 10 SU in the system, as shown in Figure 2.10. It can be seen that the classical methods performed similarly to the RNN classifier. When comparing the increase in SU in the system, we can say that the proposed ResNet also achieves better accuracy than classical machine learning approaches. We can observe that the Random Forest classifier performed better than the RNN, and the SVM had a performance similar to the RNN, but with 20 SU, the SVM performed better than both the Random Forest and the RNN, as shown in Figure 2.10. In Figure 2.11, the system response time is compared with different models; it can be observed that the classical methods had a better response time than the proposed method. However, it is not worth using the classical methods instead of the proposed ResNet, due to the significant difference in accuracy and the small difference in response time.

2.6 Conclusion

This chapter introduced a novel framework for cooperative spectrum sensing that combines a ResNet based feature extractor with a Random Forest classifier. The feature extractor reduced the complexity of the signal representations, transforming raw data into a format more suitable for classification and enabling the Random Forest classifier to achieve high performance even in challenging scenarios. Extensive simulations and numerical results demonstrated that the proposed ResNet for cooperative sensing can achieve higher accuracy, even in environments where the signals are strongly influenced by noise, reaching approximately 98% with N_0 at -130 dBm/Hz and with 10 SU in the system. Another point relates to the system response time throughout its operation. It is shown that it is possible to achieve a faster response, even with a considerable number of N_{SU}, with response times below 0.05 s.

The spectral and transformation features extracted have indeed proven to be effective in highlighting unique characteristics of the signals received by the SU. This can be verified by the performance of the proposed Random Forest classifier, which achieves a high success rate even at high levels of N_0, demonstrating that the combination of these two methods is efficient in recognizing PU in the evaluated channels, even under adverse conditions. Another contribution of this article is related to the proposed deep neural network for cooperative sensing. For instance, when comparing our results with those proposed by [63], we can observe that the proposed ResNet achieves a similar accuracy rate at higher levels of N_0 and without the need for as many SU in the system. Thus, our entire framework has demonstrated a good level of reliability under adverse channel conditions, which is necessary in high-rate data transmission systems, such as the new generations of communication systems.

As a future work, improvements can be made regarding resource optimization. A study on the performance of the proposed method by reducing the number of extracted features should be conducted, as we noticed that some of the proposed features for signal extraction showed a high level of correlation between the two classes. This may enhance performance and reduce the system response time. Another future work is related to N_{SU}. In the literature, a large number of SU in the system is proposed, but in our proposed method, only 20 were tested in the

experiments. It would be ideal to evaluate the system with a higher N_{SU} as well.

Chapter 3

Predicting Noise and User Distances from Spectrum Sensing Signals Using Transformer and Regression Models

3.1 Introduction

The frequency spectrum, essential for wireless communications, is a limited resource that has become increasingly congested [1,5,63]. Cognitive radio (CR) dynamically allocates communication for secondary users (SUs) in parts of the spectrum, known as spectrum holes, where primary user (PU) are absent [1,4,35]. With the increasing number of devices competing for limited spectral resources, efficient detection of spectrum holes becomes challenging. This emphasizes the need for advanced models to adapt to dynamic environments and varying interference levels. Several techniques exist for spectrum sensing, including energy detection, feature detection, Nyquist and sub-Nyquist methods, multi-bit and one-bit compressive sensing. More recently, the use of machine and deep learning models for detecting the presence of PU [7, 65]. Across these methods, noise can adversely affect the accurate detection of PU, impacting the efficient use of the spectrum. In multi-user systems, the distance between these users significantly influence noise level. Thus, accurately predicting noise levels and understanding user distances are crucial for optimizing spectrum utilization and enhancing communication efficiency.

In communication systems, noise refers to any unwanted or random interfer-

ence that compromises the quality of a transmitted signal. Noise can distort the original information being transmitted, leading to errors, diminished signal clarity, and reduced communication performance [66]. Almost all communication channels and systems encounter noise, which can originate from various sources. The noise level can be influenced by several factors, including the power transmitted, path loss, shadow fading, multipath fading, and the distance between users. In addition to these factors, additive white Gaussian noise is a type of noise that exists across all frequencies [67]. The noise is characterized by its randomness and uniform energy distribution throughout the frequency spectrum. White Gaussian noise is a common model for representing random background noise in communication systems. By predicting noise levels, one can more effectively optimize system parameters and frequency allocation [68].

The distance between users is an important factor that influences the signal quality and consequently the spectrum efficiency. In spectrum sensing, the strength of the received signal depends on the distance between the transmitter and receiver. Predicting these distances helps adjust transmission power levels, closer users require lower power for reliable communication, while users farther away need higher power [69]. This power control optimizes energy efficiency and minimizes interference. Especially in systems with limited resources, such as bandwidth, predicting distances is fundamental for effective resource allocation. Users closer to the transmitter can be assigned greater bandwidth and consequently higher data rates, while those farther away might be allocated smaller bandwidths to improve the signal quality at the same transmission power level. Additionally, distance prediction is essential for providing location-based services. Furthermore, the distance between users can also impact the performance of artificial intelligence models used to increase spectrum efficiency. By estimating distances, services like navigation, location-based advertisements, and emergency services can be offered.

To predict noise and distances between users, the use of regression models and deep learning architectures is recommended [70]. Regression models are a class of machine learning algorithms designed to predict continuous numerical values based on input data. These models play a crucial role in various fields, including economics, finance, healthcare, and natural sciences. They allow for the analysis and

forecasting of trends, relationships, and outcomes by learning from historical data. Among the most traditional machine learning regression approaches are support vector regression (SVR) (an extension of support vector machines to regression), Decision Trees and Random Forests [71], linear regression (one of the simplest yet widely used regression techniques), and ridge and lasso regression (variants of linear regression). In deep learning regression approaches, convolutional neural networks (CNN), initially designed for image analysis, can be adapted for regression tasks [72]. More recently, the Transformer, designed for forecasting, has shown to be interesting for regression activities [73].

In this chapter, we propose the use of regression models to predict noise levels and distances based on spectrum sensing signals. During our study, we generated a dataset that considers important parameters, including a wide range of noise power densities, an extensive sensing area, and power leakage from the PU. We have compared both traditional and deep learning models for prediction purposes. Furthermore, we evaluated the results using various metrics. Our proposed method has shown promising results, with a correlation coefficient exceeding 0.98 for noise and over 0.82 for distances. Additional metrics assessed also indicate that our method is effective in predicting noise levels and distances between users. Such predictive capability can be important in designing communication systems to enhance spectrum efficiency.

3.2 Related Works

In this section, the related works about noise and distances between users prediction in a spectral sensing scheme are described in detail.

3.2.1 Noise prediction

The authors proposed a signal-to-noise ratio (SNR) estimation method based on the sounding reference signal and a deep learning network in [74]. The proposed deep learning network, called DINet, combines a denoising convolutional neural network (DnCNN) and an image restoration convolutional neural network (IRCNN) in parallel. The method was compared to other algorithms, and the results demon-

strated superior performance in SNR estimation. The evaluation metric used was the normalized mean square error (NMSE) calculated over 200 test samples, yielding a NMSE value of 0.0012. This result was significantly better than the performance achieved by other algorithms.

In [68], the authors presented a method for estimating SNR in LTE systems and 5G. They employed a combination of a CNN and long short-term memory (LSTM), known as a CNN-LSTM neural network. The CNN was utilized to extract spatial features, while the LSTM was used to extract temporal features from the input signal. Data was generated using MATLAB LTE and 5G toolboxes, taking into consideration modulation types, path delays, and Doppler shifts. The evaluation metric used was NMSE. The NMSE achieved a value of zero in the time-domain for SNR ranging from -4 to 32 dB, demonstrating very low latency. However, in the frequency-domain, the proposed method exhibited lower performance.

In [75], the authors proposed NDR-Net, a novel neural network for channel estimation in the presence of unknown noise levels. NDR-Net comprises a noise level estimation subnet, a DnCNN, and a residual learning cascade. The noise level estimation subnet determines the noise interval, followed by the DnCNN, which processes the pure noise image. Subsequently, residual learning is applied to extract the noiseless channel image. The evaluation metric used for assessing the model's performance was the mean square error (MSE). The experiments conducted across different channel models (TDL-A, TDL-B, TDL-C) consistently demonstrated low MSE values. However, it's worth noting that the proposed model's performance was evaluated within a SNR range of 0 to 35. This limited range does not provide a comprehensive understanding of the model's robustness, particularly in scenarios with high levels of noise.

In Table 3.1, some techniques for noise prediction are summarized and compared. The methodology proposed in this article considers several variables that influence signal quality for data generation. The study incorporates noise power density across a wide range of values and explores a spectrum sensing environment where multiple users are in motion at fixed speeds over time. Several regression models are compared using various metrics to highlight the robustness of the proposed method.

Table 3.1: Related works on noise prediction.

Reference	Research Direction	Contribution	Limitation
[74]	SNR estimation method based on the sounding reference signal and a deep learning network	Proposed a DICNN for SNR estimation, which is a DnCNN and IRCNN in parallel	The number of testing samples is small
[68]	Method for estimating SNR in LTE systems and 5G	They used a CNN-LSTM neural network to extract spatial and temporal features	The proposed method exhibited lower performance in the frequency-domain
[75]	Novel neural network for channel estimation in the presence of unknown noise levels	Proposed an NDR-Net for channel estimation, which comprises a noise level estimation subnet, a DnCNN, and a residual learning cascade	Limited to a small range of noise level
Proposed	Prediction of noise power density in spectrum sensing signals	Proposed several regression algorithms for predicting noise power density in signals considering several other variables that influence the quality of the signal	Computing power was a limitation for training with more data and robust architectures

Table 3.2: Related works on distances prediction.

Reference	Research Direction	Contribution	Limitation
[76]	User equipment positioning in non-line-of-sight scenarios	Proposed customized ResNet for the path gain dataset and a ResNet-18 for the channel impulse response dataset	Without finetune and with large number of samples the models exhibited lower performance
[77]	Indoor fingerprint positioning based on measured 5G signals	A CNN was trained to locate a 5G device in an indoor environment. The experiments were conducted in a real field and demonstrated a positioning accuracy of 96% for the proposed method	The proposed method is not compared with other deep learning models
[78]	Location-aware predictive beamforming approach utilizing deep learning techniques for tracking unmanned aerial vehicle communication beams in dynamic scenarios	Designed a recurrent neural network called LR-Net, based on LSTM, to accurately predict unmanned aerial vehicle locations. Using the predicted location, it was possible to determine the angle between the unmanned aerial vehicle and the base station for efficient and rapid beam alignment in the subsequent time slot	Limited only to unmanned aerial vehicle-to-base station communication
Proposed	Prediction of initial and final distances between users during spectrum sensing	Proposed several regression algorithms for predicting distances between users considering several other variables that influence the quality of the signal	Computing power was a limitation for training with more data and robust architectures

3.2.2 Distances prediction

In [76], the authors proposed a deep learning approach for user equipment positioning in non-line-of-sight scenarios. The impact of variables such as the type of radio data, the number of base stations, the size of the training dataset, and the generalization of the trained models on 3GPP indoor factory scenarios was analyzed. The model trained consisted of a customized residual neural network (ResNet) for the path gain dataset and a ResNet-18 for the channel impulse response dataset. The metric used was the 90% quantile of the cumulative distribution function of the horizontal positioning error. The authors obtained the best performance with a large number of samples for training and tuning the models.

In [77], the authors proposed a machine learning algorithm for indoor fingerprint positioning based on measured 5G signals. The dataset was created by collecting 5G signals in the positioning area and processing them to form fingerprint data. A CNN was trained to locate a 5G device in an indoor environment. The metrics used were the root mean square error (RMSE) and the circular error probable. The experiments were conducted in a real field and demonstrated a positioning accuracy of 96% for the proposed method.

Finally, in [78], the authors introduced a location-aware predictive beamforming approach utilizing deep learning techniques for tracking unmanned aerial vehicle communication beams in dynamic scenarios. Specifically, they designed a recurrent neural network called LRNet, based on LSTM, to accurately predict unmanned aerial vehicle locations. Using the predicted location, it was possible to determine the angle between the unmanned aerial vehicle and the base station for efficient and rapid beam alignment in the subsequent time slot. This ensures reliable communication between the unmanned aerial vehicle and the base station. Simulation results demonstrate that the proposed approach achieves a highly satisfactory unmanned aerial vehicle-to-base station communication rate, approaching the upper limit attained by a perfect genie-aided alignment scheme. The exact locations and angles are perfectly known in this idealized scenario, serving as a benchmark for evaluating alignment algorithms.

In Table 3.2, some techniques for distances prediction are summarised and compared. The methodology proposed in this article considers several variables

that influence signal quality for data generation. The study incorporates Euclidean distances between users across a wide range of values and explores a spectrum sensing environment where multiple users are in motion at fixed speeds over time. Several regression models are compared using various metrics to highlight the robustness of the proposed method.

3.3 Methods

This section presents the proposed methods for data generation, noise and distance predict.

3.3.1 System model

Unlike previous methods, which used real-world datasets with corresponding ambient noise variability, our approach generates synthetic data, allowing for controlled variations in key parameters such as noise levels and mobility patterns. For this, the methodology is divided as follows: (1) database generation, where the signals that represent PUs are generated [1,63]; (2) training the proposed regression models [79, 80]; and (3) evaluation of the trained models for predicting noise level and distance [70, 81]. At the end of the process, it is expected that the models will be able to predict the level of noise and the initial and final distance between the PU and SU during the sensing period.

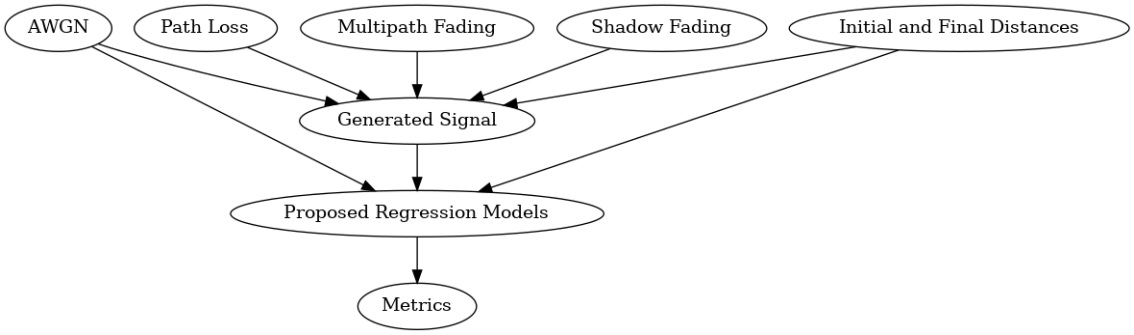


Figure 3.1: Complete scheme for noise level and distances between users prediction in the spectrum sensing network.

As illustrated in Figure 3.1, the dataset utilized for training the proposed regression models comprises signal data, noise levels, and distances between the

users. Initially, the SU and PU are positioned within the area, moving randomly for a duration of Δt. During this time, signals are sensed, and data on noise levels (N_0) as well as initial (D_i) and final (D_f) distances are collected. This collected data is then used to train and test the proposed regression models. To validate the effectiveness of the method in predicting noise levels and distances, various metrics are calculated.

The output of step (1) is expected to be the signals that represent the PU, the associated noise level for each signal, and the initial and final distances during the sensing. Using this information, two regression models are trained: one for noise level prediction and another for initial and final distance predictions. In step (2), several machine learning models are employed, including Random Forest, Decision Tree, Extra Trees, XGBoost, LightGBM, Support Vector Regression (SVR), along with deep learning models such as CNN and Transformer. The output of this step consists of predicted values for noise level and distances given a test dataset. Lastly, in step (3), several metrics are used to evaluate the best models for these tasks.

3.3.2 Signal generation

In the spectrum sensing process, the decision on the channel condition is binary, involving two hypotheses: H_1 and H_0 [82]. Here, H_1 represents the hypothesis in which the PU is present, while H_0 represents the absence of the PU [63]. For the purposes of this paper, we will only consider hypotheses involving the presence of the PU. We assume that N_{SU} SUs and a single PU are moving at a speed v, with their starting positions randomly chosen within a given area. As a result, the users' locations change over a time interval of Δt. Additionally, we are considering a multi-channel system with N_B bands, each having a bandwidth of B_W. Furthermore, we assume that the PU can utilize N_{B_P} consecutive bands [1]. Therefore, the received signal of the i-th SU on the j-th band at time n can be described as

$$y_i^j(n) = \begin{cases} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_P \\ \sqrt{\eta} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_A \end{cases} \tag{3.1}$$

where $s_i^j(n) = \kappa_i(n) g_i^j(n) x(n)$ and $w_i^j(n)$ is the additive white Gaussian noise (AWGN) whose noise power density is N_0, mean zero and standard deviation

$\sigma = \sqrt{B_W 10^{\frac{N_0}{10}}}$. Being η the proportion of power leaked to adjacent bands, then B_P are the bands occupied by the PU and B_A are the bands affected by the leaked power of the PU.

