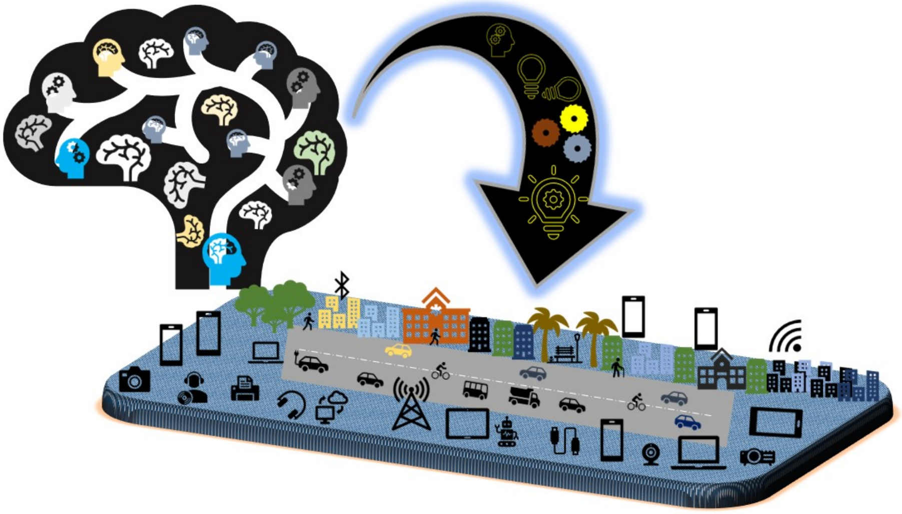
PHD THESIS

Joint Doctoral in Interactive and Cognitive Environments

Data-Driven Approach based on Deep Learning and Probabilistic Models for PHY-Layer Security in AI-enabled Cognitive Radio IoT



Muhammad Farrukh Shahid

June 6, 2021

University Of genova

Queen Mary University of London

Data-Driven Approach based on Deep learning and Probabilistic Models for PHY-Layer security in AI-enabled Cognitive Radio IoT

A thesis submitted for the degree of

Doctor of Philosophy

Scuola Politecnica - Ingegneria

Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni - DITEN

Muhammad Farrukh Shahid

UNIGE Supervisors:
Prof.Carlo Regazzoni
Prof.Lucio Marcenaro

QMUL Supervisor:
Dr.Akram Alomainy

June 2021

TO MY FAMILY AND TEACHERS

Acknowledgments

The research work has been carried out, and a Ph.D thesis has been produced according to the Joint Doctorate in Interactive and Cognitive Environments JD-ICE program between two universities:

University of Genova, Italy

DITEN - Dept. of Electrical, Electronic, Telecommunications Engineering and Naval Architecture

ISIP40 - Information and Signal Processing for Cognitive Telecommunications

Queen Mary University of London UK

EECS - School of Electronic Engineering and Computer Science

CSI - Centre for Intelligent Sensing

Gratefulness

Firstly, I want to thank profoundly Professor Carlo Regazzoni and Professor Lucio Marcenaro for their constant and motivating support throughout the Ph.D period. Without their guidance and suggestions, this work would be next to impossible. I am incredibly grateful to Professor Carlo Regazzoni for enlightening me with in-depth and well-endowed knowledge of the research field. Special thanks to Professor Yue Gao for his guidance and support during the initial period of my Ph.D. I am also highly gratified to Dr.Akram for his motivation and guidance during my Ph.D final year.

I extend my kind words and feelings of thanks to my lab colleagues Mohamad Baydon, who did guide me at the beginning of Ph.D studies, Divya Kanapram, her support was also fruitful and beneficial during research work, Andrea Toma and Ali Kryani, with whom I worked a lot and had healthy research discussion. I am deeply thankful to Hafsa Iqbal and Hassan Zaal.

Special thanks to Dr.Riccardo and Queen Mary official staff members for their support and guidance during the mobility period and for assisting in Visa related issues. A warm thanks and gratitude to UNIGE for providing an opportunity to study and conduct research activities at the Queen Mary University of London, UK.

Finally, my special gratification to Aba, Ammi, Phupho, Dr.Madiha, Dr.Baligh, Ali, Ambrenna, Maryem, Aneesa, Rabia, Adil Mamu's family, and Khala for their prayers and support.

Abstract

Cognitive Radio Internet of Things (CR-IoT) has revolutionized almost every field of life and reshaped the technological world. Several tiny devices are seamlessly connected in a CR-IoT network to perform various tasks in many applications. Nevertheless, CR-IoT suffers from malicious attacks that pulverize communication and perturb network performance. Therefore, recently it is envisaged to introduce higher-level Artificial Intelligence (AI) by incorporating Self-Awareness (SA) capabilities into CR-IoT objects to facilitate CR-IoT networks to establish secure transmission against vicious attacks autonomously. In this context, sub-band information from the Orthogonal Frequency Division Multiplexing (OFDM) modulated transmission in the spectrum has been extracted from the radio device receiver terminal, and a generalized state vector (GS) is formed containing low dimension in-phase and quadrature components. Accordingly, a probabilistic method based on learning a switching Dynamic Bayesian Network (DBN) from OFDM transmission with no abnormalities has been proposed to statistically model signal behaviors inside the CR-IoT spectrum. A Bayesian filter, Markov Jump Particle Filter (MJPF), is implemented to perform state estimation and capture malicious attacks.

Subsequently, GS containing a higher number of subcarriers has been investigated. In this connection, Variational autoencoders (VAE) is used as a deep learning technique to extract features from high dimension radio signals into low dimension latent space z, and DBN is learned based on GS containing latent space data. Afterward, to perform state estimation and capture abnormalities in a spectrum, Adapted-Markov Jump Particle Filter (A-MJPF) is deployed. The proposed method can capture anomaly that appears due to either jammer attacks in transmission or cognitive devices in a network experiencing different transmission sources that have not been observed previously. The performance is assessed using the receiver operating characteristic (ROC) curves and the area under the curve (AUC) metrics.

Author's Publications

Journal papers

1. **"AI-Based Abnormality Detection at the PHY-Layer of Cognitive Radio by Learning Generative Models,"** *IEEE Transactions on Cognitive Communications and Networking (Volume: 6, Issue: 1, March 2020)*

2. **"Jammer Detection in an OFDM Spectrum by Learning Probabilistic Latent Space Models for AI-enabled CR-IoT,"** *IEEE Communications Magazine Series on Internet of Things and Sensor Networks Series* (Submitting-June 2021)

Conference papers

1. **"Jammer detection in M-QAM-OFDM by learning a Dynamic Bayesian Model for the Cognitive Radio,"** *IEEE International 27th European Signal Processing Conference EUSIPCO*, Coruna, Spain, May, 2019.