In the expression $s_i^j(n)$, a simplified path loss model is utilized, which can be written as follows:

$$\kappa_i(n) = \sqrt{\frac{P}{\beta(d_i(n))^{\alpha} 10^{\frac{h_i(n)}{10}}}}, \tag{3.2}$$

where α and β denote the path-loss exponent and path-loss constant, respectively. Here, $d_i(n)$ represents the Euclidean distance between the PU and SU i at time n. The shadow fading of the channel, indicated by $h_i(n)$, between the PU and SU i at time n in decibels (dB) can be described by a normal distribution with a mean of zero and a variance of σ^2. The term P denotes the power transmitted by the PU within a specified frequency band. Furthermore, the multipath fading factor, denoted as $g_i^j(n)$, is modeled as an independent zero-mean circularly symmetric complex Gaussian (CSCG) random variable. Moreover, the data transmitted at time n, represented by $x(n)$, has an expected value of one [1, 63].

3.3.3 Regression models

The machine learning models used to predict interference and distance between the PU and SUs are the Random Forest, Decision Tree, Extra Trees, XGBoost, LightGBM, CNN, SVR and Transformer.

3.3.3.1 Random Forest (R.F.)

Random Forest is known for its robustness against overfitting and ability to handle noisy datasets better than individual decision trees, making it a suitable choice for scenarios with varying interference levels. The Random Forest consists of a collection of trees denoted as $h(x; \theta_k), k = 1, ..., K$. Here, x represents an input vector of length q, containing a correlated random vector X, while θ_k refers to independent and identically distributed random vectors. In the context of regression, assume that the observed data is drawn independently from the joint distribution of (X, Y), where Y represents the numerical outcome. This dataset includes $n(q + 1)$-tuples,

namely $(x_1, y_1), ..., (x_n, y_n)$ [83]. The prediction of the Random Forest regression is the unweighted average over the collection

$$h(x) = \frac{1}{K} \sum_{k=1}^{K} h(x; \theta_k) \tag{3.3}$$

3.3.3.2 Decision Tree (D.T.)

In the context of regression, the Decision Tree is based on recursively partitioning the input features space into regions and then assigning a constant value to each region. This constant value serves as the prediction for any data point that falls within that region. Assume that X is the input data, Y the target variable and θ represents the parameters that define the splits in the Decision Tree [84]. Let $h(X; \theta)$ be the predicted value for Y given input X and parameter θ. Given a set of n training samples (X_i, Y_i), where $i = 1, 2, ..., n$, the decision tree regressor seeks to find optimal split θ that minimize the sum of square differences between the predicted value and the actual target value. The prediction for a given X can be represented as

$$h(X; \theta) = \sum_{i=1}^{N} c_i I(X \in R_i) \tag{3.4}$$

where N is the number of leaf nodes (regions) in the tree, c_i is the constant value associated with the leaf node R_i and $I(X \in R_i)$ is an indicator function equals 1 if X falls within the region R_i and 0 otherwise.

3.3.3.3 Extra Trees (E.T.)

The extra trees follows the same step-by-step as Random Forest, using a random subset of features to train each base estimator [83]. Although, the best feature and the corresponding value for splitting the node are randomly selects [85]. Random Forest uses a bootstrap replica to train the model, while the extra trees the whole training dataset to train each regression tree [86].

3.3.3.4 XGBoost

XGBoost (XGB) is a highly optimized distributed gradient boosting library. It employs a recursive binary splitting strategy to identify the optimal split at each stage, leading to the construction of the best possible model [87]. Due to its tree-based structure, XGB is robust to outliers and, like many boosting methods, is effective in countering overfitting, making model selection more manageable. The regularized objective of the XGB model during the t^{th} training step [88] is illustrate in Equation (3.5). Here, $l(y_{pred}^{(t)}, y_{truth})$ represents the loss, which quantifies the disparity between the prediction of the imputed missing value $y_{pred}^{(t)}$ and the corresponding ground truth y_{truth}.

$$L^{(t)} = \sum_{i} l(y_{pred}^{(t)}, y_{truth}) + \sum_{k} \Omega(f_k) \tag{3.5}$$

where $\Omega(f_k)$ is the regularizer representing the complexity of the k^{th} tree.

3.3.3.5 LightGBM

The LightGBM (LGBM) is the gradient boosting decision tree (GBDT) algorithm with gradiente-based one-side sampling (GOSS) and exclusive feature bundling (EFB). The GOSS technique is employed within the context of gradient boosting, utilizing a training set consisting of n instances $\{x_1, ..., x_n\}$, where each instance x_i represents a s-dimensional vector in space χ^s. In every iteration of gradient boosting, we compute the negative gradients of the loss function relative to the model's output, resulting in $\{g_1, ..., g_n\}$. These training instances are then arranged in descending order, based on the absolute values of their gradients, and we select the top-$a \times 100\%$ instances with the largest gradient magnitudes to constitute a subset A [89]. For the complementing set A^c, comprising $(1-a) \times 100\%$ of instances characterized by smaller gradients, a random subset B is extracted, sized at $b \times |A^c|$. The division of instances is subsequently determined by the estimated variance gain concerning vector $V_j(d)$ over the combined subset $A \cup B$, where

$$V_j(d) = \frac{1}{n} \left(\frac{(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i)^2}{n_r^j(d)} \right) \tag{3.6}$$

where $A_l = \{x_i \in A : x_{ij} \leq d\}$, $A_r = \{x_i \in A : x_{ij} > d\}$, $B_l = \{x_i \in B : x_{ij} \leq d\}$, $B_r = \{x_i \in B : x_{ij} > d\}$, and $\frac{1-a}{b}$ is the coefficient used to normalize the the sum of the gradients over B back to the size of A^c.

3.3.3.6 CNN

The CNN for regression was designed based on two sequential 1D-convolutional layers, followed by a max pooling 1D layer

$$Y_{ij} = max[W_2 * (W_1 * x)]_{a,b} \tag{3.7}$$

where Y_{ij} is the output of the max pooling 1D layer, W_1 and W_2 are the weights of the two convolutional layers, x is the input signal, and $\langle * \rangle$ denotes the convolution operation. a ranges from is to $is + k - 1$, and b ranges from js to $js + k - 1$, where k is the pooling size and s is the strides. The network is follower by a flatten layer and a dense layer. In Figure 3.2 is shown the proposed CNN architecture.

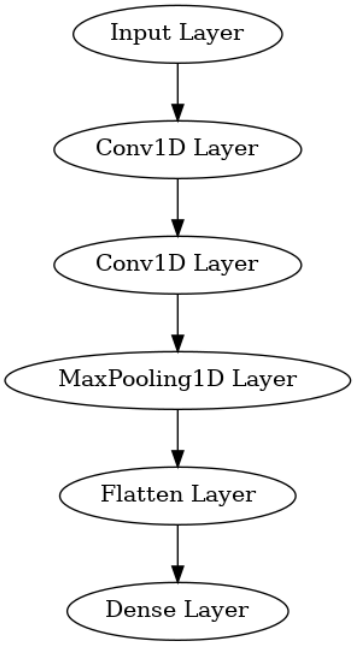


Figure 3.2: Architecture of the proposed CNN.

3.3.3.7 SVR

Given a n training data (Xi, Y_i), where $i = 1, 2, ..., n \subset \chi \times \mathbb{R}$, being χ the space of the input patterns [90]. The goal of the ε-SVR is to find a function which

exhibits a maximum deviation of ε or less from the target values y_i obtained during training, while also maintaining a minimal degree of fluctuation or variability. This function can be described as

$$f(x) = \langle w, x \rangle + b, \text{with } w \in \chi, b \in \mathbb{R} \tag{3.8}$$

where $\langle \cdot, \cdot \rangle$ denotes dot product in χ.

3.3.3.8 Transformer

A Transformer model consists of an encoder and a decoder, each composed of multiple layers of self-attention and feed-forward neural networks. The base structure of the Transformer is the self-attention mechanism. Given an input sequence, the self-attention computes a weighted sum of the values. Multi-head attention is used to capture different aspects of relationships. Let the input of a Transformer layer be $X \in \Re^{n \times d}$, where n is the number of tokens and d is the dimension of each token. Then, one block layer can be a function $f_\theta(X) =: Z$ defined by [91]:

$$A = \frac{1}{\sqrt{d}} XQ(XK)^T \tag{3.9}$$

$$\hat{X} = softmax(A)(XV) \tag{3.10}$$

$$M = LayerNorm_1(\hat{X}O + X) \tag{3.11}$$

$$F = W_2 * (\sigma(W_1 * M + b_1)) + b_2 \tag{3.12}$$

$$Z = LayerNorm_2(M + F) \tag{3.13}$$

where Equations (3.9), (3.10) and (3.11) refereed to attention computation, and Equations (3.12) and 3.13 refereed to the feed forward network layer. $softmax(\cdot)$ is the row-wise softmax function, $LayerNorm(\cdot)$ is the layer normalization function, and σ to activation function. Q, K, V and $O \in \Re^{d \times d}$, $W_1 \in \Re^{d \times d_f}$, $b_1 \in \Re^{d_f}$, $W_2 \in \Re^{d_f \times d}$, $b_2 \in \Re^d$ are the training parameters in the layer [91]. In Figure 3.6 is shown the architecture of the proposed Transformer.

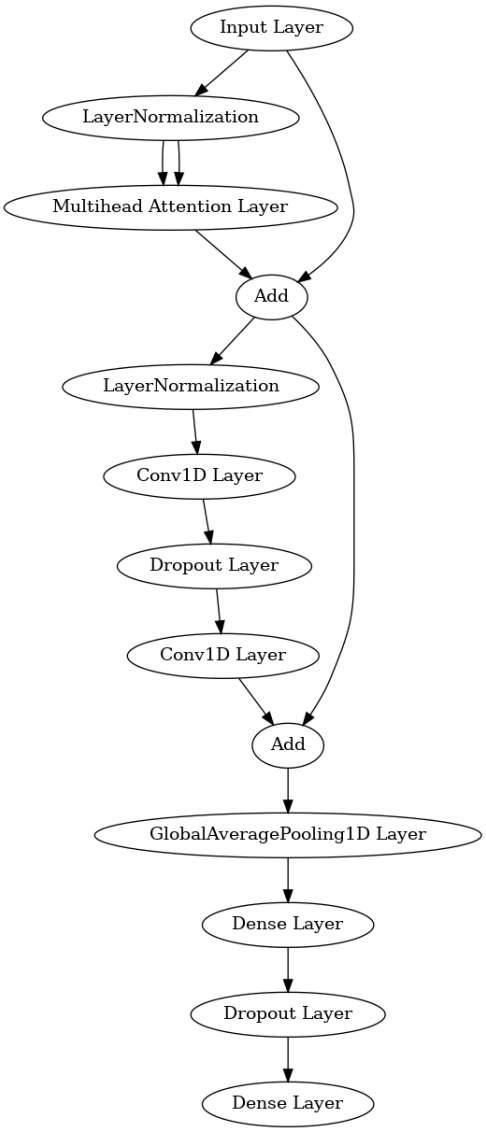


Figure 3.3: Architecture of the proposed Transformer.

3.3.4 Evaluation metrics

The metrics used for evaluation of the regression models are the mean square error, mean absolute error, root mean square error, mean absolute percentage error, R-square and correlation coefficient [90].

- Mean square error (MSE) [90]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.14}$$

where n is the number of data points in the dataset, y_i represents the actual target value of the i-th data point, and $\hat{y}_i$ represents the predicted value of the

i-th data point.

- Mean absolute error (MAE) [90]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.15}$$

where $|\cdot|$ denotes the absolute value.

- Root mean square error (RMSE) [90]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.16}$$

- Mean absolute percentage error (MAPE) [90]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \times 100 \tag{3.17}$$

- R-square (R^2) [81]:

$$R^2 = 1 - \frac{SSE}{SST} \tag{3.18}$$

where SSE represents the sum of squared differences between the the actual target values and the predicted values. The SST represents the total sum of squares, which is the sum of squared differences between the actual target values and their mean.

- Correlation coefficient (C.C.) [92]:

The correlation coefficient is given by the Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^{n}(y_i - y_{mean})(\hat{y}_i - \hat{y}_{mean})}{\sqrt{\sum_{i=1}^{n}(y_i - y_{mean})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \hat{y}_{mean})^2}} \tag{3.19}$$

where y_{mean} is the mean of the actual target values and $\hat{y}_{mean}$ is the mean of the predicted values.

3.4 Experiments and Results

This section presents the experiments conducted and the results achieved for data generation, noise and distance predict.

3.4.1 Data generation

The first stage of the experiments involves signal generation. It is assumed that multiple SU and a single PU are moving at a velocity of $v = 3$ km/h, and their initial positions are randomly chosen within an area of 250 meters $\times$ 250 meters. This configuration was chosen to mimic typical small outdoor environments, such as urban microcells, where user mobility and dense deployment create challenging spectrum sensing conditions. As a result, the users' positions change over a time period of $\Delta t = 5$ seconds. Each occupied band N_B has bandwidth B_W of 10 MHz, and the PU can simultaneously use 1 to 3 bands. Additionally, $P = 23$ dBm, $\beta = 10^{3.453}$, $\alpha = 3.8$, $\sigma = 7.9$ dB, and N_0 is randomly chosen between -114 and -174 dBm/Hz. The ratio of leaked power to adjacent bands, η, is 10 dBm, resulting in leaked power to adjacent bands being half of the PU signal power. For the experiments, $42,000$ instances were generated, divided into 80% for training and 20% for testing. For the CNN and Transformer methods, the training dataset was divided into 80% for training and 20% for validation. A total of $1,024$ samples per second of the signal were generated. In Table 3.3 is presented the parameters and their respective values.

Table 3.3: Parameters and values for data generation.

Parameter	Value	Mean
v	3 km/h	Velocity
Δt	5 seconds	Time period
B_W	10 MHz	Bandwidth of each N_B band
N_{Bp}	1 to 3	B_A
P	23 dBm	Power transmitted by the PU
α	3.8	Path-loss exponent
β	$10^{3.453}$	Path-loss constant
σ	7.9 dB	Standard deviation
N_0	-114 to -174 dBm/Hz	Noise power density
η	10 dBm	Leaked power to adjacent bands

3.4.2 Model parameters

The Random Forest, Decision Tree, Extra Trees, and SVR models were parameterized following the default settings of the *scikit-learn* library. In Tables 3.4 and 3.5, the parameters of the XGB, LGBM, and CNN models are presented. And, in the Table 3.6, the parameters of the Transformer model is presented.

Table 3.4: XGB and LGBM parameters.

Parameter	XGB	LGBM
Estimators	100	100
Max depth	6	-
Learning rate	0.1	0.05
Subsample	0.8	-
Column sample by tree	0.8	-
Random state	42	42
Boosting type	-	GBDT
Number of leaves	-	31

Table 3.5: CNN parameters.

Parameter	CNN
Conv1D layers	2
Kernel size	$3-3$
Strides	$1-1$
Filters	$1024-512$
Optimizer	Adam
Loss	MSE
Batch size	256
Learning rate	0.001

3.4.3 Noise predict

Figure 3.4 (left) presents the graph of the MSE with the variation of the N_0 in the proposed regression methods. It is noticeable that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, Extra Trees, XGB, LGBM, Transformer, CNN and Decision Tree showed similar performances, as shown in the graph. However, the Random Forest, XGB, LGBM and Extra Trees exhibited better performance, especially at lower levels of

Table 3.6: Transformer parameters.

Parameter	Transformer
Head size	32
Number of heads	4
Filter dimension	32
Transformer block	1
Dense layer	32
Drop out	0.25
Batch size	32
Optimizer	Adam
Learning rate	0.001
Loss	MSE

N_0. The XGB was the best model, achieving the lowest level of MSE, achieving 15.53, which indicates that the predicted values are close to the actual ones.

Figure 3.4 (right) presents the graph of the MAE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, Extra Trees, XGB, LGBM, Transformer, CNN and Decision Tree showed similar performances, as shown in the graph. The similarity in performance among these models can be attributed to their shared ability to capture complex patterns in high-dimensional data. However, the slight variations suggest that hyperparameter tuning or adding additional features could lead to improvements. However, the Random Forest, XGB, LGBM and Extra Trees exhibited better performance, especially at levels of -164 dBm/Hz and -154 dBm/Hz. The Random Forest was the best model, achieving the lowest level of MAE, achieving 1.9473, which indicates that the predicted values are close to the actual ones.

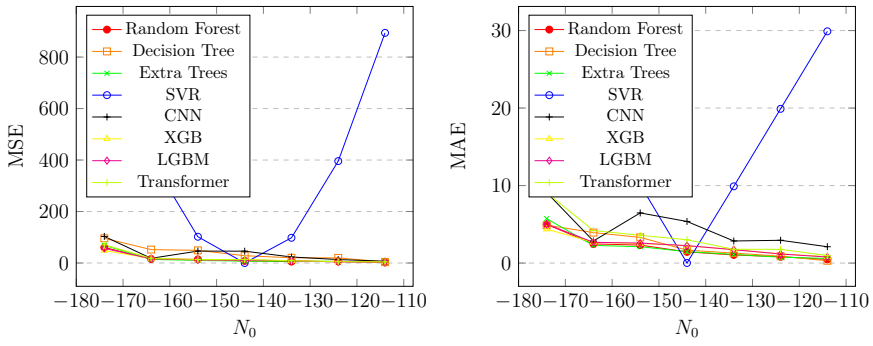


Figure 3.4: Graphic of the MSE (left) and MAE (right) of proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 3.5 (left) presents the graph of the RMSE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. The Decision Tree and CNN only showed similar performance at -114 dBm/Hz. The XGB was the best model, achieving the lowest level of RMSE, achieving 3.94, which indicates that the predicted values are close to the actual ones.

Figure 3.5 (right) presents the graph of the MAPE with the variation of the N_0 in the proposed regression methods. It is noticeable, also, that the SVR had the worst performance, only achieving good results at an N_0 of -144 dBm/Hz. The Random Forest, XGB, LGBM Extra Trees, Transformer, CNN and Decision Tree showed similar performances, as shown in the graph. However, the Random Forest and Extra Trees exhibited better performance. The Random Forest was the best model, achieving the lowest level of MAPE, achieving 1.2534, which indicates that the predicted values are close to the actual ones.

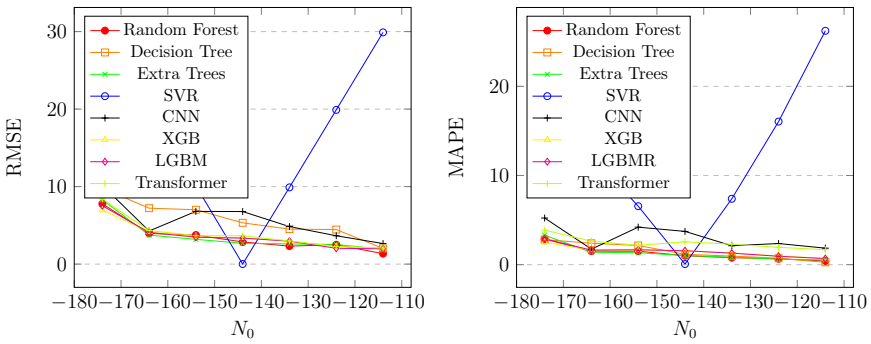


Figure 3.5: Graphic of the RMSE (left) and MAPE (right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

In Table 3.7, the general metrics for all proposed models are presented for noise prediction. It is worth noting that, in terms of the correlation coefficient, Random Forest, XGB, LGBM, Transformer and Extra Trees exhibited the highest and similar performances. However, Random Forest performed slightly better on the MAE and MAPE metrics, while XGB performed better on the correlation coefficient, MSE, RMSE, and R^2. CNN and Decision Tree had similar performances, and SVR presented the worst performance in the general metrics.

Table 3.7: Noise regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	CNN	XGB	LGBM	Transformer
C.C.	0.9801	0.9521	0.9794	0.0079	0.95	**0.9806**	0.979	0.9697
MSE	16.084	38.6547	16.8942	404.64335	36.97	**15.53**	16.35	19.2086
MAE	**1.9473**	2.32738	2.0186	17.2736	4.54	2.23	2.32	3.4854
RMSE	4.0104	6.2172	4.11026	20.11575	6.08	**3.94**	4.04	4.3827
MAPE	**1.2534**	1.49406	1.2971	12.3579	16.15	1.48	1.52	2.4489
R^2	0.96025	0.9044	0.9582	$-4.0934e^{-06}$	0.908	**0.9616**	0.959	0.9365

3.4.4 Distance predict

Figure 3.6 presents the graphics of the MSE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, Transformer, LGBM and Extra Trees showed similar performances, as shown in the graph. However, the LGBM exhibited slightly better performance than the others, especially at the lowest level of noise, achieving 100.67. In the final distance prediction, the Extra Trees model exhibited slightly better performance, achieving 107.77.

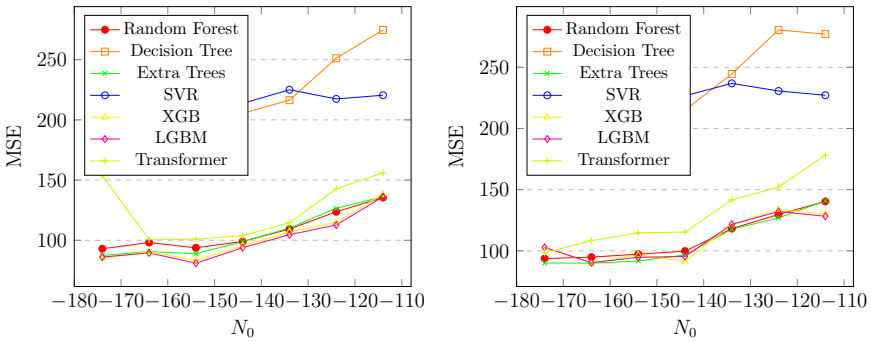


Figure 3.6: Graphic of the MSE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 3.7 presents the graphics of the MAE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is also noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. However, the XGB exhibited slightly better performance than the others, achieving 7.19. In the final distance prediction, the Extra Trees model exhibited

slightly better performance, achieving 7.33.

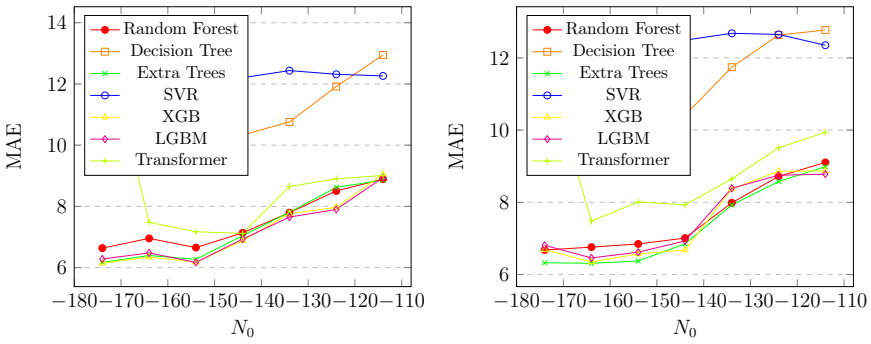


Figure 3.7: Graphic of the MAE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 3.8 presents the graphics of the RMSE of the initial and final distance between the SU and PU during the spectrum sensing with the variation of N_0 in the proposed regression methods. It is also noticeable that the SVR and Decision Tree had the worst performances at all levels of noise. The Random Forest, XGB, LGBM, Transformer and Extra Trees showed similar performances, as shown in the graph. However, the LGBM exhibited slightly better performance than the others, achieving 10.03. In the final distance prediction, the Extra Trees model exhibited slightly better performance, achieving 10.38.

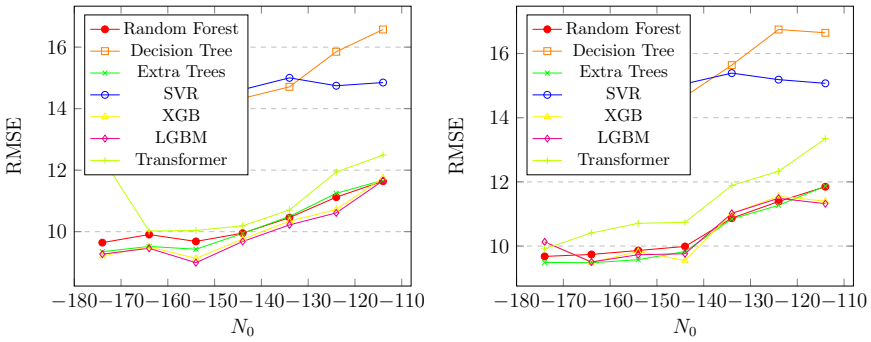


Figure 3.8: Graphic of the RMSE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between -114 dBm/Hz and -174 dBm/Hz.

Figure 3.9 presents the graphics of the MAPE of the initial and final distance between the SU and PU during spectrum sensing with variations in N_0 using the proposed regression methods. It is also noticeable that SVR and Decision Tree had the worst performance levels across all levels of noise. Random Forest and Extra Trees showed similar performance, as shown in the graph on the left. However,

Transformer exhibited slightly better performance than Random Forest, especially at −174 dBm/Hz, −164 dBm/Hz, and −155 dBm/Hz, achieving 30.23. In the graph on the right, Random Forest, XGB, LGBM, and Extra Trees showed similar performance. However, Transformer exhibited slightly better performance than the others, achieving 31.25.

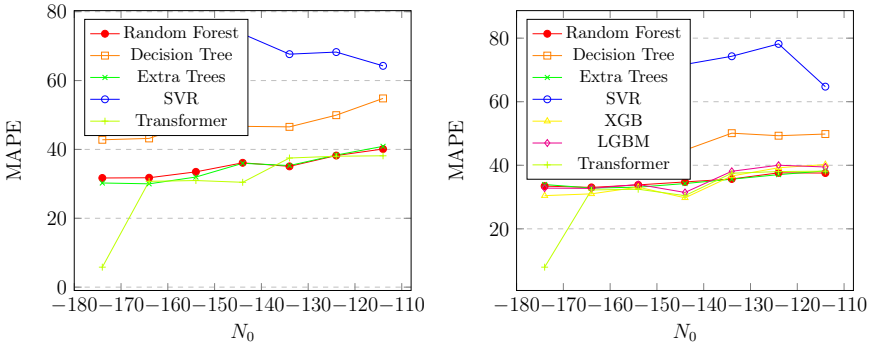


Figure 3.9: Graphic of the MAPE (initial distance at the left and final distance at the right) of the proposed methods with N_0 between −114 dBm/Hz and −174 dBm/Hz.

In Table 3.8, general metrics for all proposed regression models are presented for initial distance prediction. It is worth noting that, in all metrics, Random Forest, XGB, LGBM, Transformer and Extra Trees exhibited the highest and similar performances. Transformer performed better in correlation coefficient, MAPE, and R^2, demonstrating a high level of correlation between the predictions and the actual values of distances. In terms of MSE and RMSE, LGBM achieved the lowest values, indicating that the differences between predictions and actual values are small. The lowest MAE values were exhibited by XGB.

Table 3.8: Initial distance regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	CNN	XGB	LGBM	Transformer
C.C.	0.7222	0.4876	0.7290	0.08103	nan	0.718	0.725	**0.841**
MSE	107.56	226.2186	105.5776	218.8497	nan	102.17	**100.67**	124.63
MAE	7.5115	10.9095	7.31	12.2795	nan	**7.19**	7.20	8.87
RMSE	10.3713	15.0405	10.275	14.7935	nan	10.10	**10.03**	11.16
MAPE	35.2191	47.4401	34.69	68.5288	nan	inf	inf	**30.23**
R^2	0.51807	−0.0135	0.5269	0.00399	nan	0.516	0.523	**0.58**

In Table 3.9, general metrics for all proposed regression models are presented for final distance prediction. The Extra Trees exhibited the best performance in all metrics except for MAPE, R^2 and correlation coefficient, where Transformer

exhibited the best performance.

Table 3.9: Final distance regression comparison metrics of the proposed methods.

Metrics	R.F.	D.T.	E.T.	SVR	CNN	XGB	LGBM	Transformer
C.C.	0.7225	0.47	0.7321	0.0923	nan	0.712	0.714	**0.821**
MSE	110.61	245.51	**107.77**	228.4979	nan	109.51	109.36	129.805
MAE	7.58	11.405	**7.33**	12.45	nan	7.49	7.54	9.06
RMSE	10.51	15.66	**10.38**	15.1161	nan	10.46	10.45	11.39
MAPE	35.13	47.78	35.01	74.74	nan	34.36	35.52	**31.25**
R^2	0.5182	−0.0069	0.5306	0.004	nan	0.5081	0.5083	**0.573**

3.5 Discussion

In 3.4.3, the results for noise predict were presented, and several important aspects need to be highlighted. In Figure 3.4 (left), it is noticeable that the models achieved similar performances, except for Decision Tree, CNN and SVR. Interestingly, the best performance occurred at the highest levels of noise, except for SVR, which specialized in a single noise level, −144 dBm/Hz. Figures 3.4 (right), 3.5 (left), and 3.5 (right) exhibit similar characteristics to Figure 3.4 (left). The overall results for Random Forest, XGB, LGBM, Transformer and Extra Trees are similar across all metrics, with Random Forest and XGB performing particularly well, as demonstrated in Table 3.7. In contrast, SVR exhibited poor performance overall. Additionally, the correlation coefficients reveal that Random Forest, XGB, LGBM and Extra Trees achieved strong correlations between predicted and real noise values (Table 3.7). Due to the limited computational resources and the large amount of generated data, it was not possible to enhance the robustness of the CNN and Transformer architectures. Furthermore, a search for optimal hyperparameters values for all models could be a proposed enhancement for the work.

In 3.4.4, the results for initial and final distance predict were presented, and several notable observations arise. In Figure 3.6, it is evident that Extra Trees and LGBM achieved the highest performances. Unlike the noise predict, the best performances for both distance predicts occurred at the lowest levels of noise for all models, which was expected. The behavior of predict the initial distance mirrors that of predict the final distance, as seen in Figures 3.6, 3.7, 3.8, and 3.9. The general results for Random Forest, XGB, LGBM, Transformer and Extra Trees

exhibit similarity across all metrics, with Extra Trees, Transformer and LGBM slightly outperforming in almost all metrics for both initial and final distance, as shown in Tables 3.8 and 3.9, respectively. In contrast, Decision Tree and SVR showed poor performance overall. Moreover, the correlation coefficients reveal that the predicted distances exhibited indices greater than 0.82 in relation to the real values in both distances, initial and final. Interestingly, the CNN architecture used for noise prediction was the same as the one used for distance prediction; however, the model was unable to perform. Another interesting factor is that in the initial distance prediction, the MAPE values for the XGB and LGBM models returned 'inf'. More studies are needed to understand what happened.

3.6 Conclusion

In this paper, noise and distance prediction based on spectrum sensing signals using regression models was proposed. The conducted experiments have shown that the proposed methods hold promise for predict noise levels, as well as the initial and final distances between the PU and SU. Future research can explore integrating these models with reinforcement learning techniques to dynamically adapt predictions in real-time, further enhancing their applicability in rapidly changing environments.The correlation coefficient value for XGB is the highest and closest to one (Table 3.7), indicating a strong correlation between predicted and actual noise values in the test database. As a result, the proposed methods can greatly benefit various applications, especially in telecommunication and networking, enabling the design of communication systems that meet appropriate requirements for ensuring reliable and efficient data transfer.

Additionally, the predicts for the initial and final distances between PU and SU are presented as results. The conducted experiments have demonstrated that the proposed methods show promise in predicting distances between users. The correlation coefficient values for Transformer are the highest, exceeding 0.82 (Tables 3.8 and 3.9), which implies a good level of correlation between the predicted distances and the actual distances in the test database. It's important to highlight that the number of possible noise levels is limited to 7 levels (-114 to -174 dBm/Hz), while

the number of possible distances is unknown since the distance was chosen randomly within a certain range of the area. Therefore, there may be a difference in performance between the two approaches. Hence, the proposed methods hold the potential to benefit numerous applications, including signal attenuation and path loss, interference and frequency reuse, fading and multipath effects, localization and tracking, and power control.

Chapter 4

Spoofing Deep Cooperative Spectrum Sensing Using Generative Adversarial Network

4.1 Introduction

The dynamism of spectrum sensing depends on the efficiency of cognitive radio (CR) [5,6,35]. Cooperative spectrum sensing utilizes information from several secondary users (SUs) to increase the probability of detecting the primary user (PU) in the sensed channels [1, 4, 63]. However, with the rapid development of generative adversarial networks (GANs), malicious users (MUs)—defined as entities that intentionally interfere with the spectrum by simulating legitimate primary user (PU) signals—can emulate the signal of the PU, disrupting cooperative spectrum sensing and reducing the overall efficiency of cognitive radio (CR) systems. This type of interference, known as PU spoofing, not only diminishes the performance of CR but also poses significant security challenges for ensuring reliable spectrum access [48,54,55]. Therefore, the use of these networks to prevent SUs from accessing the frequency spectrum raises questions that lead to debates about security in the spectrum, which is, even more recently, a major point of concern. Thus, this chapter aims to explore the vulnerabilities of cooperative spectrum sensing under GAN-based spoofing attacks, demonstrating how such attacks can compromise the decision-making models of cognitive radio systems. The objective is to comprehensively

analyze the threats posed by GANs and propose countermeasures to strengthen spectrum security.