2. **"Learning a Switching Bayesian Model for Jammer Detection in the Cognitive-Radio-Based Internet of Things,"** *IEEE International Conference on IoT*, Limerick, Ireland, April, 2019.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACS	Autonomic Computing Systems
A-MJPF	Adapted-Markov Jump Particle Filter
ANN	Artificial Neural Network
CR	Cognitive radio
CRN	Cognitive Radio Network
CR-IoT	Cognitive Radio Internet of Things
CNN	Convolutional Neural Network
DBN	Dynamic Bayesian Network
DL	Deep Learning
DSRC	Dedicated Short Range Communication
DoS	Denial of Service
DNN	Deep Neural Network
FDM	Frequency Division Multiplexing
GNG	Growing Nueral Gas
HMM	Hidden Markov Model
IoT	Internet of Things
INTRACC	Intra-Cognitive Communication
ISM	Industrial Scientific and Medical
IoV	Internet of Vehicles
ICC	Information and Communication

IoT	Internet of Thing
IP	Internet Protocol
KF	Kalman Filter
MJPF	Markov Jump Particle Filter
M2M	Machine to Machine
MAC	Medium Access Control
NBI	Narrowband Interference
PSD	Power Spectral Density
PU	Primary User
PUE	Primary User Emulation
SOM	Self-Organizing Map
SS	Spectrum Sensing
LTE	Long Term Evolution
RFID	Radio Frequency Identification Device
TF	Time-Frequency
TVWS	TV White Spaces
USRP	Universal Software Radio Peripheral
OFDM	Orthogonal Frequency Division Multiplexing
UAV	Unmanned Aircraft Systems
UI	Ubiquitous Identifier
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WRAN	Wireless Radio Access Network

Chapter 1

Introduction

The modern advancements in emerging wireless enabling technologies and the profound growth of wireless devices have become the driving force to introduce a new paradigm, known as the Internet of Things (IoT) [1]. Featured with ubiquitous and interactive capabilities to link a massive number of electronic objects, IoT has facilitated a plethora of fields such as healthcare, industry, farming control, environment monitoring, smart cities, and many more [2]. Due to the evolution of successive wireless technologies, Machine-to-Machine communication network (M2M), Vehicular-to-Vehicle (V2V) communication, Unmanned Aerial Vehicle (UAV), and 6G mobile networks, and at the same time, the motivation to deploy IoT functionalities into such networks will construct IoT system more dense and heterogeneous [3],[4].

IoT network exposes to spectrum utilization challenges because of such significant escalation in the number of devices connected in a network. Therefore, an inventive solution is required to address spectrum-related issues. Cognitive Radio (CR) was devised in 1991 by Mitola and Maguire [5] to bestow the concept of intelligent radios capable of learning, reasoning, and acclimating to the environment. An essential feature of CR is the mastery of self-programming and autonomous learning. According to Haykin [6], CR radio meliorates spectrum utilization by using brain-empower devices to achieve efficient exploitation of spectrum and reliable communication objectives. In essence, it integrates model-based reasoning with software radio technologies to build adaptive, smart, and self-configured radios that learn the operating environment parameters and adapt changes accordingly.

Thereupon, it is imperative to acquaint the capability to learn, think, and action into the IoT devices. Hence, unfolding a new paradigm, named Cognitive Radio Internet of Things (CR-IoT) network [7] to deal with spectrum utilization and management challenges. Lately, it has been realized that embodying cognitive functionalities into CR is exclusively not adequate due to the followings reasons [8]: **(1)** The nature of traffic volume in the emerging CR-IoT network is highly dynamic due to the users' demands for several services (e-g voice, audio, text, image, or videos). Such highly vigorous data desire brings a hurdle for CR to learn and predict accurately. **(2)** CR devices will require time and cost (more hardware capacity) to learn the exhaustive and precise information about the radio environment of highly dense CR-IoT and other wireless networks, consisting of several base stations, mobile devices, and other cognitive objects. Therefore, it is necessitous to incorporate more robust and diverse capabilities into CR by introducing artificial intelligence (AI) functionalities. AI teaches Self-Awareness (SA) to the CR by enhancing learning and reasoning functionalities [9].

Consequently, CR-IoT devices equipped with data-driven SA enabled capabilities based on AI techniques autonomously learn models by observing their state and the variation happening in the encircling ambient. The CR-IoT network is vulnerable to numerous kinds of malicious attacks (jammer signals) during transmission. The fundamental objectives of adversary attacks are to destroy communication, deplete the spectrum, and mislead the CR devices. There has been considerable work that presents abnormality detection, more specifically jammer detection in the CR spectrum. In this perspective, numerous techniques, most prestigious include machine learning techniques [10],[11], probabilistic Bayesian networks[12], and deep learning models [13],[14],[15] have been proposed and presented to capture abnormalities in a network spectrum.

Machine learning methods have some limitations; such as distinct features that must be selected and extracted from the data to train the machine learning model [16]. On the contrary, deep learning (DL) methods have gained recent attention due to capturing more sophisticated and dense hidden features and retaining strong abilities in generalizing raw data relationships for diversified IoT applications. Moreover, deep network models exhibit satisfactory performance on a high volume of data. In contrast, machine learning models may encounter an over-fitting issue while handling a considerable amount of information [17]. However, deep learning models can't handle uncertainties and exploit the temporal relationships in the data at different inference levels. In this context, dynamic Bayesian networks allow to exploit temporal relationships in the data and perform inference at different levels (from low to high abstract levels), but they meet certain limitations in the case of high dimensional data. Therefore, a joint framework is required to take advantage of both deep learning models and probabilistic graphical models to achieve abnormality detection tasks in the CR-IoT network spectrum.

1.1 Motivation and Objectives

As introduced in the precedent section, making CR devices more intelligent will build the entire CR-IoT network robust, autonomous, and cognitive to next greater extent. Mainly, SA endowment makes CR understand the normal signal transmission and, if it deviates from the normal operation, ultimately detect such abnormal behavior inside the spectrum. Under such situation, either a control system can then utilize the captured anomalous behaviors as abnormalities in a network to implement anomaly alleviation techniques, or the incorporated SA unit inside the cognitive device can learn new dynamic models that demonstrate diverse scenarios not experienced previously, and prevent attacks.

Implementing a data-driven SA-enabled module based on AI techniques into CR-IoT will build an entire network secure and enable the system to conduct reliable transmission against threats and attacks. The incorporated SA module facilitates the CR-IoT system to detect vicious attacks in the spectrum at the physical layer (PHY-layer) and mitigate their sequels, and it will furnish a road-map to bring a higher level of SA capabilities into the future AI-enabled CR-IoT network.

The over all objectives are as follows:

- Building CR-IoT network more protective and attack-free to conduct secure transmission.
- Making CR devices more intelligent, cognitive, and aware by proposing a method to bring SA capabilities into the CR-IoT objects.
- Providing a probabilistic framework to statistically exploit CR signals and model CR dynamic behavior inside a spectrum evolving with time.
- Capturing and detecting abnormalities at PHY-layer in CR-IoT network.
- Develop a data-driven approach that takes advantage of the deep learning method to handle high dimensional radio signals and dynamic Bayesian model to provide spectrum inference at continuous and discrete levels and perform state estimation tasks. In addition to that, the proposed method should be capable of capturing abnormalities in the CR-IoT network spectrum.