GANs are powerful networks that can modify and create new data. They are composed of a generator and a discriminator network and operate in a competitive manner, where the generator aims to produce realistic data mimicking authentic data, while the discriminator distinguishes between real and generated data [37, 93, 94]. These networks are the basis for some of the most current tools for creating artificial image data, such as DALL·E [95, 96]. With this versatility, creating or modifying data for malicious purposes has become more dangerous. One of the most recent examples is deepfakes, used to deceive and manipulate individuals [97].

In this context, GANs can generate synthetic signals that closely resemble those encountered in real-world environments, facilitating various applications such as spectrum sensing, channel modeling, protocol testing, and spoofing spectrum sensing [54,55]. Moreover, the flexibility of GANs allows for the generation of signals across different frequency bands, bandwidths, and modulation schemes, providing researchers and engineers with a versatile tool for exploring and experimenting with wireless communication systems. However, the use of GANs in signal generation also raises concerns about security and trustworthiness, as malicious users could exploit this technology to create deceptive signals for nefarious purposes, underscoring the importance of robust authentication and verification mechanisms in communication networks.

An initial research was conducted using one variant of GAN, a semi-supervised GAN (SGAN), to spoof state-of-the-art models for automatic modulation recognition (AMC) [48]. In this paper, utilizing the RML2016.10a dataset [98], we trained an SGAN to generate false modulated signals. The significant distinction between GAN and SGAN is that the SGAN's discriminator has two outputs: one supervised, used for classification, and the other unsupervised, used to discriminate between real and false signals. In the best-case scenarios, where the signal-to-noise ratio (SNR) is higher, the modulated signals generated by the SGAN were able to deceive over 70% of three other AMC models presented in the literature. This preliminary research shows that the approach of using GANs to deceive other artificial intelligence (AI) models is feasible and opens up ideas for employing such an

approach for cooperative spectrum sensing.

Cooperative spectrum sensing creates a cooperative matrix with information gathered from several secondary users (SUs). These matrices contain sensing information from multiple sensed channels. At the fusion center, a trained model makes decisions about the channel conditions based on these matrices. In this chapter, we propose the use of GANs to generate fake signals and consequently fake cooperative matrices in order to deceive the decision-making models at the fusion center. This method has shown that, with GANs, malicious users (MUs) can deceive these decision-making models at the fusion center with a considerable rate of success. This raises questions about the security of the frequency spectrum and sparks debates about how to protect and ensure trustworthiness in the system.

4.2 Related Works

In [99], the authors proposed the use of an adversarial machine learning approach to develop decision-making algorithms resistant to attacks. As a defense strategy, they deliberately programmed the transmitter to take incorrect actions to mislead potential attackers. They employed a neural network based on Microsoft CNTK for the transmitter algorithm. The attack algorithm is based on a feedforward neural network. The results showed that the attack algorithm was effective in reducing the transmitter throughput, while the defense strategy proved to be efficient in misleading the attacker.

In [45], the authors conducted a study on the use of GANs in next-generation (NextG) communications for spectrum sharing, anomaly detection, and mitigating security attacks. In terms of security, the authors demonstrate the significance and the wide range of potential applications for GANs in this field. The simulations show that these networks are powerful tools for deceiving decision-making models at the receiver and assisting in anomaly detection, which helps increase the efficiency and dynamics of spectrum allocation.

In [100], the authors employed a GAN-based approach to detect rogue transmitters in radio frequency. The proposed GAN learns from known real transmitted signals to generate fake data. The discriminator trained was able to distinguish

between real and fake signals in approximately 99% of the cases. Additionally, they trained a convolutional neural network (CNN) and a fully connected deep neural network (DNN) to classify trusted transmitters, achieving approximately 81% and 91% accuracy, respectively. This work is closely related to the research presented here, as both utilize GANs to manipulate the signal environment. Still, it focuses on detecting anomalies rather than performing spoofing attacks. Thus, the current chapter builds on these findings by shifting the focus from detection to direct manipulation of cooperative spectrum sensing matrices to evaluate the resilience of CR systems under adversarial conditions.

The authors in [55], proposed a GAN-based approach to spoofing wireless networks. They utilized GANs to generate signals indistinguishable from real ones, aiming to simulate attacks on classification models in the receiver. They applied multiple-input and multiple-output (MIMO) techniques and studied their impact on the system. Additionally, they evaluated the success of the attack based on the movement of the transmitter. In the best scenarios, where the transmitter was closer to the receiver, the success probability of spoofing was approximately 76%. However, at greater distances, the success probability of spoofing decreased drastically.

4.3 Metodology

This section presents the proposed methods for spoofing deep cooperative spectrum sensing.

4.3.1 System model

In cooperative spectrum sensing, several SUs share information on spectrum sensing with a fusion center, where decisions about the presence or absence of a PU in the evaluated channels are made. This information is organized in cooperative matrices $[N_{SU} \times N_B]$, where N_{SU} is the number of SUs cooperating in the system, and N_B is the number of bands evaluated by each SU. Due to the 2D nature of these matrices, deep learning models, as shown in the literature [1, 4, 63, 101, 102], have proven promising for distinguishing the presence of a PU in the evaluated channels. However, if a MU mimics PU signals, and consequently, these cooperative matrices

in the fusion center, the efficiency of the CR will be drastically affected since these matrices could contain information from several SUs and channels.

We propose the use of GANs to generate fake signals and consequently fake cooperative matrices to simulate attacks on a fusion center, deceiving the decision-making model that evaluates wideband spectrum sensing. For this purpose, the proposed method is divided into several steps: (1) signal generation; (2) training classical machine learning approaches for individual spectrum sensing; (3) generating cooperative matrices; (4) training deep cooperative spectrum sensing models used in the fusion center to evaluate whether there is a presence of PU in the cooperative information; (5) training the proposed GAN to create fake signals and consequently fake cooperative matrices to simulate attacks; and (6) evaluating the success probability of spoofing deep cooperative spectrum sensing in the fusion center. Figure 4.1 presents the flow of development from step (1) to step (4) of the proposed methodology.

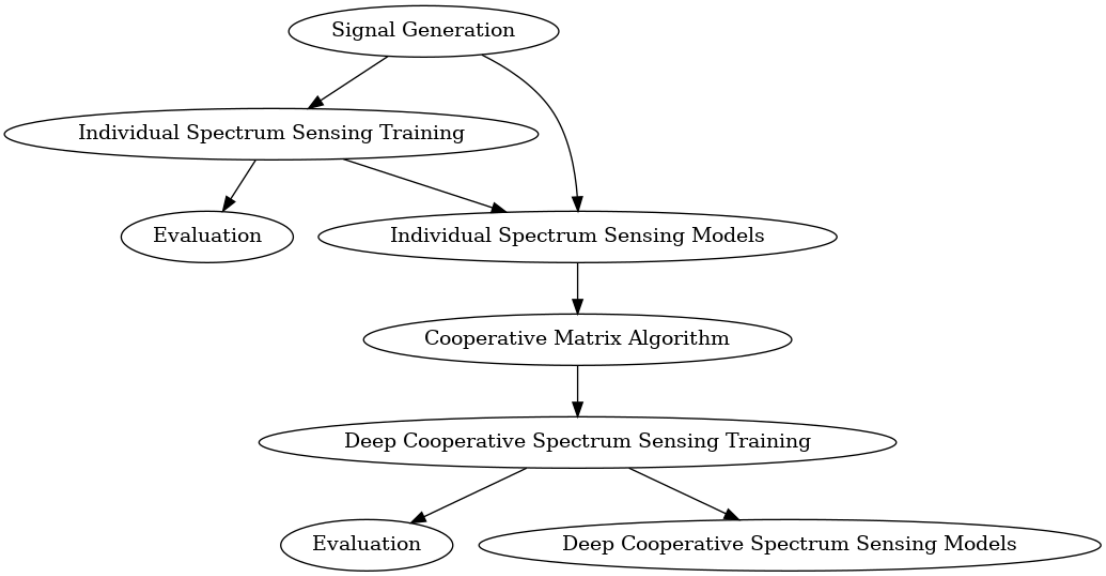


Figure 4.1: Scheme of the proposed method from step (1) to step (4).

Figure 4.2 presents the flow of development for step (5) and (6) of the proposed methodology. In this scheme, note that just PU signal is generated, and the cooperative matrices are composed by the output of the models given the fake signals generated by the GAN as input. The deep cooperative models will only attempt to recognize fake cooperation matrices that contain PU signals.

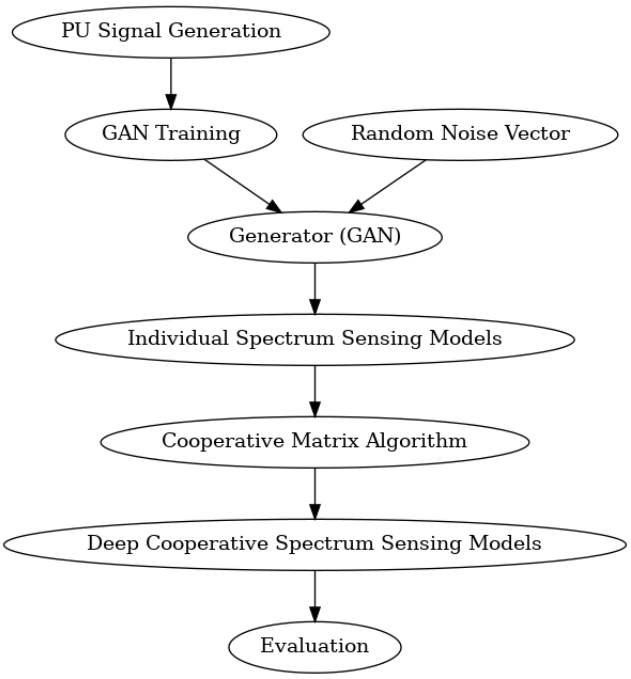


Figure 4.2: Scheme of the proposed method for step (5) and (6).

4.3.2 Signal generation

In the spectrum sensing process, the decision about the channel condition is binary, and two hypotheses are considered, H_1 and H_0, where H_1 represents the signal with PU presence and H_0 without PU presence. For signal generation, it is assumed that N_{SU} and a single PU move at a velocity v, and their initial locations are random in a certain area, so their locations change over a time period Δt. It is also considered N_B bands, with B_W being the bandwidth. Additionally, it is assumed that the SU has no information about which bands are used by the PU, and the PU can use N_{B_P} consecutive bands. Then, the signal received by SU i in band j at time n can be described as [1, 63]:

$$y_i^j(n) = \begin{cases} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_P \\ \sqrt{\eta} s_i^j(n) + w_i^j(n), & \text{for } H_1 \text{ and } j \in B_A \\ w_i^j(n), & \text{for } H_0 \end{cases} \tag{4.1}$$

where $s_i^j(n) = \kappa_i(n) g_i^j(n) x(n)$ and $w_i^j(n)$ is the Gaussian white noise (AWGN) with zero mean, and standard deviation $\sigma = \sqrt{B_W 10^{\frac{N_0}{10}}}$. And N_0 is the noise power density in dbm/Hz. Let η be the proportion of power leaked to adjacent bands, then B_P are the bands occupied by the PU, and B_A are the bands affected by power

leakage from the PU.

In $s_i^j(n)$, we have the simplified model and path, which can be written as:

$$\kappa_i(n) = \sqrt{\frac{P}{\beta(d_i(n))^{\alpha} 10^{\frac{h_i(n)}{10}}}} \tag{4.2}$$

where α and β are the path loss exponent and path loss constant, respectively. $d_i(n)$ is the Euclidean distance between the PU and the SU i at time n. The shadow fading of the channel, $h_i(n)$, between the PU and the SU i at time n in dB, can be described as a normal distribution with zero mean and variance σ^2, and P is the power transmitted by the PU in a specific band. Also, the multipath fading, $g_i^j(n)$, is modeled as an independent circularly symmetric complex Gaussian random variable (CSCG) with zero mean. Finally, $x(n)$ is the data transmitted by the PU at time n with an expected value of $x(n)$ equal to 1.

A special type of filter that shifts the phases of a signal while leaving all the amplitudes of the spectral components unchanged is the Hilbert transform [103].

$$\mathcal{H}\{y(n)\} = \frac{1}{\pi} \int_{m=-\infty}^{\infty} \frac{y(m)}{n-m} dm \tag{4.3}$$

We applied the Hilbert transform to better highlight singular information from the signals. Sequentially, we modulated the signals at a frequency f_c:

$$a(n) = \left| \mathcal{H}\{y(n)\} e^{i2\pi f_c n} \right| \tag{4.4}$$

where $a(n)$ is the output of the signal generation step.

4.3.3 Individual spectrum sensing

For individual spectrum sensing, we propose using several classical supervised machine learning methods. To discover the best architectures for classifying noise and PU signals, we utilized the Lazy Predict library. This library helps build numerous basic models with minimal code and aids in understanding which models perform better without requiring parameter tuning [104, 105]. For the classification task, we trained and evaluated more than 30 models, including XGBoost (XGB), LightGBM (LGBM), Random Forest, Bagging Classifier, Decision Tree, Extra Trees

Classifier, Extra Tree Classifier, SGD Classifier, AdaBoost Classifier, among others. Of these models, the five with the best performance are chosen to classify whether the generated signal $a(n)$ contains the presence of a PU or not.

4.3.4 Cooperative matrix generation

Two groups of cooperation matrices will be generated:

1. In this group, some bands will have PU signals, while the remaining bands will have noise signals.

2. In this group, all bands will have noise signals only.

The format of the matrices is given by $[N_{SU} \times N_B]$, where N_{SU} and N_B are fixed. Thus, each SU in the cooperation senses the bands and uses the trained models. The models return 1 if there is a presence of PU in the channel and 0 if there is no presence of PU. Therefore, when this information is shared, the cooperation matrix consists of 1s and 0s in a 2D format.

In (1), since the SU doesn't know in which bands the PU may be present, the generation of the matrices is done randomly. For each generated matrix, a band is randomly chosen, and a different random output result of the classifier, which had a PU signal as input, is selected for each cooperating SU. The number of bands affected by power leakage is also randomly chosen, respecting adjacency to the initial position of the PU. In Algorithm 2, the logic for creating the cooperation matrix is presented. The number of occupied bands (N_{OB}) is randomly chosen between 1 and N_B. The notation $a(n) \rightarrow B_P$ represents a random PU signal, while $a(n) \rightarrow B_A$ represents a signal leaked to adjacent bands of the chosen PU signal. Each $a(n)$ signal is randomly created with unknown N_0 and distances. The variable *prediction* denotes the output of the model for individual spectrum sensing, which is binary, consisting of 0s and 1s. The other bands are filled with output results of the classifier, which had a noise signal as input. In (2), all bands are filled with output results of the classifier, which had a noise signal as input only.

Algorithm 2 Cooperation Matrix Algorithm for (1)

function COOPERATION MATRIX (CM)
  $CM \leftarrow matrix(N_{SU}, N_B)$
  $N_{BO} \leftarrow random(1, N_B)$
  $positions \leftarrow choice(N_{SU}N_B, size = N_{BO})$
  for *pos* in *positions* **do**
    $row, col \leftarrow divmod(pos, N_B)$
    $CM[row, col] \leftarrow predict(a(n) \rightarrow B_P)$
    $direction \leftarrow choice([left, right, both, none])$
    if *direction* is $left$ **then**
      if $col > 0$ **then**
        $CM[row, col - 1] \leftarrow predict(a(n) \rightarrow B_A)$
      end if
    else if *direction* is *right* **then**
      if $col < N_B - 1$ **then**
        $CM[row, col + 1] \leftarrow predict(a(n) \rightarrow B_A)$
      end if
    else if *direction* is *both* **then**
      if $col > 0$ **then**
        $CM[row, col - 1] \leftarrow predict(a(n) \rightarrow B_A)$
      end if
      if $col < N_B - 1$ **then**
        $CM[row, col + 1] \leftarrow predict(a(n) \rightarrow B_A)$
      end if
    else if *direction* is *none* **then** $None$
    end if
  end for
  return CM
end function

4.3.5 Deep cooperative spectrum sensing

In this step, some deep neural architectures from literature [1,63,101,102] will be trained. These models are used to make the final decision about the conditions of the channels sensed by the SUs in the cooperation. The goal is to simulate attacks based on GAN on these models, and evaluate how sensitive they are to MUs.

In [1], the ResNet proposed is a simplified architecture with two residual units. A residual unit is designed to address the vanishing gradient problem and enable the training of very deep neural networks. The residual unit can be described as

$$y = W_2 * ReLU(W_1 * x) + x \tag{4.5}$$

where y is the output of the residual unit, x is the input of the residual unit, W_1 and W_2 are the weights from two convolutional layers, and $\langle * \rangle$ denotes the convolution operation. $ReLU$ is the Rectified Linear Unit. A residual unit applies two convolutional layers to the input, and then adds the original input to the result of these layers, enabling the network to learn identity mappings more easily and alleviating the vanishing gradient problem [106].

The network proposed by [63] is composed of three convolutional blocks. Each block consists of a 2D convolutional layer with $ReLU$ activation function, followed by a max pooling 2D layer, described as

$$Y_{ij} = max[ReLU(W_1 * x)]_{a,b} \tag{4.6}$$

where Y_{ij} is the output of the max pooling 2D layer, W_1 is the weights of the convolutional layer, and x is the input signal. a ranges from is to $is + k - 1$, and b ranges from js to $js + k - 1$, where k is the pooling size and s is the strides. The network is follower by a flatten layer, a dense layer with $ReLU$ activation function, and dense with softmax activation function.

In [101], the authors used a ShuffleNetV2 for cooperative spectrum sensing. ShuffleNetV2 is a lightweight convolutional neural network architecture designed for efficient computation on low-cost devices. The core idea behind ShuffleNetV2 is the channel split operation, where the input xx is split into two branches x_1 and x_2. The branch x_1 can be described as

$$y_1 = ReLU(W_3 * \theta(W_2 * ReLU(W_1 * x_1))) \tag{4.7}$$

where y_1 is the output of the first branch given x_1 as input. W_1, W_2, and W_3 are the weights of the three convolutional layers, and θ represents the depthwise convolution operation. After the transformations, the two branches are concatenated along the channel dimension:

$$y = concat(y_1, x_2) \tag{4.8}$$

To ensure that information is evenly distributed across channels, a channel shuffle operation is applied. This combination in ShuffleNetV2 leads to an optimized

network and improved accuracy.

And, lastly, [102] proposes a reinforcement learning model for cooperative spectrum sensing. The proposed network is called Deep Q-Network (DQN). The DQN combines deep neural networks with Q-learning, which are the core of the reinforcement learning algorithm. The Q-values are updated using the target network and the loss function for training the Q-network [107]. The target network can be described as

$$y_j = r_j + \gamma[max_{\hat{a}} Q(\hat{s}_j, \hat{a} : \theta^-)] \tag{4.9}$$

where y_j is the target Q-value given r_j the reward received, $\hat{s}_j$ the next state, and $\hat{a}$ the next action. θ^- are the parameters of the deep neural network and γ is the discount factor. And the loss function is the mean square error (MSE) described as

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - Q(s_j, a_j : \theta))^2 \tag{4.10}$$

where $Q(s_j, a_j : \theta)$ are the predicted Q-values.

4.3.6 Proposed GAN

GANs are networks that can modify and create synthetic data. These networks consist in two neural networks, a generator G and a discriminator D, which contest in a zero-sum game [108]. In this setup, the generator G and discriminator D are trained in a two-player adversarial game:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

where $D(x)$ represents the discriminator's probability estimate that the input x is from the real data distribution $p_{\text{data}}(x)$. $G(z)$ is the generator's output when given a noise vector z sampled from a prior distribution $p_z(z)$.

The discriminator D is a binary classifier tasked with distinguishing between real data (sampled from $p_{\text{data}}(x)$) and fake data (generated by the generator G). It aims to:

- Maximize $\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]$: This term represents the expected value of the

log probability that D correctly classifies real data x as real (i.e., $D(x) \approx 1$).

- Maximize $\mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$: This term represents the expected value of the log probability that D correctly classifies generated (fake) data $G(z)$ as fake (i.e., $D(G(z)) \approx 0$).

The generator G attempts to generate realistic data that fools the discriminator D. Its objective is to:

- Minimize $\mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$: This forces G to generate data such that $D(G(z)) \approx 1$, meaning that the discriminator mistakenly classifies generated data as real.

The Minimax Game:

- The discriminator D tries to maximize the objective by improving its ability to correctly classify real data as real and generated data as fake.
- The generator G tries to minimize the objective by generating more realistic data that can fool the discriminator into classifying it as real.

In the training process the discriminator D is trained to maximize $\log D(x)$ for real data and $\log(1 - D(G(z)))$ for generated data, improving its classification accuracy. The generator G is trained to minimize the loss function by making $D(G(z)) \approx 1$, which forces it to generate more realistic data to fool the discriminator. Thus, the GAN training process involves a back-and-forth game between G and D, where both models improve their performance over time. The minimax nature of the loss function reflects this adversarial relationship.

The generator, G, proposed for fake signal generation is composed by the followed layers shown in the Figure 4.3. The input of the G, is the random noise vector, and the output is the fake signal.

The discriminator, D, proposed for distinguish real and fake data is composed by the followed layers shown in the Figure 4.4. The input of the D, are the real and fake signals, and the output is the binary decision if the input signal is real or fake.

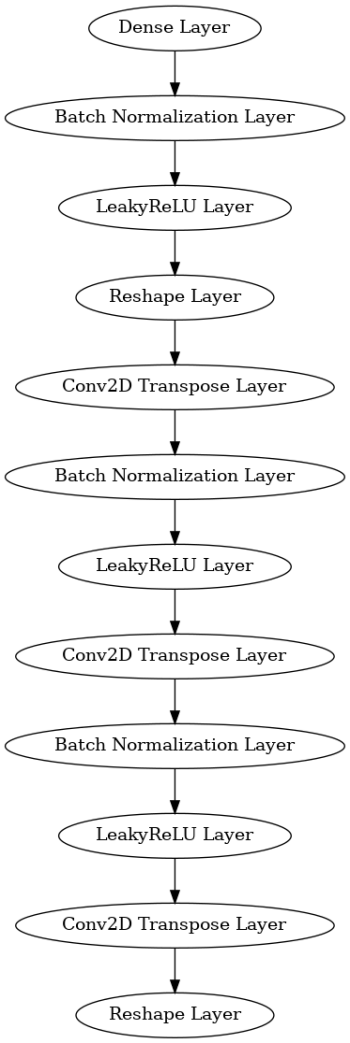


Figure 4.3: Generator architecture of the proposed GAN.

4.3.7 Simulation attack and evaluation

In Figure 4.5, the simulation of an attack on the deep cooperative spectrum sensing models is shown. At this stage, all models are already trained. The MU uses the GAN generator to create fake PU signals by inputting a random noise vector. These signals are then fed into the individual spectrum sensing models that achieved higher accuracy (Random Forest, Extra Trees, Bagging, LGBM, and XGB). If the fake signals are similar to the real ones, the output of these models will mostly be 1s, indicating the presence of PU signals. An evaluation of these models is conducted by comparing the metrics with those achieved using real signals.

The next step is creating the cooperative matrices. Each matrix has a shape of $[N_{SU} \times N_B]$, constructed with 1s and 0s, representing the output of the individual spectrum sensing models. These individual models create the fake matrices following

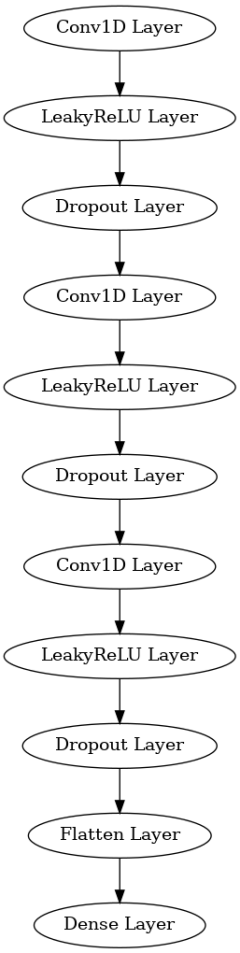


Figure 4.4: Discriminator architecture of the proposed GAN.

Algorithm 2, providing more diversity since each model can output different results for the same signal. However, if the models are easily deceived by fake signals, the construction of these matrices will be strongly impacted. Finally, the attack on these deep models (ResNet [1], CNN [63], ShuffleNetV2 [101], and DQN [102]) used for the final decision at the fusion center is conducted using these fake cooperative matrices. An evaluation is then made to determine how sensitive these models are to fake data.

4.4 Experiments and results

In this section, the results and analysis of the proposed method for spoofing deep cooperative spectrum sensing are presented. Following the methodology, we divided the results into two steps: one related to the flow presented in Figure 4.1, and the other related to the flow presented in Figure 4.2.

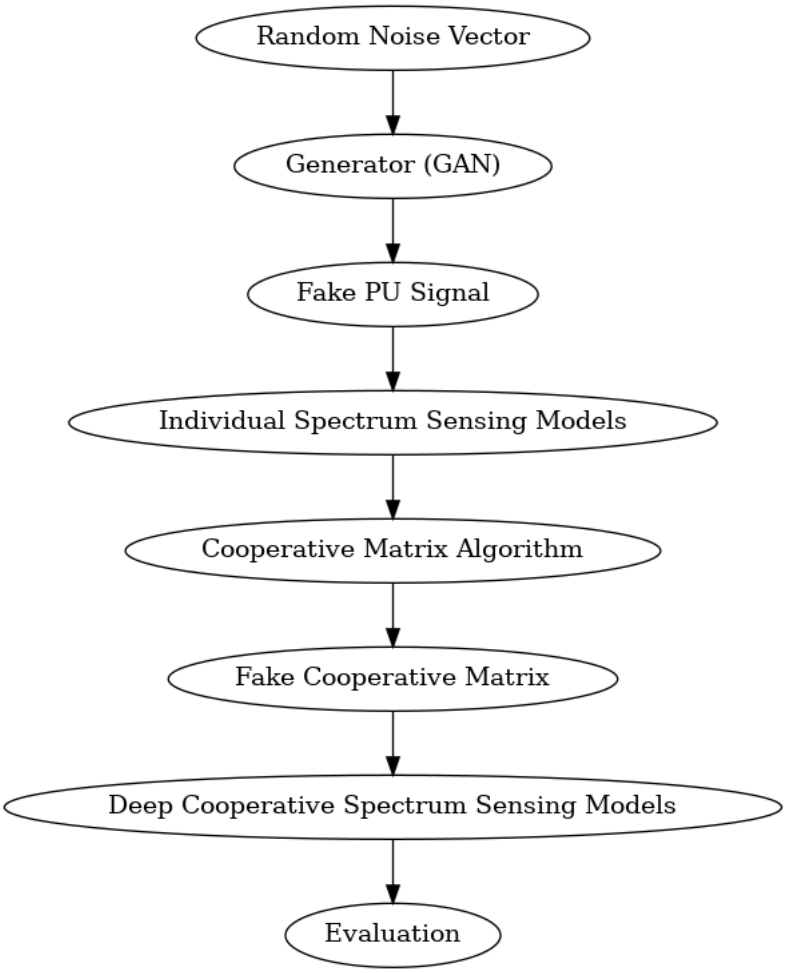


Figure 4.5: Block diagram for the simulation of attack and evaluation.

4.4.1 Deep cooperative spectrum sensing

4.4.1.1 Signal generation

The first stage of the experiments involves signal generation. It is assumed that multiple SUs and a single PU are moving at a velocity of $v = 3$ km/h, and their initial positions are randomly chosen within an area of 250 meters $\times$ 250 meters. As a result, the users' positions change over a time period of $\Delta t = 5$ seconds. Each occupied band has bandwidth B_W of 10 MHz, and the PU can simultaneously use 1 to 3 bands. Additionally, $P = 23$ dBm, $\beta = 10^{3.453}$, $\alpha = 3.8$, $\sigma = 7.9$ dB, and N_0 is randomly chosen between -114 and -174 dBm/Hz. The ratio of leaked power to adjacent bands, η, is 10 dBm, resulting in leaked power to adjacent bands being half of the PU signal power. The carrier frequency used is 2.412 GHz, which is widely employed in various wireless communication standards, including Wi-Fi and Bluetooth. $1,024$ samples per second of the signal were generated. For the

experiments, 42,000 instances were generated, divided into 80% for training and 20% for testing. In Figure 4.6, is shown a example of real signal generated.

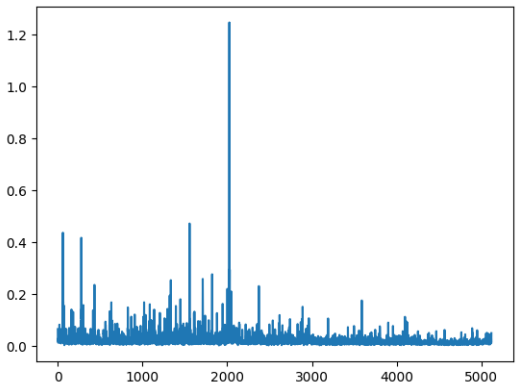


Figure 4.6: Example of real signal generated.

4.4.1.2 Individual spectrum sensing metrics

In Table 4.1, the accuracy, balanced accuracy, receiver operating characteristic area under the curve (ROC AUC), and F1 score metrics are presented using the Lazy Predict library. For the following steps of the method, we selected the five best performing models, which are LGBM, XGB, Random Forest, Bagging, and Extra Trees, respectively, based on their performances.

In Figure 4.7, we present the accuracy over the noise level. It is noticeable that below -154 dBm/Hz, the models almost exhibit the same performance. At higher levels of noise, Random Forest demonstrated the best performance, while at other levels, LGBM exhibited the best performance.

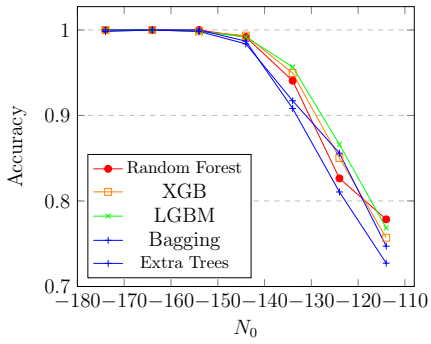


Figure 4.7: Graph showing the accuracy of the five best-performing models with N_0 ranging between -114 dBm/Hz and -174 dBm/Hz.

Table 4.1: Metrics of the best models trained with the Lazy Predict library.

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
LGBM	0.94	0.94	0.94	0.94
XGB	0.93	0.93	0.93	0.93
Random Forest	0.93	0.93	0.93	0.93
Bagging	0.93	0.93	0.93	0.93
Extra Trees	0.92	0.92	0.92	0.92
Ada Boost	0.90	0.90	0.90	0.90
Decision Tree	0.89	0.89	0.89	0.89
K-Neighbors	0.87	0.87	0.87	0.87
Extra Tree	0.83	0.83	0.83	0.83
Passive Aggressive	0.75	0.75	0.75	0.73
QDA	0.74	0.74	0.74	0.74
LDA	0.72	0.72	0.72	0.71
Linear SVC	0.70	0.70	0.70	0.69
Ridge	0.65	0.65	0.65	0.60
Logistic Regression	0.65	0.65	0.65	0.63
Ridge CV	0.62	0.62	0.62	0.55
SGD	0.61	0.61	0.61	0.59
NuSVC	0.58	0.58	0.58	0.50
Calibrated CV	0.54	0.54	0.54	0.42
Gaussian NB	0.54	0.54	0.54	0.42
Label Propagation	0.52	0.52	0.52	0.38
Label Spreading	0.52	0.52	0.52	0.37
Bernoulli NB	0.52	0.52	0.52	0.37
Nearest Centroid	0.50	0.50	0.50	0.34
SVC	0.50	0.50	0.50	0.34
Dummy	0.50	0.50	0.50	0.33
Perceptron	0.50	0.50	0.50	0.34

4.4.1.3 Cooperation matrices

The cooperation matrices are formed by N_{SU} and N_B, which are 50 and 32, respectively. For each of the five previous trained models for individual spectrum sensing, a group of cooperation matrices is created. Since each model has a different output, this increases the diversity of the data. The cooperative matrices consist of 0s and 1s. In Tables 4.2 and 4.3, we present one example of a matrix representing bands with the presence of a PU and one example of a matrix representing the absence of a PU. Notice that in Table 4.2, almost all lines indicate the presence of a PU, and in some cases, even leaked power is identified. In Table 4.3, only some lines can identify the presence of a PU, indicating model mistakes. To train deep cooperative models, we created 17,500 cooperation matrices of each model:

43,750 matrices representing bands with PU signals and 43,750 matrices without the presence of PU, totaling 87,500 matrices divided into training (80%), validation (10%), and testing (10%).

Table 4.2: Cooperation matrix with the presence of PU, $N_{SU} = 50$ and $N_B = 32$.

0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1
1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1
0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

4.4.1.4 Deep cooperative spectrum sensing metrics

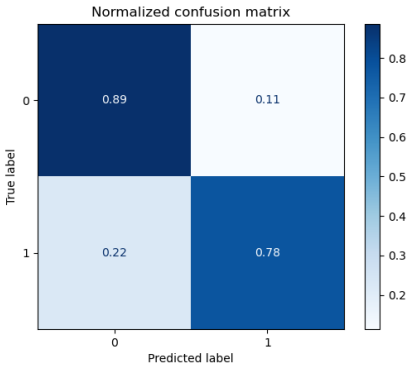
Four architectures were proposed for deep cooperative spectrum sensing: ResNet [1], CNN [63], ShuffleNetV2 [101], and DQN [102]. The ResNet, CNN, and ShuffleNetV2 were trained as classification models, while DQN a reinforcement learning model. The classification models were trained using the Adam optimizer with a learning rate of 0.001, categorical cross-entropy as the loss function, accuracy as the metric, a batch size of 1,024, 1,000 epochs, and early stopping with a patience

Table 4.3: Cooperation matrix without the presence of PU, $N_{SU} = 50$ and $N_B = 32$.