1.2 Thesis Key contributions

The work presents a framework toward the development of SA based AI-enabled CR-IoT network. Incipiently, the entire approach consists of two phases investigating generalized state vectors, which contain Orthogonal Frequency Division Multiplexing (OFDM) modulated sub-band information extracted from the radio spectrum for abnormalities detection. The first phase addresses low dimension data (few sub-carriers). Notably, a jammer detection method is proposed based on learning a switching Dynamic Bayesian Network (DBN) from typical OFDM data transmissions capable of detecting abnormal situations. This work aims to analyze signals behavior through Dynamic Bayesian Network, realize a probabilistic switching model consisting of two hidden levels for each temporal slice, and to detect malicious signals inside the spectrum for the CR-IoT network. The inferences at discrete and continuous levels of the spectrum is achieved using a combination of Particle filter (PF) for the discrete level and Kalman Filter (KF) respectively. After that, Single and Bank-Parallel DBN models have been implemented to capture jammer in an OFDM modulated transmission. Features of all sub-carriers have been learned in the form of a single generalized state vector in the Single DBN approach. On the contrary, individual characteristics of each sub-carrier are learned as a separate generalized state vector associated with each subcarrier in the Bank-Parallel DBN method. Single-DBN is favorable in tracking a CR and keeping the device's profile history whereas, Bank-Parallel is more suitable to distinguish and characterize different sources.

In essence, DBN learns switching models from data series to generalize state vectors where different linear models are described with the switching variables. DBN exhibits good performance when the data dimension is low, and several possible switching dynamic models in DBN is confined. On the other hand, generative models from the deep learning domain can efficiently deal with a significant quantity of dynamic models, but they are impotent to address uncertainties. Therefore, a more robust and powerful method is presented and developed during the second phase of the work, taking advantage of both deep learning and probabilistic network, and giving an abnormalities detection framework. Variational autoencoder (**VAE**) from the DL domain facilitates the achievement of the data dimension reduction step, and the **DBN** from the Bayesian field fulfills the need for state estimation tasks effectively for the CR-IoT spectrum. We deploy VAE to transform high dimension data into low and compact representation. Then latent variables of VAE are clustered to learn temporal dependencies among them and constitute a probabilistic representation. We use Adapted-Markov Jump Particle Filter (A-MJPF) to perform state estimation, which considers the uncertainties in the spectrum and consequently spot any malicious behavior that deviates from the standard etiquette in the spectrum at the continuous level.

As far as we know, for the first time, generalized state vectors are explored and investigated based on latent space information (encoding OFDM modulated data extracted features) obtained from the trained VAE in this work. We deploy two VAEs, one for signal and another for signal derivative.

Following are significant contributions:

- Abnormalities (Jammer attacks) detection at the PHY-layer of CR-IoT network.
- The proposed method can be deployed to less dense networks that generate low-dimension data and a highly dense network containing significantly high dimension data.
- The strength of the proposed methods provokes the motivation to implement such strategies into the CR-IoT devices to develop autonomous devices where they furnish an opportunity to select either of the methods depending on data dimensions.
- The probabilistic models are inevitably learned from complex data under different scenarios by the proposed method that follows a data-driven approach.
- The proposed methods facilitate the development of the SA module in AI-enabled CR-IoT by learning either DBN (low dimension signals) or VAE for obtaining compact latent space representations and learning DBN (for high dimension signals) and eventually deploy any of the method (based on application's dimension) to capture abnormalities in the CR-IoT spectrum.

Besides, the work describes the theoretical background along with relevant work in this field. The thesis also highlights state-of-the art work.

1.3 Thesis structure

Fig.1.1 advertises the flow of work investigated throughout the research phase. Each block shows stages with corresponding related work being carried out.

Chapter 2 unleashes IoT network proliferation in emerging networks such as M2M, V2V, UAV, and 6G mobile networks. It highlights the importance of introducing intel-

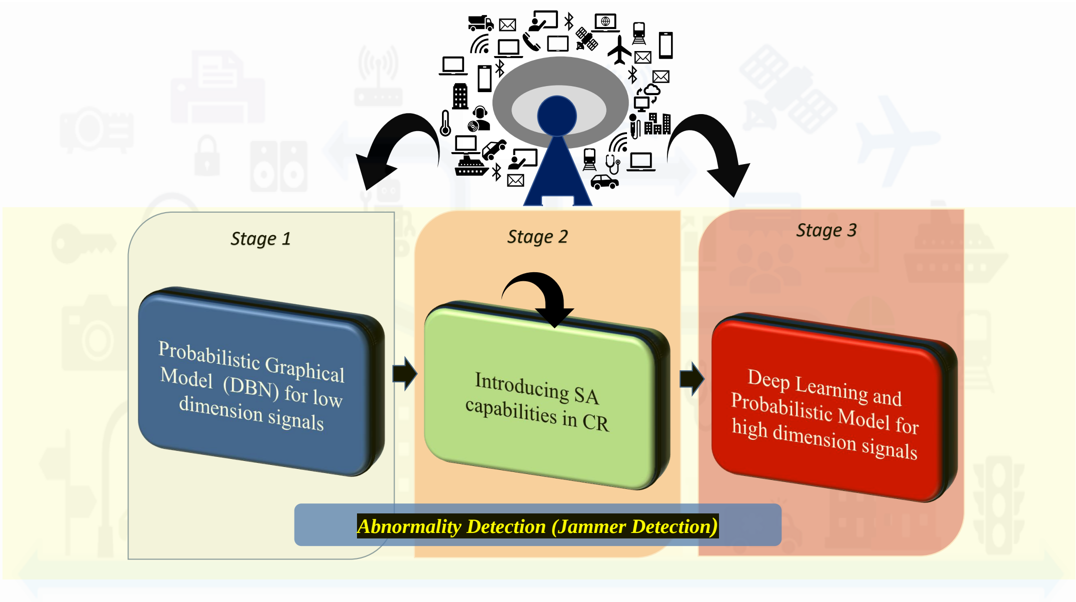

Stage 1
Probabilistic Graphical Model (DBN) for low dimension signals
Stage 2
Introducing SA capabilities in CR
Stage 3
Deep Learning and Probabilistic Model for high dimension signals
Abnormality Detection (Jammer Detection)

Figure 1.1: Workflow investigated in this thesis

ligence into IoT devices, giving a new paradigm CR-IoT. The chapter also provides insights into cognitive radio (CR) and its cognitive cycle. The chapter ends with notable CR-IoT impacts in a plethora of fields.

Chapter 3 It exclusively discusses the security threat in CR and CR-IoT networks. Specifically, related work covering jammer attacks with detecting methods have been presented. It also discusses and highlights jammer detection work present in the literature for the CR-IoT network. Moreover, it describes the developed method's motivation and feasibility of the work investigated in this thesis compared to the current work.

Chapter 4 CR behavior inside the spectrum can be modeled through probabilistic graphical models. Such a model facilitates predicting the current and future state of an object (CR) by using probabilities theories. In this context, this chapter introduces how CR can be modeled as an agent in the environment through dynamic models. This chapter also presents and proposes a DBN model to detect abnormalities in a CR-IoT spectrum. The proposed method deals with low-dimensional data and detect jammer attacks in the spectrum.