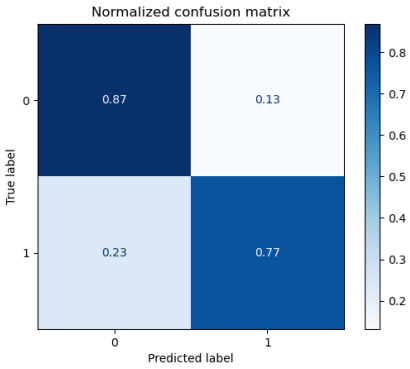
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

of 15. The DQN was trained with γ of 0.9, ϵ of 1, ϵ_{min} of 0.01, and ϵ_{decay} of 0.995. The model was build with MSE as loss function, mean absolute error as metric, Adam optimizer with learning rate of 0.001, and batch size of 32.

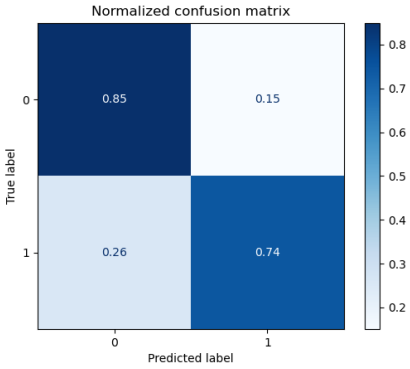
The ResNet achieved an overall accuracy of 83.20% in correctly classifying matrices from the test dataset. The CNN achieved an overall accuracy of 79.65%, and the ShuffleNetV2 achieved 81.71%. Figure 4.8 shows the confusion matrices for the three proposed classification models. The ResNet achieved the highest overall accuracy, followed by ShuffleNetV2 and CNN, respectively. We also observed that all three models performed better in recognizing matrices without the presence of PU. As shown in the confusion matrix in Figure 4.8(a), the ResNet achieved an accuracy of 78% in recognizing matrices that contain PU and 89% in recognizing



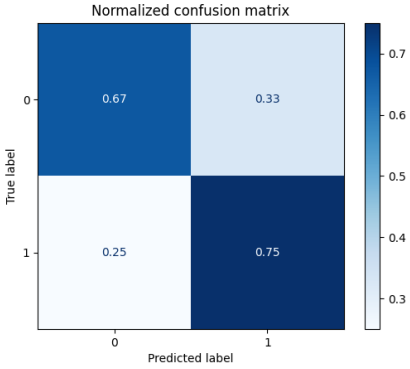
(a) Confusion matrix of the ResNet [1].



(b) Confusion matrix of the ShuffleNetV2 [101].



(c) Confusion matrix of the CNN [63].



(d) Confusion matrix of the DQN [102].

Figure 4.8: Confusion matrices for the classification models proposed.

matrices that do not contain PU. This was followed by ShuffleNetV2, Figure 4.8(b), which achieved 77% in recognizing matrices that contain PU and 87% in recognizing matrices that do not contain PU. The CNN, Figure 4.8(c), achieved 74% accuracy in recognizing matrices that contain PU and 85% in recognizing matrices that do not contain PU.

Differently from the classification models, the DQN was trained with the MSE loss function and with two actions: target cooperative matrices with the presence of a PU signal or target matrices without the presence of a PU signal. The model tried to approximate the categorical classes, and the reward was 1 if it hit the target and -1 otherwise. Using this method, the DQN achieved an overall accuracy of 71.21% in correctly classifying matrices from the test dataset. In Figure 4.8(d), the confusion matrix of the DQN proposed by [102] is shown. Notice that the DQN is better at recognizing class 1 than class 0, with over 74% accuracy in classifying cooperative matrices with the presence of a PU signal. In Table 4.4, the comparison

between the models is shown.

Table 4.4: Accuracy comparison between proposed deep cooperative spectrum sensing models.

Model	Accuracy
ResNet [1]	83.20%
CNN [63]	79.65%
ShuffleNetV2 [101]	81.71%
DQN [102]	71.21%

4.4.2 Spoofing deep cooperative spectrum sensing

In this stage, the GAN is trained and used to create a fake dataset of PU signals. To train the GAN, $42,000$ PU signals were generated based on Equation (4.1). Half of these signals are PU signals, and the other half represents the power leakage of PU signals. The proposed GAN was trained for 200 epochs with a noise dimension of $1,000$, using the Adam optimizer with a learning rate of 0.0001 for both the generator and discriminator. The loss function was binary cross-entropy, and the batch size was 256. The generator has $6,180,224$ trainable parameters, and the discriminator has $3,694,049$ trainable parameters.

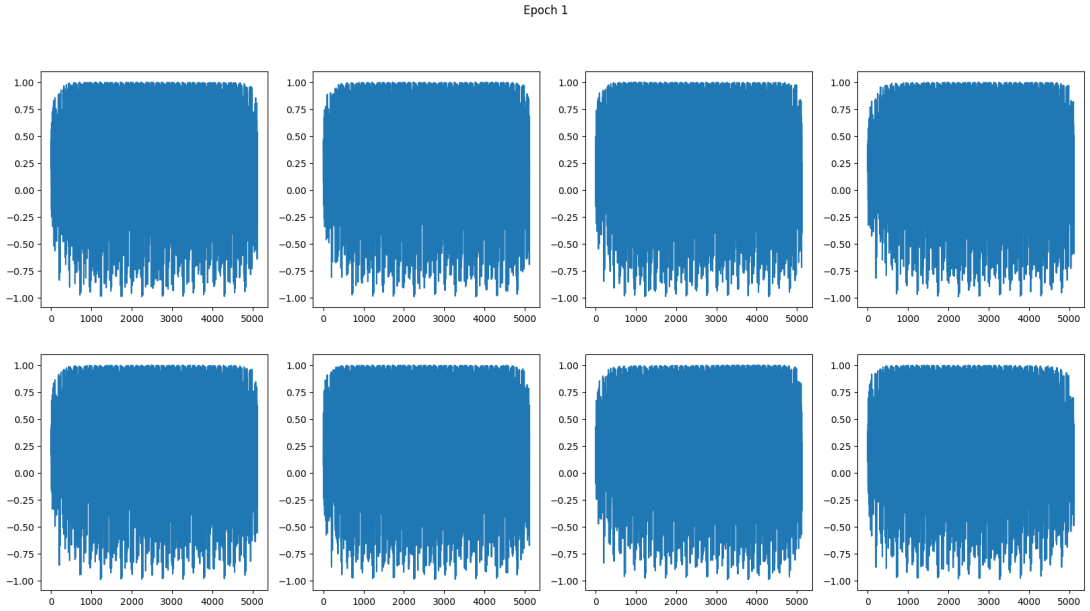


Figure 4.9: Example of signals created by the generator of the proposed trained GAN at the 1th epoch.

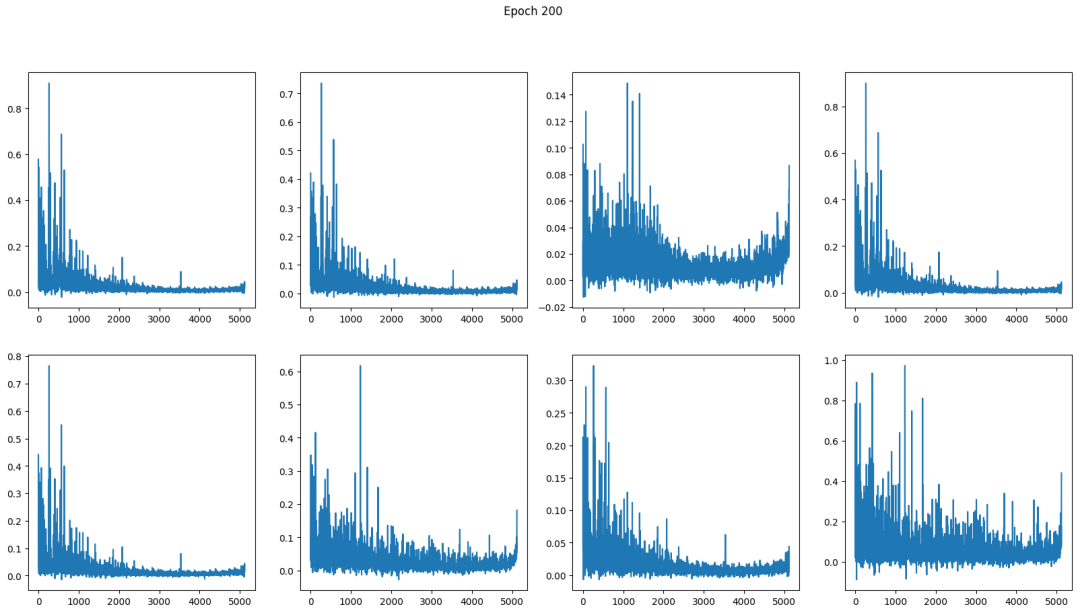


Figure 4.10: Example of signals created by the generator of the proposed trained GAN at the 200th epoch.

In Figure 4.9, examples of signals created by the generator in the first epoch are shown. It is possible to compare these signals with those created in the 200th epoch, as shown in Figure 4.10. It is notable that as the epochs progress, the GAN becomes capable of creating similar PU signals, as comparing with real signal shown in Figure 4.6.

In the training of GANs, it's expected that the loss curves of the generator and discriminator converge and be as low as possible. In Figure 4.11, the losses of the generator and discriminator are shown over the epochs. Notice that the curves try to converge over the epochs. Even so, the discriminator in the experiments at the 200th epoch presented a lower loss of 0.699 compared to the generator's loss of 1.3914. This means that the discriminator does a better job of identifying real and fake signals than the generator does in trying to deceive the discriminator.

The next step is to create fake signals from the noise using the generator. A total of $10,000$ noise samples, each of size $1,000$, are randomly created to use as input for the generator, which outputs $10,000$ fake PU signals. These signals are then used as input for the individual spectrum sensing models. These models attempt to classify whether the fake input signal is a PU signal or not. An evaluation of these models is conducted, and the results are used to create the cooperative matrices. The performance of these models directly impacts these matrices. If the models

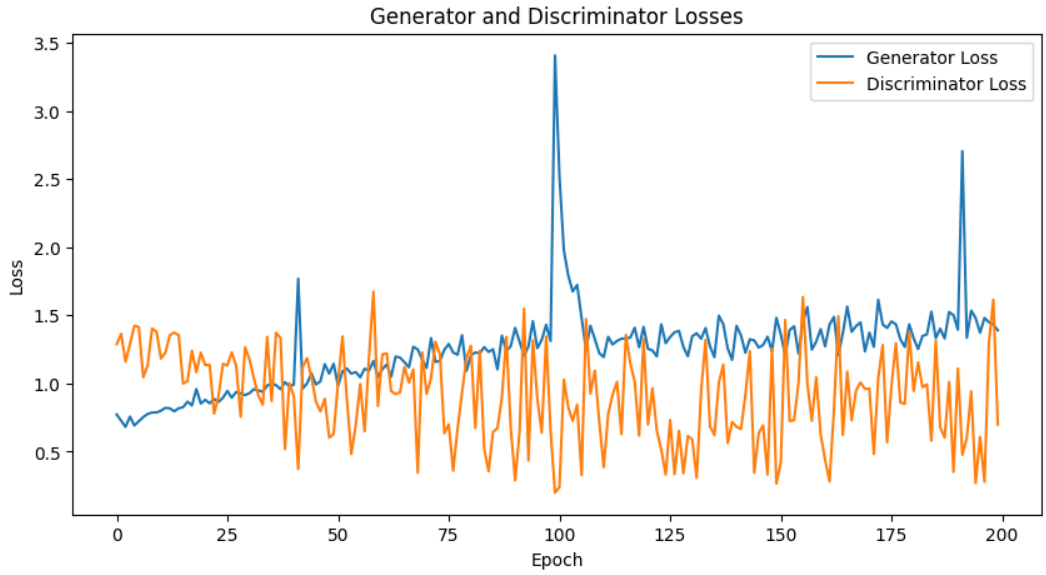


Figure 4.11: Graphic of loss of the discriminator and generator over the epochs.

have difficulty distinguishing real signals from fake ones, it will directly affect the content of these matrices and, consequently, the performance of the deep cooperative spectrum sensing models. In Table 4.5, the evaluation of how much these individual spectrum sensing models are deceived is shown. Out of $10,000$ fake PU signals created, the LGBM was deceived in 94.64% of the cases, XGB in 95.62%, Random Forest in 92.32%, Bagging in 62.86%, and Extra Trees was deceived in 99.12%.

Table 4.5: Deceive rate of each individual spectrum sensing model.

Model	Deceive rate
LGBM	94.64%
XGB	95.62%
Random Forest	92.32%
Bagging	62.86%
Extra Trees	99.12%

The next step is to create the cooperative matrices. Following Algorithm 2, $1,000$ fake cooperative matrices for each individual spectrum sensing model are created. These matrices are formed by the output of these models. For example, the Extra Trees model classified almost all fake signals as real PU signals, which will impact the cooperative matrix. On the other hand, the Bagging model showed some resistance, and only 62.86% of the fake PU signals were recognized as real PU signals.

We created $5,000$ fake cooperation matrices, with $1,000$ matrices gener-

ated using each of the individual spectrum sensing models. Four deep cooperative spectrum sensing models were proposed for spoofing: ResNet [1], CNN [63], ShuffleNetV2 [101], and DQN [102]. Table 4.6 shows the evaluation of how much these models were deceived. All models were deceived in more than 98% of the cases, with ResNet being the most deceived model at 99.72%. The DQN was the model that was deceived the least, classifying 98.32% of the fake cooperative matrices as real matrices with PU.

Table 4.6: Deceive rate of each deep cooperative spectrum sensing model.

Model	Deceive rate
ResNet [1]	99.72%
CNN [63]	99.56%
ShuffleNetV2 [101]	99.52%
DQN [102]	98.36%

In Table 4.7, the deceive rate of the deep cooperative spectrum sensing models with cooperative matrices created using each individual spectrum sensing model is shown. The matrices created by the Bagging model were the only ones that did not achieve a 100% deceive rate by the deep models. This is likely because the Bagging model showed some resistance to the fake signals, as shown in the results in Table 4.5. The other models, except for the DQN model, achieved a deceive rate of more than 99%.

Table 4.7: Deceive rate of each deep cooperative spectrum sensing model with cooperative matrices created by each individual spectrum sensing model.

Model	LGBM	XGB	Random Forest	Bagging	Extra Trees
ResNet [1]	100%	100%	100%	98.6%	100%
CNN [63]	99.92%	100%	100%	97.9%	100%
ShuffleNetV2 [101]	100%	100%	100%	97.6%	100%
DQN [102]	99.1%	99.7%	97.4%	97.1%	98.5%

These results demonstrate the vulnerability of cooperative spectrum sensing networks to MU. Using a GAN, it was possible to mimic PU signals, deceive individual spectrum sensing models, negatively influence the generation of cooperative matrices, and deceive deep cooperative spectrum sensing models responsible for the decision-making of several channels. This could severely damage the dynamics of cognitive radio and, consequently, reduce the efficiency in the use of the frequency

spectrum.

4.5 Discussions

Analyzing the experiments and results achieved in this chapter, we can summarize the discussions in five points: (1) real data generation, (2) individual spectrum sensing, (3) deep cooperative spectrum sensing, (4) training of the GAN, and (5) spoofing deep cooperative spectrum sensing. In point (1), some analysis can be performed by comparing the work in [1]. The data generated for training the individual spectrum sensing and deep cooperative spectrum sensing models show one difference. In this chapter, the Δt was 5 seconds, compared to 2 seconds in the referenced work. This single difference can impact the future models proposed, as the number of samples increases from $2,048$ to $5,120$.

The first impact of this change in signal generation is evident in individual spectrum sensing, as discussed in point (2). In [1], a feature extraction method combined with a Random Forest classifier was used as the individual spectrum sensing approach. In contrast, we used Lazy Predict directly with the real generated signals as input. In terms of performance, we achieved 94% overall accuracy with the LGBM model in our experiments, compared to over 95% overall accuracy in the referenced work. This difference can be attributed to the use of the feature extractor and the reduced number of samples of the generated signals. The five models that best performed (LGBM, XGB, Random Forest, Bagging, and Extra Trees) will suffer attempt of attack of fake PU signals generated by GAN, in Table 4.1 the overall accuracy of these models is shown.

In point (3), the cooperative matrices used in this chapter have a fixed size, with 50 SUs and 32 bands evaluated by each SU. In [1] and [63], the N_{SU} and N_B are reduced, which can certainly impact the final performance of these models. Additionally, [101] and [102] achieved better results in their work, attaining a higher level of accuracy than what was achieved in this chapter. The simplified ResNet proposed in [1] achieved the highest level of accuracy in comparison to the other models. These models (ResNet [1], CNN [63], ShuffleNetV2 [101], and DQN [102]) will suffer the attempt of attack of cooperative matrices created by fake signals

generated by GAN, in Table 4.4 they overall accuracy's is shown.