Chapter 5 Nowadays, deep learning models have gained a lot of immersion due to promising performance achieved by such models in various fields. In this chapter, we briefly discuss the motivation to bring consciousness into computing systems. Or, in other words, introducing self-awareness into the network. Followed by this, bringing cognitions into CR devices with a higher level of AI have been discussed. We demonstrate the jammer signal classification method based on deep learning models (AlexNet and GoogLeNet). Both models classify abnormal signals of the FFT and CWT based images. Towards the end of the chapter, we discuss popular generative models such as GAN, AE, and VAE. The current state-of-art work is also discussed throughout the chapter wherever it was necessary.

Chapter 6 unveils firstly the abnormalities detection relevant work in the CR-IoT spectrum. Followed by this, high dimension approach combing deep learning and DBN for abnormality detection is described in detail. The chapter discusses layered structure implementation of the VAE model for signal and signal derivative to obtain latent space information. In the derivative VAE, the decoder output is forced to reconstruct the null version of the signal by introducing an activation regularizer in a network. Hence, realizing architecture similar to the null force filter structure. The latent spaces are then clustered to learn temporal relationship among them. We present an analysis of the developed method in terms of Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) metrics. Moreover, a comparison between VAE implementation with and without activation regularizer is described and demonstrated. Finally, this chapter concludes by comparing VAE implementation (with and without activation regularizer) under different scenarios such as varying jammer power and changing latent space vector size.

Chapter 7 highlights the motivation of proposed methods based on data dimensions. It explicitly describes the advantages and limitations of the proposed methods and compares data-driven techniques and model-driven methods. The chapter ends with the future directions and highlights the road-map for the next step in the investigated research work.

References

[1] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, “Iot connectivity technologies and applications: A survey,” *IEEE Access*, vol. 8, pp. 67 646–67 673, 2020.

[2] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, “Internet of things meets brain–computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2084–2092, 2019.

[3] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, “Artificial intelligence enabled internet of things: Network architecture and spectrum access,” *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[4] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, “An in-depth analysis of iot security requirements, challenges, and their countermeasures via software-defined security,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 250–10 276, 2020.

[5] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[6] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. . Neel, Y. Zhao, J. H. Reed, and W. H. Tranter, “A survey of artificial intelligence for cognitive radios,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1578–1592, 2010.

[7] A. A. Khan, M. H. Rehmani, and A. Rachedi, “When cognitive radio meets the internet of things?” in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 469–474.

[8] Z. Qin, X. Zhou, L. Zhang, Y. Gao, Y. Liang, and G. Y. Li, “20 years of evolution from cognitive to intelligent communications,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 6–20, 2020.

[9] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni, “Ai-based abnormality detection at the phy-layer of cognitive radio by learning generative models,” *IEEE Transactions on Cognitive Communications and Net-*

working, vol. 6, no. 1, pp. 21–34, 2020.

[10] F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and k-nearest neighbor," in *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 2018, pp. 68–72.

[11] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33 789–33 795, 2018.

[12] A. Krayani, M. Farrukh, M. Baydoun, L. Marcenaro, Y. Gao, and C. S.Regazzoni, "Jammer detection in m-qam-ofdm by learning a dynamic bayesian model for the cognitive radio," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[13] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[14] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5.

[15] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2020.

[16] V. Porkodi, D. Yuvaraj, J. Khan, S. A. Karuppusamy, P. M. Goel, and M. Sivaram, "A survey on various machine learning models in iot applications," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 2020, pp. 1–4.

[17] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, and J. Liu, "A survey on deep learning empowered iot applications," *IEEE Access*, vol. 7, pp. 181 721–181 732, 2019.

Chapter 2

Cognitive Radio Internet of Things (CR-IoT) and its applications

CR-IoT is an emerging technology connecting tiny objects and furnishes communication for several applications. This chapter introduces the development and penetration of IoT into many fields. Nevertheless, IoT network capabilities remain incomplete and incompetent without cognitive abilities. Therefore, cognitive radio (CR) and its implementation into IoT have been introduced, followed by some notable CR-IoT applications in this chapter.

2.1 IoT-A new paradigm for seamless connectivity

Recent evolution in information and communication technologies invoked an emerging method to connect diverse objects in a network smartly, known as the Internet of Things (IoT) [1]. Such a network contains various devices to furnish ubiquitous connectivity and grant access to end-users for multi-fold applications. In IoT, several things present around us and, such items include tablets, sensors, smartphones, laptops, wireless headphones, smartwatches, and other intelligent devices. IoT objects are empowered with mighty data processing potency and use several communication protocols to avail any service by using ideally any available link [2]. IoT is a rising and emerging network that influences a plethora of domains, such as healthcare, smart cities, home automation, industrial process, intelligent transportation, and many more. From a fundamental point of view, IoT connects everything to the internet in the world. Technically, the "Internet of Things" can be explained as things or objects connected to the internet in time than people [3]. More comprehensively, IoT can be perceived as a network dispensing connectivity to non-internet-enabled devices or objects. IoT can connect heat monitors, air conditioner, remote control, streetlights, cars, motorbikes, cycles, kitchen appliances, and virtually anything to the network, as shown in Fig.2.1. IoT has been evolving and, more smart devices will become part of the IoT network forthwith. More formally, IoT can be defined as [4],

A blanket of connected objects operating in a smart environment using autonomous links to communicate intelligently within environmental, social, and user contexts.

Figure 2.1: IoT-paradigm is connecting a plethora of applications to the single network.

The proliferation of IoT technology in the market is shown in Fig.2.2 and, is summarized as follows [3],

1) The number of connected devices/objects will reach approximately 10×10^9 by 2020, whereas, in 2018, 7×10^9 devices were connected to the internet.

2) According to Gartner, 14×10^9 devices will be connected in 2019, and by 2021, this connection will reach up to 25×10^9.

3) The smartphones are the integral component in the IoT network, and according to Newzoo, 3×10^9 smartphones were connected to the network in 2018.

4) Most of the devices are being used in the IoT network at home, and such devices are connected to Wireless Personal Area Networks (WPAN) such as Bluetooth, Zigbee.

There are myriad enabling technologies that have greased the implementation of IoT network in numerous applications such as clouding computing, Wireless Sensor Network (WSN), Wireless Sensor and Actuator Networks (WSAN), and Machine to Machine (M2M) networks. Primarily, cloud computing provides the user with on-demand services such as ubiquitous access, resource pooling, and service provision [6]. It is foreseen that IoT functionalities will be brought by cloud-based IoT into the cloud servers whereas, the IoT-centric cloud will import cloud functionalities into IoT. WSN has been identified as a potential candidate for IoT networks over the few decades; particularly, IEEE 802.15.4, which supports low power and less bit rate, is widely deployed in WSN networks to connect tiny sensors. M2M communication has recently gained the popularity that provides ubiquitous connections among devices and enables devices to interact without human

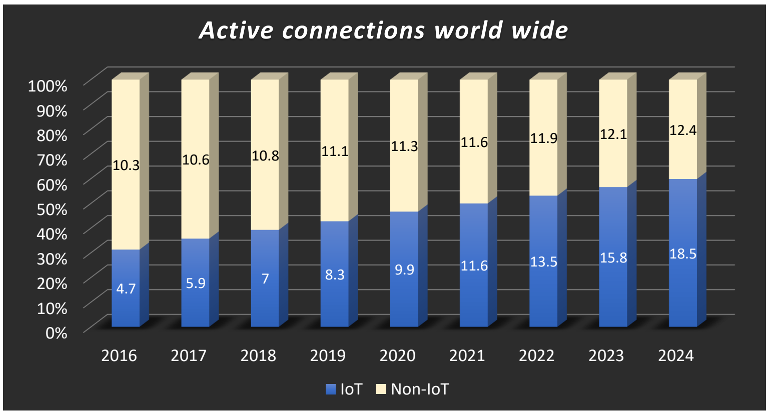


Figure 2.2: Total number of active connections worldwide [5].

interaction [7]. IoT network comprises of several objects which autonomously operate and adapt to changes according to the network requirements. Consequently, M2M technology has been the most relevant and vital option for IoT networks. IoT is supposed to be a single global network; nonetheless, IoT comprises multiple independent and complementary networks in various disciplines such as home automation, transportation, industrial processes, and health care. IoT has penetrated many-particle applications and has become an integral part of many technologies [8]. Such applications include smart cities, environment control, industrial process, home appliances, and healthcare. In smart cities, IoT potential use includes parking slot management, indoor localization, traffic light management, and crowd movement analysis. In contrast, IoT in environment control covers smart farming, air quality monitoring, and much more. Radio-Frequency Identification (RFID) based object tagging is one of the main applications of IoT in the industry. Fitness tracking, health monitoring, and baby monitoring come under the IoT application for healthcare. The most exciting forms of IoT exist in a home for controlling temperature, open/close doors, etc. Fig.2.3 shows IoT growth in certain applications [9].

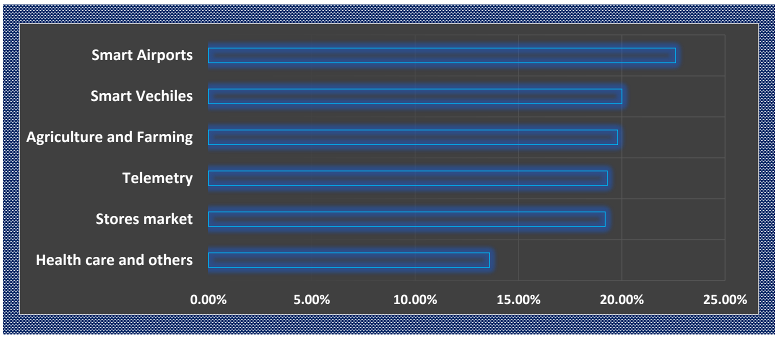


Figure 2.3: IoT growth in certain applications.

2.2 Bringing intelligence into the IoT devices

IoT network comprises various devices to furnish ubiquitous connectivity and grant access for the multi-fold applications to the end-users. IoT-oriented mesh expedites the substantial connectivity of billions of devices. In this regard, connectivity among IoT devices in a network can be hastened through wired or wireless technologies. Wherein the wireless method is a more befitting and feasible solution for the connections among devices due to the heterogeneous nature of the IoT network [10]. However, furnishing wireless connection among devices in IoT is simply not adequate due to the following reasons [11]. **1)** Wireless connections endure from channel congestion sequel as there remains a chance of spectrum under-utilization by the devices. Studies have revealed that the spectrum is not exploited by the legitimate user most of the time and remains unoccupied. In a nutshell, the wireless network meets spectrum management issues. **2)** IoT network is vulnerable to vicious attacks during the transmission that causes disruption to the communication and eventually deteriorates the overall network performance. Thereupon, it is imperative to acquaint the capability to learn, think, and action into the IoT devices. Hence, unfolding a new paradigm, named CR-IoT network. CR-IoT empowers the existing IoT with a *Brain* for high-state intelligence as shown in Fig.2.4. Invoking AI endowments into the CR-IoT system eventually settle the problem of spectrum under-utilization wherein devices equipped with a level of intelligence can efficiently utilize the available channels and manage the spectrum.

Moreover, AI introduces a level of Self-Awareness (SA) into the CR-IoT network, which ensures devices not only aware of their own state but their operating environment as well. Accordingly, when intimidating signals appear in a system, devices will automatically detect such abnormal behavior, combat such attacks, and learn a new model to avoid such an experience. CR-IoT devices generate a large volume of data. AI methods act as a catalyst that extracts useful information from that raw data and executes the intelligent decision based on the information encoded in the data. We can formally defined CR-IoT network as,

CR-IoT comprises objects that are seamlessly connected and interlinked in a network with less or no external intervention. According to the CR cognitive cycle, such devices equipped with cognitive capabilities learn, interact, and perform several tasks.

Hence, CR-IoT networks are self-organized, self-configured, and self-adaptive due to the

AI techniques that yield consciousness into the devices.

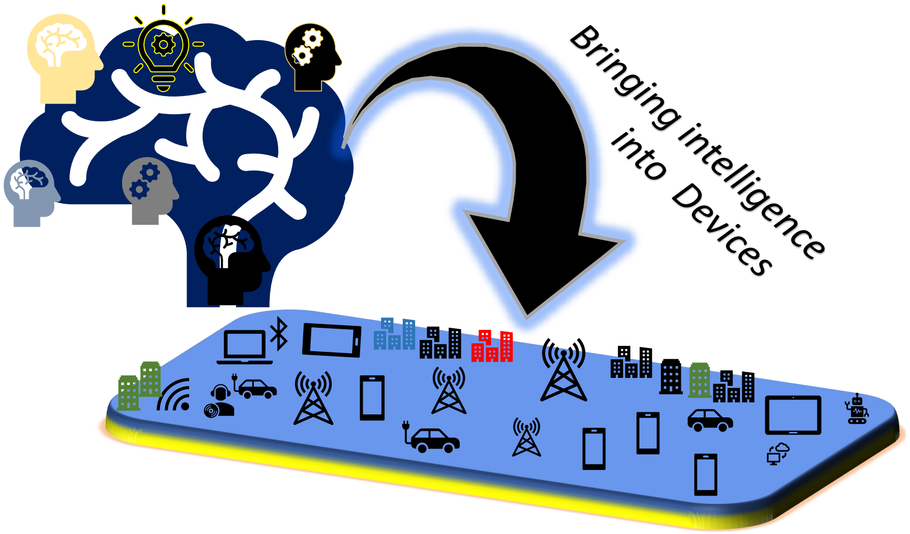


Figure 2.4: Introducing intelligence into IoT devices.