Regarding point (4), the generator and discriminator losses, shown in Figure 4.11, converged. However, the discriminator presented a lower loss than the generator, indicating that the discriminator was better at distinguishing real from fake signals than the generator was at creating realistic fake signals. Nevertheless, as presented in Table 4.5, the individual spectrum sensing models were deceived at a considerable rate. All models were deceived at a rate of over 92%, except for the Bagging model, which showed some resistance to fake PU signals. Therefore, even though the generator's loss was not as low, it was sufficient to deceive these models.

And finally, point (5) concerns spoofing deep cooperative spectrum sensing. The cooperative matrices created using fake PU signals employed the individual spectrum sensing models. As shown in Table 4.7, the matrices created by the Bagging model did not achieve the highest deception rate. Evaluating by model, the DQN [102] demonstrated more resistance to these fake cooperative matrices. However, we cannot conclude that the DQN is a resistant model to MU, as it presented the lowest overall accuracy in distinguishing matrices with and without the presence of PU. A deeper investigation into these architectures is necessary.

In general, we can say that deep cooperative spectrum sensing network is highly vulnerable to MU, especially when using generative networks such as GANs. The experiments and results showed a deception rate of over 98%. Given this information's, it is urgent to increase discourse and efforts to recognize fake data to avoid compromising the dynamism and efficiency of the frequency spectrum.

4.6 Conclusion

In this chapter, we investigated the vulnerability of deep cooperative spectrum sensing models to GAN-based spoofing attacks. By leveraging the generator of a trained GAN, we produced highly deceptive PU signals that compromised the decision-making models in a simulated cognitive radio environment. These findings contribute to the ongoing research on the security of cognitive radio networks, highlighting the critical need for advanced detection mechanisms to counter such adversarial threats. The idea was to create fake PU signals based on noise using the

generator of a trained GAN, deceive individual spectrum sensing models, create cooperative matrices with these fake outputs, and deceive deep cooperative spectrum sensing models present in the literature. In the experiments, we achieved over a 92% deception rate for individual models, except for the Bagging model, which showed some resistance. Finally, we achieved over a 98% deception rate for deep cooperative spectrum sensing models. These results highlight the urgent need for research into methods that can recognize these fake signals to prevent MU from disturbing the efficiency of the spectrum frequency, especially in today's world where this resource is even more contested.

Future work should focus on developing hybrid detection models that integrate adversarial training with real-time signal authentication techniques. One promising approach could involve multi-view learning, where different neural networks are trained to recognize specific signal features, such as amplitude, phase, and frequency patterns. Their outputs are combined using an ensemble method, such as weighted voting or stacking, to make a final decision. Additionally, exploring the incorporation of temporal coherence analysis and graph-based anomaly detection could further enhance the robustness of the models against sophisticated spoofing attacks. These enhancements would provide a more comprehensive framework for distinguishing between real and fake signals under diverse environmental conditions.

Chapter 5

Conclusions

It was proposed in an initial study the use of a simplified ResNet for cooperative spectrum sensing [4], however, the computational cost was high, compromising potential experimental applications. To address this difficulty, the use of feature extractor in conjunction with the Random Forest classifier was proposed for the creation of cooperation matrices to be used by a simplified ResNet model [1]. As presented, the achieved results demonstrate that the proposed method is capable of differentiating noise signals from signals with presence of primary user with high levels of accuracy with few secondary users cooperating in the system. It is also noteworthy that, even at high noise levels, the proposed method manages to achieve good levels of accuracy. In addition to accuracy, it is also observed that the method presents low latency, which is desirable in current communication systems, such as 5G and 6G.

Due to the influence of noise and distance perceived in the first research, a second study shows promising results. To better design communication systems and AI models used to increase the efficiency of the spectrum, predicting noise levels and the distances between users in a cooperative network is highly desirable. In this second study, the use of regression models was proposed, ranging from lightweight models such as LGBM to robust networks such as transformers, to predict noise levels and distances between users in a cooperative communication network. The proposed method achieved excellent results, proving to be promising in predicting noise and the distances between PUs and SUs in a system. This definitely helps engineers design better communication systems, provide services, and assist in the

better fitting of AI models for each specification and client.

During these studies on spectrum sensing, some questions about security have arisen. With the recent development of generative networks such as GANs, which can create and modify almost all types of data, concerns about spectrum security have begun to be discussed. A third study shows promising results: using GANs, it is possible for an MU to mimic PU signals and deceive AI models used in deep cooperative spectrum sensing. The experiments conducted have shown a high level of vulnerability, where fake signals are easily mistaken for real PU signals by these models, leading to efficiency reduction and a lack of dynamism in the use of spectrum frequency. Beyond that, these results also raise ideas on how to recognize these MUs before they compromise the performance of communication systems.

In this thesis, three major studies were conducted on spectrum sensing, spectrum estimation, and spectrum security. The objectives described in the introduction were achieved, and the experiments in all three studies contributed to the field of communication. As future work for spectrum sensing, we propose deeper studies on feature extraction and the use of more recent models for both individual spectrum sensing and deep cooperative spectrum sensing. For spectrum estimation, we suggest using real signals instead of synthetic data to increase the reliability of the proposed method. Finally, for spectrum security, we propose developing AI models to identify fake PU signals, thereby dynamically mitigating damage to communication systems.

Bibliography

[1] VALADÃO, M. D., AMOEDO, D., COSTA, A., et al., "Deep Cooperative Spectrum Sensing Based on Residual Neural Network Using Feature Extraction and Random Forest Classifier", *Sensors*, v. 21, n. 21, pp. 7146, 2021.

[2] KIM, C., PARK, D., LEE, H.-N., "Compressive sensing spectroscopy using a residual convolutional neural network", *Sensors*, v. 20, n. 3, pp. 594, 2020.

[3] MISHRA, A., DEHALWAR, V., JOBANPUTRA, J. H., et al., "Spectrum hole detection for cognitive radio through energy detection using random forest". In: *2020 International Conference for Emerging Technology (INCET)*, pp. 1–7, 2020.

[4] VALADÃO, M. D., AMOEDO, D. A., PEREIRA, A. M., et al., "Cooperative Spectrum Sensing System using Residual Convolutional Neural Network". In: *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5, 2022.

[5] VALADÄO, M. D., CARVALHO, C. B., JÚNIOR, W. S., "Trends and challenges for the spectrum sensing in the next generation of communication systems". In: *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, pp. 1–2, 2020.

[6] VALADÃO, M. D., JÚNIOR, W. S., CARVALHO, C. B., "Trends and Challenges for the Spectrum Efficiency in NOMA and MIMO based Cognitive Radio in 5G Networks". In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, 2021.

[7] ARJOUNE, Y., KAABOUCH, N., "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions", *Sensors*, v. 19, n. 1, pp. 126, 2019.

[8] UL HASSAN, M., REHMANI, M. H., REHAN, M., et al., "Differential privacy in cognitive radio networks: A comprehensive survey", *Cognitive Computation*, pp. 1–36, 2022.

[9] CHOWDHURY, S., "Resource allocation in cognitive radio networks using stackelberg game: A survey", *Wireless Personal Communications*, v. 122, n. 1, pp. 807–824, 2022.

[10] JAIN, P. P., PAWAR, P. R., PATIL, P., et al., "Narrowband spectrum sensing in cognitive radio: Detection methodologies", *International Journal of Computer Sciences and Engineering*, v. 7, n. 11, pp. 105–113, 2019.

[11] SUN, H., NALLANATHAN, A., WANG, C.-X., et al., "Wideband spectrum sensing for cognitive radio networks: a survey", *IEEE Wireless Communications*, v. 20, n. 2, pp. 74–81, 2013.

[12] BOULOGEORGOS, A.-A. A., CHATZIDIAMANTIS, N. D., KARAGIANNIDIS, G. K., "Energy detection spectrum sensing under RF imperfections", *IEEE Transactions on Communications*, v. 64, n. 7, pp. 2754–2766, 2016.

[13] ATAPATTU, S., TELLAMBURA, C., JIANG, H., *Energy detection for spectrum sensing in cognitive radio*. Springer, 2014.

[14] SOBRON, I., DINIZ, P. S., MARTINS, W. A., et al., "Energy detection technique for adaptive spectrum sensing", *IEEE Transactions on Communications*, v. 63, n. 3, pp. 617–627, 2015.

[15] DANNANA, S., CHAPA, B. P., RAO, G. S., "Spectrum sensing using matched filter detection", In: *Intelligent Engineering Informatics*, pp. 497–503, Springer, 2018.

[16] BHARGAVI, D., MURTHY, C. R., "Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing". In: *2010*

IEEE 11th international workshop on signal processing advances in wireless communications (SPAWC), pp. 1–5, 2010.

[17] ZHANG, X., CHAI, R., GAO, F., "Matched filter based spectrum sensing and power level detection for cognitive radio network". In: *2014 IEEE global conference on signal and information processing (GlobalSIP)*, pp. 1267–1270, 2014.

[18] YAWADA, P. S., WEI, A. J., "Cyclostationary Detection Based on Non-cooperative spectrum sensing in cognitive radio network". In: *2016 IEEE international conference on cyber technology in automation, control, and intelligent systems (CYBER)*, pp. 184–187, 2016.

[19] KUMAR, A., NANDHAKUMAR, P., "OFDM system with cyclostationary feature detection spectrum sensing", *ICT Express*, v. 5, n. 1, pp. 21–25, 2019.

[20] VERMA, P. K., TALUJA, S., DUA, R. L., "Performance analysis of Energy detection, Matched filter detection & Cyclostationary feature detection Spectrum Sensing Techniques", *International Journal Of Computational Engineering Research*, v. 2, n. 5, pp. 1296–1301, 2012.

[21] SHAH, H. A., KOO, I., "Reliable machine learning based spectrum sensing in cognitive radio networks", *Wireless Communications and Mobile Computing*, v. 2018, 2018.

[22] XIE, J., FANG, J., LIU, C., et al., "Deep learning-based spectrum sensing in cognitive radio: A CNN-LSTM approach", *IEEE Communications Letters*, v. 24, n. 10, pp. 2196–2200, 2020.

[23] PAUL, A., CHOI, K., "Joint spectrum sensing and D2D communications in Cognitive Radio Networks using clustering and deep learning strategies under SSDF attacks", *Ad Hoc Networks*, v. 143, pp. 103116, 2023.

[24] YELALWAR, R., RAVINDER, Y., "Artificial neural network based approach for spectrum sensing in cognitive radio". In: *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1–5, 2018.

[25] QUAN, Z., CUI, S., SAYED, A. H., et al., "Wideband spectrum sensing in cognitive radio networks". In: *2008 IEEE international conference on communications*, pp. 901–906, 2008.

[26] ASWATHY, G., GOPAKUMAR, K., "Sub-Nyquist wideband spectrum sensing techniques for cognitive radio: A review and proposed techniques", *AEU-International Journal of Electronics and Communications*, v. 104, pp. 44–57, 2019.

[27] WANG, H., FANG, J., DUAN, H., et al., "Compressive wideband spectrum sensing and signal recovery with unknown multipath channels", *IEEE Transactions on Wireless Communications*, 2022.

[28] HAMDAOUI, B., KHALFI, B., GUIZANI, M., "Compressed wideband spectrum sensing: Concept, challenges, and enablers", *IEEE Communications Magazine*, v. 56, n. 4, pp. 136–141, 2018.

[29] MARQUES, E. C., MACIEL, N., NAVINER, L., et al., "A review of sparse recovery algorithms", *IEEE access*, v. 7, pp. 1300–1322, 2018.

[30] ARJOUNE, Y., KAABOUCH, N., EL GHAZI, H., et al., "A performance comparison of measurement matrices in compressive sensing", *International Journal of Communication Systems*, v. 31, n. 10, pp. e3576, 2018.

[31] LI, Z., WU, W., LIU, X., et al., "Improved cooperative spectrum sensing model based on machine learning for cognitive radio networks", *IET Communications*, v. 12, n. 19, pp. 2485–2492, 2018.

[32] TAN, Y., JING, X., "Cooperative Spectrum Sensing Based on Convolutional Neural Networks", *Applied Sciences*, v. 11, n. 10, pp. 4440, 2021.

[33] LI, Z., LIU, F., YANG, W., et al., "A survey of convolutional neural networks: analysis, applications, and prospects", *IEEE transactions on neural networks and learning systems*, 2021.

[34] GHIMIRE, D., KIL, D., KIM, S.-H., "A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration", *Electronics*, v. 11, n. 6, pp. 945, 2022.

[35] VALADÃO, M., SILVA, L., SERRÃO, M., et al., "MobileNetV3-based Automatic Modulation Recognition for Low-Latency Spectrum Sensing". In: *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–5, 2023.

[36] WANG, C., WANG, B., LIU, H., et al., "Anomaly detection for industrial control system based on autoencoder neural network", *Wireless Communications and Mobile Computing*, v. 2020, 2020.

[37] GUI, J., SUN, Z., WEN, Y., et al., "A review on generative adversarial networks: Algorithms, theory, and applications", *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[38] SHAWEL, B. S., WOLEDEGEBRE, D. H., POLLIN, S., "Deep-learning based cooperative spectrum prediction for cognitive networks". In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 133–137, 2018.

[39] NASSER, A., CHAITOU, M., MANSOUR, A., et al., "A deep neural network model for hybrid spectrum sensing in cognitive radio", *Wireless Personal Communications*, v. 118, n. 1, pp. 281–299, 2021.

[40] SHACHI, P., SUDHINDRA, K., SUMA, M., "Convolutional neural network for cooperative spectrum sensing with spatio-temporal dataset". In: *2020 International conference on artificial intelligence and signal processing (AISP)*, pp. 1–5, 2020.

[41] DAVASLIOGLU, K., SAGDUYU, Y. E., "Generative adversarial learning for spectrum sensing". In: *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018.

[42] GAN, Y., JIANG, C., BEAULIEU, N. C., et al., "Secure collaborative spectrum sensing: A peer-prediction method", *IEEE Transactions on Communications*, v. 64, n. 10, pp. 4283–4294, 2016.

[43] PATEL, M., WANG, X., MAO, S., "Data augmentation with conditional GAN for automatic modulation classification". In: *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, pp. 31–36, 2020.

[44] ROY, D., MUKHERJEE, T., RIDEN, A., et al., "GANSAT: A GAN and SATellite Constellation Fingerprint-Based Framework for GPS Spoof-Detection and Location Estimation in GPS Deprived Environment", *IEEE Access*, v. 10, pp. 45485–45507, 2022.

[45] AYANOGLU, E., DAVASLIOGLU, K., SAGDUYU, Y. E., "Machine Learning in NextG Networks via Generative Adversarial Networks", *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[46] SAGDUYU, Y. E., ERPEK, T., SHI, Y., "Adversarial machine learning for 5G communications security", *Game Theory and Machine Learning for Cyber Security*, pp. 270–288, 2021.

[47] NAVIDAN, H., MOSHIRI, P. F., NABATI, M., et al., "Generative Adversarial Networks (GANs) in networking: A comprehensive survey & evaluation", *Computer Networks*, v. 194, pp. 108149, 2021.

[48] VALADÃO, M. D., AMOEDO, D. A., TAVARES, S. A., et al., "Rede Adversária Generativa Semi-Supervisionada para Falsificação de Sinais Modulados Utilizados em Simulação de Ataque a Modelos de Reconhecimento Automático de Modulações". In: *XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)*, pp. 1–5, 2022.

[49] LIU, H., ZHU, X., FUJII, T., "Ensemble deep learning based cooperative spectrum sensing with semi-soft stacking fusion center". In: *2019 IEEE wireless communications and networking conference (WCNC)*, pp. 1–6, 2019.

[50] ZHANG, Q., NICOLSON, A., WANG, M., et al., "DeepMMSE: A deep learning approach to MMSE-based noise power spectral density estimation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 28, pp. 1404–1415, 2020.

[51] WU, Y., LI, X., CAO, Z., "Effective doa estimation under low signal-to-noise ratio based on multi-source information meta fusion", *Journal of Beijing Institute of Technology*, v. 30, n. 4, pp. 377–396, 2021.

[52] PANDIAN, A. P., "Novel distance estimation based localization scheme for wireless sensor networks using modified swarm intelligence algorithm", *IRO Journal on Sustainable Wireless Systems*, v. 2, n. 4, pp. 171–176, 2021.

[53] LUO, X., QIN, Q., GONG, X., et al., "A Survey of Adversarial Attacks on Wireless Communications". In: *International Conference on Edge Computing and IoT*, pp. 83–91, 2022.

[54] SHI, Y., DAVASLIOGLU, K., SAGDUYU, Y. E., "Generative adversarial network for wireless signal spoofing". In: *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, pp. 55–60, 2019.