2.3 The foundation of CR-IoT Network

CR-IoT has revamped the digital world by linking a plethora of technologies to bring comfort to life. Nowadays, people can access various applications on their smartphones and avail many services at any time, no matter where they are. Incontestably, cognitive radio has paved the way for robustness, cognition, and level of awareness to the IoT network. The significance of incorporating cognition capabilities into IoT devices is inevitable, as mentioned in the precedent section 2.2. The term Cognitive Radio was devised in 1991 by Mitola and Maguire [12] to sketch the concept of intelligent radios capable of learning, reasoning, and acclimating to the environment. An essential feature of CR is the mastery of self-programming and autonomous learning. According to Haykin [13], CR radio meliorates spectrum utilization by using brain-empower devices to achieve fundamentally two objectives: 1) Efficient exploitation of spectrum. 2) Reliable communication. We can formally define CR as,

Intelligent radio that continuously learns the operating environment, tune parameters (modulation order, power level, coding schemes), and adapts the

best transmission strategy to pursue smooth communication while effectively utilizing the radio spectrum

The preeminent driving force to bring the CR into existence comes from the escalated demand of data rates by the users to avail of different services such as Long-Term Evolution (LTE), 5G network, Wireless Fidelity (Wi-Fi), Bluetooth, Local Area Network (WLAN). Moreover, at the same time, new wireless technologies are being developed. As a result, the radio spectrum has become clogged and overcrowded. However, the spectrum measurements have revealed that a major portion of the spectrum is not exploited most of the time and can't be accessed by other unlicensed users. Spectrum allocated to the licensed user is never exploited by that user at its full extent. Therefore, one way to prevail over such a problem is to build the radio intelligent and awake enough to sense the spectrum proactively and access the vacant band when a licensed user is not conducting transmission in the frequency band inside a spectrum and grant empty channels to the secondary users. CR is one of the emerging wireless communication technologies that intelligently boots the spectral efficiency by allocating vacant spaces (also called spectrum holes) in the spectrum to the secondary user when a legitimate user is not using the spectrum.

The CR network's main actors are legitimate users (primary (PU) or license users), non-legitimate users (secondary users (SU)), and a base station. The legitimate agent accesses the spectrum in the given network at any time and avails services. On the con-

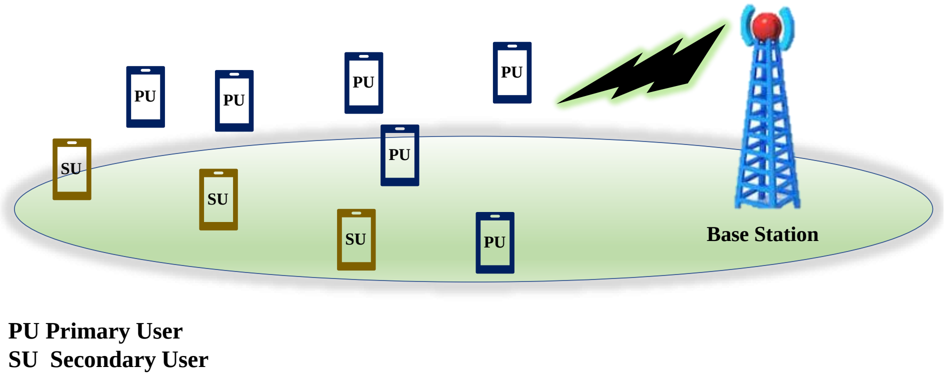


Figure 2.5: CR network consists of licensed user, non licensed user and a tower.

trary, a secondary agent scans the network spectrum and communicates opportunistically whenever there is a vacant channel in the spectrum.

2.3.1 CR cognitive cycle

CR discovers all types of radio frequency (RF) activities in the functioning environment and accomplishes cognitive tasks. Moreover, CR can't conduct cognitive operations without being conscious of its environment. In this perspective, spectrum sensing (SS) is identified as a basic yet crucial process that acts as a bridge between CR and the radio environment. In SS, CR senses the spectrum and gather the information. Recently, SS has been perceived as the perception process in the modern intelligent communication system [14]. Sensing the spectrum is a radio perception ability to gather information about the network agents (PU and SU). In autonomous radios, the perception process specifically targets spotting vacant channels in the spectrum for SU to grant access. According to the crowning trends of AI in CR, we can classify the perception process in the following two categories [14]: **1)** Conventional Spectrum Sensing **2)** Direct learning from RF using deep learning/machine learning methods.

Conventional Spectrum Sensing

Conventional spectrum sensing has two potential classes based on the frequency band.

a) Narrowband SS: Narrowband SS is used when the frequency band is essentially narrow and the channel frequency response is flat. Technically, narrowband SS is preferred when the bandwidth of the signal is smaller than the coherence bandwidth of the channel. Specifically, in the SS process, CR senses the spectrum and detects vacant spaces, which are also called spectrum holes. After detecting holes, characteristics of the detected holes are estimated through spectrum sensing. Spectrum decision is then taken to select the best available band inside the spectrum for the user with respect to its requirements. The most popular narrowband SS methods are 1) Energy detection 2) Matched filtered 3) Feature detection.

b) Widdband SS: In wideband SS, SU scans the wide range of frequency channels sequentially until a vacuous channel is discovered. SU exploits spectral opportunities in (time, frequency, space) over a wideband and detect multiple spectrum holes under one sensing slot. Wideband SS techniques include 1) FFT-based detector 2) Wavelet-based detector 3) Filter band detector 4) Compressive sensing.

Direct learning from RF using deeplearning/machine learning methods
Due to the remarkable developments of machine learning (ML) and deep learning (DL) techniques, it is now realizable to deploy such methods to learn the radio environment. ML technique extracts patterns and learns the features from the huge volume of data. Such features are then employed to learn the ML model, which adapt changes according to the operating environment. Basically, ML methods have been classified into the following categories:1) Supervised ML 2) Unsupervised 3) Reinforcement learning.

Apart from perception, which makes CR wide-awake about ambient, learning, and reasoning are also an epicenter for overall CR operation to realize CR fully cognitive. Learning involves transmuting acquired spectrum information into valuable knowledge by using classification methodologies along with certain hypotheses. Then such knowledge is used to achieve certain objectives through reasoning. The reasoning is a CR potency to exploit the knowledge obtained through learning to attain certain goals [12]. To sum-up overall CR operation: ***Perception*** is obtained through spectrum sensing measurements in which CR becomes aware about RF activities. Followed by perception, CR learns the patterns and observations information and transform such information into some useful knowledge (classes or categories) through ***Learning***. At last, ***Reasoning/Acting*** facilitates CR to achieve specific goals based on knowledge acquired from learning process. Perception, learning, and reasoning constitute a basic cognitive cycle, firstly avowed by Mitola [12].

2.3.2 OFDM as a potential modulation candidate for CR

CR perception operation is heavily influenced by the waveform design, modulation schemes, and propagation models for a particular network. In this regard, OFDM is the most elegant modulation candidate for CR to meet its specific goals due to the adaptive nature of OFDM modulation [15],[16]. OFDM has been implemented into many wireless technologies such as Long Term Evolution (LTE) mobile network, WiFi networks, Wireless Regional Area Network in TV white spaces (TVWS), UAVs, and Vehicular technology [17]. OFDM fulfills the CR needs in a very elegant way where CR requirements of spectrum sensing are provided by inherited FFT operation in OFDM. FFT module in OFDM receiver can be deployed to accomplish spectrum sensing. Hence, there is no need to use another technique to perform spectrum sensing explicitly. CR devices efficiently utilize spectrum by minimizing interference between PU and SU. OFDM has a feature to make subcarriers ON/OFF; thus, the waveform can be shaped to minimize interference between PU and SU in the spectrum. Moreover, CR radios are adaptive, and OFDM is an adaptive modulation technique in which FFT size, number of subcarriers, number of support antennas can be changed or modified according to the network requirement. Such flexibility and adaptiveness have envisaged OFDM modulation as an absolute choice for the PHY-layer modulation technique in the CR network [18]. In addition to this, the effect of propagation models on the operation of CR is diversified with respect to the range of the deployed spectrum, such as cellular band, fixed wireless band, and millimeter-wave. Therefore, the propagation model should be carefully selected for the optimal CR operation [14].

Motivated by the aspiring performance and implementation of OFDM into various notable technologies such as LTE, CR-IoT, UAV, V2V, and mm-wave cognitive radio, we have considered OFDM modulated signal transmission to develop an abnormality detection framework based on SA oriented ability for AI-enabled CR-IoT network for our principle investigation of research work. First of all, we analyze and investigate a few numbers of subcarriers and present a probabilistic framework for jammer detection in the OFDM subcarriers.

After that, we take into consideration large of a number of subcarriers and develop an approach that takes the advantages of generative models and the Bayesian networks jointly to capture jammer attacks inside the CR-IoT network spectrum.

2.4 CR-IoT applications

CR-IoT has been incorporated into a variety of applications [19].

In-Home CR-IoT will be incorporated into home appliances to improve quality of life. Sensors will be deployed inside homes to perform home automation functions and home energy management. Smart fridge, smart meters, and smart lights are an example of intelligent home automation. For these examples, Wi-Fi access points are usually installed, but this can cause severe Industrial Scientific and Medical (ISM) band interference. It is suggested to provide sensors with intelligent capabilities to alleviated deterrent in the ISM band.

Smart Cities In cognitive cities, Information and Communication Technology (ICT), and IoT are integrated to provide development in the cities. In smart cities, e-services to users are provided to enhance their lifestyle. To provide e-services, continuous connectivity is required. Moreover, data gathering and user interaction are also important. Such data acquisition and gathering requirements can be facilitated by deploying cognitive capability in IoT.

Internet of Vehicles (IoV) IoV is a new paradigm in which vehicle control is achieved through communication, power, and embedded systems with less or no human intervention. The availability of spectrum for mobile vehicles is demanding to support and facilitate IoV. Therefore, deploying CR functionalities into IoV will be an excellent solution to provide services on time.

Environmental Application The temperature measurement, waste management, pollution monitoring, weather forecasting has been facilitated by deploying IoT network, and developments are being reported in this field. A heterogeneous network with several miniature devices are required to acquire such functionalities for environmental applications, and for such devices, static spectrum allocation is not viable. Therefore, IoT with cognitive capabilities is a plausible solution for environmental applications.

Health Care In health care, temperature monitoring, fitness monitoring, heart rate status has been around, and people are using such application to keep themselves streamline with their health conditions. However, spectrum allocation for such applications is static and may become a challenge if a patient demands continuous monitoring of his/her health condition. Therefore, in healthcare IoT, it is essential to deploy devices with cognitive capabilities and functionalities to access the spectrum whenever it is required.

Social activities CR-IoT has been popular in social activities such as Intelligent Transportation Systems (ITS), which use multiple sensors on the road and in vehicles to monitor traffic and congestion on roads. In case of emergencies, Dedicated Short Range Communication (DSRC) with a channel bandwidth of 10 MHz is allocated, which can deliver small data over a short distance. Communication over longer distances requires the exchange of huge data, and DSRC won't be feasible. CR-IoT can eliminate this problem in a more efficient way. The traffic light management system is also supposed to incorporate cognitive capabilities in objects to perform a certain task related to traffic on the road.

2.5 Summary

In this chapter, we discuss the significance of IoT network that has revolutionized the technology world and has brought comfort to the people's lives. Moreover, the demand to incorporate intelligence and consciousness into cognitive devices inside the IoT network has been described. We also discuss OFDM modulation as a potential candidate for signal transmission in the CR-IoT network. Finally, the chapter highlights notable CR-IoT applications that include home, healthcare, industrial, etc.

References

[1] A. A. Khan, M. H. Rehmani, and A. Rachedi, "When cognitive radio meets the internet of things?" in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sep. 2016, pp. 469–474.

[2] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "Iot connectivity technologies and applications: A survey," *IEEE Access*, vol. 8, pp. 67 646–67 673, 2020.

[3] Y. Saleem, N. Crespi, M. H. Rehmani, and R. Copeland, "Internet of things-aided smart grid: Technologies, architectures, applications, prototypes, and future research directions," *IEEE Access*, vol. 7, pp. 62 962–63 003, 2019.

[4] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[5] K. L. Lueth, "State of the iot 2018: Number of iot devices now at 7b – market accelerating," https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018/, 2018.

[6] A. R. Biswas and R. Giaffreda, "Iot and cloud convergence: Opportunities and challenges," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 375–376.

[7] A. Daniel, A. Ahmad, and A. Paul, "Machine-to-machine communication - a survey and taxonomy," *Journal of platform Technology*, vol. 2, pp. 3–15, 06 2014.

[8] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[9] IDC, "Idc forecasts worldwide spending on the internet of things to reach 772 billion in 2018," https://informationmatters.net/internet-of-things-statistics/.

[10] M. Farrukh, A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Learning a switching bayesian model for jammer detection in the cognitive-radio-based internet of things," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 380–385.

[11] A. A. Khan, M. H. Rehmani, and A. Rachedi, "When cognitive radio meets the internet of things?" in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 469–474.

[12] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[13] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. . Neel, Y. Zhao, J. H. Reed, and W. H. Tranter, "A survey of artificial intelligence for cognitive radios," *IEEE Transactions on Vehicular Technology*,

vol. 59, no. 4, pp. 1578–1592, 2010.

[14] Z. Qin, X. Zhou, L. Zhang, Y. Gao, Y. Liang, and G. Y. Li, "20 years of evolution from cognitive to intelligent communications," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 6–20, 2020.

[15] W. P. Peng and Z. H. Ye, "Ofdm blending modulation systems in practical cognitive radio application," in *2011 4th International Congress on Image and Signal Processing*, vol. 4, 2011, pp. 2237–2240.

[16] R. Kumari Sathi and K. Babulu, "Spectrum sensing using energy algorithm for ofdm based cognitive radio," in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 1019–1024.

[17] H. A. Shah and I. Koo, "A novel physical layer security scheme in ofdm-based cognitive radio networks," *IEEE Access*, vol. 6, pp. 29 486–29 498, 2018.

[18] H. A. Mahmoud, T. Yucek, and H. Arslan, "OFDM for cognitive radio: merits and challenges," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 6–15, April 2009.

[19] M. A. Shah, S. Zhang, and C. Maple, "Cognitive radio networks for internet of things: Applications, challenges and future," in *2013 19th International Conference on Automation and Computing*, 2013, pp. 1–6.

Chapter 3

Physical Layer security in CR-IoT network

Even though CR is a smart and intelligent network nonetheless, it can be vulnerable to adversary attacks. This chapter describes the various threats in the CR network, followed by attacks at CR's physical layer. Afterward, security challenges for CR-IoT are presented. More specifically, jammer attacks in CR-IoT are discussed, along with the most relevant work associated with jammer detection techniques in the network.