[55] SHI, Y., DAVASLIOGLU, K., SAGDUYU, Y. E., "Generative adversarial network in the air: Deep adversarial learning for wireless signal spoofing", *IEEE Transactions on Cognitive Communications and Networking*, v. 7, n. 1, pp. 294–303, 2020.

[56] GATTOUA, C., CHAKKOR, O., AYTOUNA, F., "An overview of cooperative spectrum sensing based on machine learning techniques". In: *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, pp. 1–8, 2020.

[57] VALADÃO, M. D., PEREIRA, A. M., AMOEDO, D. A., et al., "Classificação Automática de Modulações utilizando Redes Neurais Artificiais com Regularização Bayesiana e Algoritmo de Retropropagação de Levenberg-Marquardt". In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)*, pp. 1–5, 2020.

[58] FURTADO, R. S., TORRES, Y. P., SILVA, M. O., et al., "Automatic Modulation Classification in Real Tx/Rx Environment using Machine Learning and SDR". In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, 2021.

[59] SOLANKI, S., DEHALWAR, V., CHOUDHARY, J., "Deep learning for spectrum sensing in cognitive radio", *Symmetry*, v. 13, n. 1, pp. 147, 2021.

[60] GHASEMI, A., SOUSA, E. S., "Spectrum sensing in cognitive radio networks: the cooperation-processing tradeoff", *Wireless Communications and Mobile Computing*, v. 7, n. 9, pp. 1049–1060, 2007.

[61] ZHANG, W., MALLIK, R. K., LETAIEF, K. B., "Cooperative spectrum sensing optimization in cognitive radio networks". In: *2008 IEEE International Conference on Communications*, pp. 3411–3415, 2008.

[62] COURONNÉ, R., PROBST, P., BOULESTEIX, A.-L., "Random forest versus logistic regression: a large-scale benchmark experiment", *BMC bioinformatics*, v. 19, n. 1, pp. 1–14, 2018.

[63] LEE, W., KIM, M., CHO, D.-H., "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks", *IEEE Transactions on Vehicular Technology*, v. 68, n. 3, pp. 3005–3009, 2019.

[64] SHI, Z., GAO, W., ZHANG, S., et al., "Machine learning-enabled cooperative spectrum sensing for non-orthogonal multiple access", *IEEE Transactions on Wireless Communications*, v. 19, n. 9, pp. 5692–5702, 2020.

[65] GUPTA, M. S., KUMAR, K., "Progression on spectrum sensing for cognitive radio networks: A survey, classification, challenges and future research issues", *Journal of Network and Computer Applications*, v. 143, pp. 47–76, 2019.

[66] SHEN, X., SHI, D., PEKSI, S., et al., "A multi-channel wireless active noise control headphone with coherence-based weight determination algorithm", *Journal of Signal Processing Systems*, v. 94, n. 8, pp. 811–819, 2022.

[67] CHEON, B.-W., KIM, N.-H., "AWGN Removal Using Modified Steering Kernel and Image Matching", *Applied Sciences*, v. 12, n. 22, pp. 11588, 2022.

[68] NGO, T., KELLEY, B., RAD, P., "Deep learning based prediction of signal-to-noise ratio (SNR) for LTE and 5G systems". In: *2020 8th International*

Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6, 2020.

[69] NAGAH AMR, M., ELATTAR, H. M., ABD EL AZEEM, M. H., et al., "An enhanced indoor positioning technique based on a novel received signal strength indicator distance prediction and correction model", *Sensors*, v. 21, n. 3, pp. 719, 2021.

[70] SPÜLER, M., SARASOLA-SANZ, A., BIRBAUMER, N., et al., "Comparing metrics to evaluate performance of regression methods for decoding of neural signals". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1083–1086, 2015.

[71] JAIN, P., CHOUDHURY, A., DUTTA, P., et al., "Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes", *Processes*, v. 9, n. 11, pp. 2095, 2021.

[72] WANG, X., ZHAO, Y., POURPANAH, F., "Recent advances in deep learning", *International Journal of Machine Learning and Cybernetics*, v. 11, pp. 747–750, 2020.

[73] SU, X., LI, J., HUA, Z., "Transformer-based regression network for pansharpening remote sensing images", *IEEE Transactions on Geoscience and Remote Sensing*, v. 60, pp. 1–23, 2022.

[74] YAO, G., HU, Z., "SNR estimation method based on SRS and DINet". In: *Proceedings of the 2023 15th International Conference on Computer Modeling and Simulation*, pp. 218–224, 2023.

[75] LI, Y., BIAN, X., LI, M., "Denoising generalization performance of channel estimation in multipath time-varying OFDM systems", *Sensors*, v. 23, n. 6, pp. 3102, 2023.

[76] CHATELIER, B., CORLAY, V., CIOCHINA, C., et al., "Influence of dataset parameters on the performance of direct ue positioning via deep learning".

In: *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pp. 126–131, 2023.

[77] WANG, C., XI, J., XIA, C., et al., "Indoor fingerprint positioning method based on real 5G signals". In: *Proceedings of the 2023 7th International Conference on Machine Learning and Soft Computing*, pp. 205–210, 2023.

[78] CONTI, A., MORSELLI, F., LIU, Z., et al., "Location awareness in beyond 5G networks", *IEEE Communications Magazine*, v. 59, n. 11, pp. 22–27, 2021.

[79] YANG, H., XIE, X., KADOCH, M., "Machine learning techniques and a case study for intelligent wireless networks", *IEEE Network*, v. 34, n. 3, pp. 208–215, 2020.

[80] ARJOUNE, Y., KAABOUCH, N., "On spectrum sensing, a machine learning method for cognitive radio systems". In: *2019 IEEE International Conference on Electro Information Technology (EIT)*, pp. 333–338, 2019.

[81] CHICCO, D., WARRENS, M. J., JURMAN, G., "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation", *Peerj computer science*, v. 7, pp. e623, 2021.

[82] CICHOŃ, K., KLIKS, A., BOGUCKA, H., "Energy-efficient cooperative spectrum sensing: A survey", *IEEE Communications Surveys & Tutorials*, v. 18, n. 3, pp. 1861–1886, 2016.

[83] SEGAL, M. R., "Machine learning benchmarks and random forest regression", 2004.

[84] LUO, H., CHENG, F., YU, H., et al., "SDTR: Soft decision tree regressor for tabular data", *IEEE Access*, v. 9, pp. 55999–56011, 2021.

[85] AHMAD, M. W., REYNOLDS, J., REZGUI, Y., "Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees", *Journal of cleaner production*, v. 203, pp. 810–821, 2018.

[86] GEURTS, P., ERNST, D., WEHENKEL, L., "Extremely randomized trees", *Machine learning*, v. 63, pp. 3–42, 2006.

[87] ZHANG, X., YAN, C., GAO, C., et al., "Predicting missing values in medical data via XGBoost regression", *Journal of healthcare informatics research*, v. 4, pp. 383–394, 2020.

[88] CHEN, T., GUESTRIN, C., "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[89] KE, G., MENG, Q., FINLEY, T., et al., "Lightgbm: A highly efficient gradient boosting decision tree", *Advances in neural information processing systems*, v. 30, 2017.

[90] BOTCHKAREV, A., "Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio", *Available at SSRN 3177507*, 2018.

[91] MIN, E., CHEN, R., BIAN, Y., et al., "Transformer for graphs: An overview from architecture perspective", *arXiv preprint arXiv:2202.08455*, 2022.

[92] COHEN, I., HUANG, Y., CHEN, J., et al., "Pearson correlation coefficient", *Noise reduction in speech processing*, pp. 1–4, 2009.

[93] CRESWELL, A., WHITE, T., DUMOULIN, V., et al., "Generative adversarial networks: An overview", *IEEE signal processing magazine*, v. 35, n. 1, pp. 53–65, 2018.

[94] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., et al., "Generative adversarial networks", *Communications of the ACM*, v. 63, n. 11, pp. 139–144, 2020.

[95] SAPKOTA, R., AHMED, D., KARKEE, M., "Creating Image Datasets in Agricultural Environments using DALL. E: Generative AI-Powered Large Language Model", *Qeios*, 2024.

[96] SAKIRIN, T., KUSUMA, S., "A Survey of Generative Artificial Intelligence Techniques", *Babylonian Journal of Artificial Intelligence*, v. 2023, pp. 10–14, 2023.

[97] PAWELEC, M., "Deepfakes and democracy (theory): How synthetic audio-visual media for disinformation and hate speech threaten core democratic functions", *Digital society*, v. 1, n. 2, pp. 19, 2022.

[98] O'SHEA, T. J., WEST, N., "Radio machine learning dataset generation with gnu radio". In: *Proceedings of the GNU radio conference*, v. 1, n. 1, 2016.

[99] SHI, Y., SAGDUYU, Y. E., ERPEK, T., et al., "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies". In: *2018 IEEE international conference on communications workshops (ICC Workshops)*, pp. 1–6, 2018.

[100] ROY, D., MUKHERJEE, T., CHATTERJEE, M., et al., "Detection of rogue RF transmitters using generative adversarial nets". In: *2019 IEEE wireless communications and networking conference (WCNC)*, pp. 1–7, 2019.

[101] CHEN, Z., XU, Y.-Q., WANG, H., et al., "Federated learning-based cooperative spectrum sensing in cognitive radio", *IEEE Communications Letters*, v. 26, n. 2, pp. 330–334, 2021.

[102] KOTHARI, M., KURUMBANSHI, S., "DQN Based Distributed Cooperative Spectrum Sensing for Multiband Multiuser CRN". In: *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, pp. 633–638, 2024.

[103] FELDMAN, M., "Hilbert transform in vibration analysis", *Mechanical systems and signal processing*, v. 25, n. 3, pp. 735–802, 2011.

[104] RASHED, A. E. E., ELMORSY, A. M., ATWA, A. E. M., "Comparative evaluation of automated machine learning techniques for breast cancer diagnosis", *Biomedical Signal Processing and Control*, v. 86, pp. 105016, 2023.

[105] PANDALA, S. R., "Lazy Predict", `https://github.com/shankarpandala/lazypredict/tree/master`, 2024, Accessed on April 18, 2024.

[106] XU, W., FU, Y.-L., ZHU, D., "ResNet and its application to medical image processing: Research progress and challenges", *Computer Methods and Programs in Biomedicine*, pp. 107660, 2023.

[107] TAVAKOLI, A., PARDO, F., KORMUSHEV, P., "Action branching architectures for deep reinforcement learning". In: *Proceedings of the aaai conference on artificial intelligence*, v. 32, n. 1, 2018.

[108] AGGARWAL, A., MITTAL, M., BATTINENI, G., "Generative adversarial network: An overview of theory and applications", *International Journal of Information Management Data Insights*, v. 1, n. 1, pp. 100004, 2021.

[109] AZZOUZ, E. E., NANDI, A. K., "Automatic identification of digital modulation types", *Signal processing*, v. 47, n. 1, pp. 55–69, 1995.

Appendix A

Feature extraction

In order to reduce system complexity, feature extraction from the signals received by the SU is proposed to highlight their unique properties [1, 4, 57, 58, 109]. The proposed features and their descriptions are:

- Maximum value of the power spectral density (PSD) of the normalized and centered instantaneous amplitude (γ_{max}):

$$\gamma_{max} = \frac{\max |\mathcal{DFT}\{a_{nc}(n)\}|^2}{N_s} \tag{1}$$

  where N_s is the number of samples in each segment, and $a_{nc}(n)$ is the normalized and centered instantaneous amplitude, $a_{nc}(n) = \frac{|\mathcal{H}\{y(n)\}e^{i2\pi f_c n}|}{m_a} - 1$. Here, $\mathcal{H}\{y(n)\}$ represents the Hilbert transform, $y(n)$ is the received signal sampled at $t = \frac{n}{fs}$, and m_a is given by $\frac{1}{N_s}\sum_{n=1}^{N_s} |\mathcal{H}\{y(n)\}\, e^{i2\pi f_c n}|$.

- Standard deviation of the normalized and centered instantaneous amplitude (σ_{aa}):

$$\sigma_{aa} = \sqrt{\frac{1}{N_s}\sum_{n=1}^{N_s}\left(a_{nc}(n) - \overline{a_{nc}(n)}\right)^2} \tag{2}$$

  where $\overline{a_{nc}(n)}$ is the mean of the normalized and centered instantaneous amplitude.

- Standard deviation of the centered nonlinear absolute instantaneous phase (σ_{ap}) is evaluated over a non-weak range of signal segments. The weak segment refers to the amplitude value, a_n, which is susceptible to phase distortions due

to the insertion of Gaussian noise, so the non-weak segment region is defined when $a_n(n) \geq 0.1$ [109]. The σ_{ap} is expressed below:

$$\sigma_{ap} = \sqrt{\frac{1}{C}\left(\sum_{a_n(n)\geq 0.1} \phi_{NL}^2(n)\right) - \left(\frac{1}{C}\sum_{a_n(n)\geq 0.1} |\phi_{NL}(n)|\right)^2} \tag{3}$$

where $a_n(n) = \frac{|\mathcal{H}\{y(n)\}e^{i2\pi f_c n}|}{m_a}$ and C is the total number of samples in the non-weak segment of the signal. The variable ϕ_{NL} is the non-linear phase described by the angle between the real and imaginary components of the Hilbert transform of the received signal $\mathcal{H}\{y(n)\}$. Furthermore, $\phi_{NL}(n)$ is the value of the non-linear component of the instantaneous phase at time instant $t = \frac{n}{f_s}$.

- Standard deviation of the centered direct nonlinear phase (σ_{dp}) [109]:

$$\sigma_{dp} = \sqrt{\frac{1}{C}\left(\sum_{a_n(n)\geq 0.1} \phi_{NL}^2(n)\right) - \left(\frac{1}{C}\sum_{a_n(n)\geq 0.1} \phi_{NL}(n)\right)^2} \tag{4}$$

- Standard deviation of the normalized and centered instantaneous frequency (σ_{af}) is calculated over non-weak ranges of a signal segment [109]. σ_{af} is obtained according to the following expression:

$$\sigma_{af} = \sqrt{\frac{1}{C}\left(\sum_{a_n(n)\geq 0.1} f_N^2(n)\right) - \left(\frac{1}{C}\sum_{a_n(n)\geq 0.1} f_N(n)\right)^2} \tag{5}$$

where $f_N(n) = \frac{f(n)-m_f}{r_s}$, where r_s is the symbol rate of the digital sequence, $m_f = \frac{1}{N_s}\sum_{n=1}^{N_s} f(n)$, and $f(n)$ is the instantaneous frequency given by the relative time derivative $\phi_{NL}(n)$ divided by 2π, $\frac{1}{2\pi}\frac{d\phi_{NT}}{dt}$.

- Standard deviation of the absolute value of the normalized and centered instantaneous frequency (σ_f) [109]:

$$\sigma_f = \sqrt{\frac{1}{C}\left(\sum_{a_n(n)\geq 0.1} f_N^2(n)\right) - \left(\frac{1}{C}\sum_{a_n(n)\geq 0.1} |f_N(n)|\right)^2} \tag{6}$$

- Maximum value of the PSD of the normalized and centered instantaneous frequency (γ_{maxf}) is given by the following equation:

$$\gamma_{maxf} = \frac{\max |\mathcal{DFT}\{f_N(n)\}|^2}{N_s} \tag{7}$$

- Maximum value of the Discrete Cosine Transform (max_{dct}):

$$C_x(k) = \begin{cases} \sum_{n=0}^{N-1} 2\mathcal{H}\left\{y(n)\right\} \cos\left(\frac{\pi}{2N} k(2n+1)\right), & \text{for } 0 \leq k \geq N \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The result of the maximum value of the Discrete Cosine Transform over the complex envelope of the signal, given by the expression $\mathcal{H}\left\{y(n)\right\}$, represents the characteristic."

- Maximum value of the Walsh-Hadamard Transform (σ_{wht}):

$$\mathcal{WTH}_N = \overbrace{\mathcal{DFT}_2 \bigotimes \cdots \bigotimes \mathcal{DFT}_2}^{n \text{ times}} \tag{9}$$

where $\mathcal{DFT}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is the 2-point $\mathcal{DFT}$ matrix and $\bigotimes$ denotes the Kronecker product. The characteristic is obtained by calculating the maximum value of the coefficients of the Walsh-Hadamard transform of the complex envelope of the signal.

- Standard deviation of the Discrete Wavelet Transform (σ_{dwt}):

$$\sigma_{dwt} = \sqrt{\frac{1}{N_s} \sum_{n=1}^{N_s} \left(\mathcal{DWT}\{\mathcal{H}\left\{y(n)\right\}\} - \sum_{n=1}^{N_s} \frac{\mathcal{DWT}\{\mathcal{H}\left\{y(n)\right\}\}}{n} \right)^2} \tag{10}$$

where $\mathcal{DWT}$ is the Discrete Wavelet Transform.