3.1 Background

The proliferation of cognitive and smart wireless devices have panned out a high volume of data. Due to such a climb in wireless traffic volume, efficient spectrum utilization becomes critical and more exhausting. In this context, CR has emerged as a viable solution to effectively manage spectrum-related challenges, as introduced in the last **section.2.3**. Like other wireless networks, CR is vulnerable to various kinds of threats and attacks. Basically, attacks in CR is an exertion to desolate legitimate user transmission.Accordingly, CR attacks include active threats in which invaders interact legitimate users with wrong information. On the contrary, in passive threat of CR, attacker learns the communication medium and tries to obtain secure information. There has been tremendous work being carried out to develop robust methods to make the CR network attack-free. An overview of the attacks and detection methods for the CR network is presented in [1].

Cognitive capability and reconfigurability attributes of CR are highly associated with the security hazards in the network. Threats to CR's cognitive ability include transmitting the false report of sensing operation and blocking primary and secondary users channels. In contrast, attacks related to the reconfigurability property of CR exploit the configuration parameters during the acting phase of the CR operation and alter parameters [2]. CR network suffers attacks on all layers (from the application layer to the physical layer), and there are various attacks which are common to both CR and traditional wireless network. We now present an overview of different kinds of threats on all layers of the CR network.

1 − *Physical Layer*

The threats on this layer include Primary User Emulation Attacks (PUEA) [3], objective function attacks [4] and jamming attacks [5].

2 − *MAC Layer*

Denial of serivices [6], Spectrum Sensing Data Falsification (SSDF) [7] and channel saturation [8] are main threats at MAC layer.

3 − *Network Layer*

At this layer, Hello floods attacks [9], Sink hole attacks [10] are considerably frequent.

4 − *Transport Layer*

Threats include key depletion attacks [11].

5 − *Application Layer*

Software virus and malware are dominant threat at this layer [12].

Since our work mainly focuses on PHY-layer security, we will describe some notable PHY-layer threats in CR network.

3.2 PHY-Layer security threats in CR

Dynamic spectrum access (DSA) and spectrum sensing operations make security at PHY-layer in CR more demanding and sophisticated. Therefore, various techniques have been investigated and proposed by the researcher for the PHY-layer protection in CR. PHY-layer security in CR cognitive cycle can be categorized into the following classes. 1) Observation phase security 2) Action phase security.

Observation phase security

In the observation phase, spectrum sensing is vulnerable to various kinds of attacks. The most notable perils are location-based attacks, Primary User Emulation Attacks (PUEA), jamming attacks and Spectrum Sensing Data Falsification Attack (SSDF) [13]. PUEA attacks enforce SU to leave the PU channel or abstain from accessing the channel by sending fake signals. There has been a lot of work being carried out in this area that addresses PUEA attacks [3],[14] and [15]. Specifically, a localization-based defense technique has been developed which authenticate legitimate user presence by observing location and signal characteristics [16]. The paper [17] describes a joint approach that uses links based on location signature and cryptographic authentication method to abstain from PUEA attacks in the CR network.

In SSDF threats, attackers transmit false information of sensing operation to the network. SSDF attacks are common in cooperative sensing where false reports are being sent by participating in secondary devices. The work [18] presents attack-aware collaborative spectrum sensing approach against SSDF attacks in CR network. The paper [19] describes method for the defense against SSFD attacks. Bayesian models based method has been developed against SSDF attacks in CR [20].

Location-based threats include service denial attacks to the SU, which block the location of the primary user [21]. Accurate location and tracking are necessary for PU in cognitive radio so that SU can scan the activities of PU and consequently access the spectrum. However, if there are attacks in the link, SU will be misled regarding the location and existence of the PU, and eventually, interference between PU and SU will emerge. For accurate attack navigation, framework is presented in [22]. Jammer attacks are daunting threats in the CR network that aim to block legitimate user transmission and degrade overall network performance [23].

Acting phase security

Action-based security is considered during the action cycle of CR, and it is quite analogous to the conventional wireless communication network security. However, due to the dynamic nature of CR, security is rather tough and complex as compared to other wireless networks. Eavesdropping has been identified as a potential attack in the CR network during secondary user transmission [24]. Consequently, there have been techniques developed to combat such attacks, which include a multi-antenna-based method and a relay-based technique.

3.3 CR-Network under attack-Jamming threats

Since its inception during world war II [25], the jammer has evolved as more powerful and intelligent than ever before in the wireless communication network field. In this perspective, CR can be deployed to develop a smart jammer capable of learning the behavior of adversary communication. On the contrary, CR can facilitate to learn jammer behavior in the spectrum and, consequently, launch anti-jamming techniques to eradicate jamming effects [26]. Primarily, jamming is a kind of interference caused by a jammer (an external entity). Jammer aims to disrupts signal transmission by injecting its own signals, thereby causing loss or change of information encoded in a signal, deplete bandwidth, and degrade overall network performance. Jammer's existence can be traced back to world war II, where it was the first time utilized to block hostile transmission and deceive pilots.

Jammer types and their associated model have been extensively investigated and studied for the last couple of decades. Specifically, jammers for military and industrial applications have been an epicenter of many researchers and scientists [27]. We will focus on the sensing-based jamming model. For more details regarding other jamming models refer

[26]. There are two types of sensing-based jamming model [28].

1) Reactive Jammer: It starts sending signals whenever there is a transmission in the wireless channel. Reactive jammer must precisely detects activities in the spectrum for its operation to be effective. It is worth mentioning here that the reactive jammer is not identical to the smart jammer. Smart jammer [29] is equipped with cognitive capabilities capable of learning the environment and execute decisions. Smart jammer briskly learns the transmission strategies, modulation type, and other main characteristics of legitimate user signal.

2) Proactive Jammer: It transmits jamming signals with a pre-defined strategy without having knowledge of the wireless channel.

There are more practical jammers exist that integrate multiple jamming functionalities into one stand-alone jammer entity. In this context, a statistical jammer exploits temporal activities in a channel and maintains a histogram of transmission. Based on such knowledge, jammer launches attacks [30]. In [31], a strategic jammer is presented, which learns anti-jamming strategies inside the network and seizes the user transmission. A coordinated jammer is formulated in [32], which cooperates with other adversary agents in a network to invade attacks jointly. There is a Markov jammer described in that follows a Markov chain to pick a transmission band to be jammed [33].

After discussing potential jammer models and jammer types, we now present jammer detection methods that have been investigated and developed in the literature. In this perspective, [34], discusses method to reduce attacks in a wireless network. Spectral contents based jammer detection is given in [35]. The [36] describes jammer detection methods based on packet delivery ratio (PDR) and packet sensing ratio (PSR). These methods detect random and reactive jammer. The authors present a synchronization indicator in the transmission, which uses signal-to-jammer with noise ratio as a metric to detect jammer [37]. In [38], a joint approach to jammer detection and spectrum sensing is described and formulated. The work [5] and [39] consider jammer detection based on cyclostationary features extracted from a wideband spectrum and using NN technique. Table.3-A provides an over view of jammer type with associated detection

Table 3-A: Different Jammer types at CR PHY-Layer with detection methods

Ref and Year	***Jammer Type***	***CR Layer***	***Detection Strategies***
[40] 2012	Reactive	Physical	Cross-layer detection
[41] 2013	Reactive	Physical	Cross-layer detection
[42] 2013	Pulse	Physical	Intrusion detection
[38] 2015	Random	Physical	Weight Energy detection
[43] 2015	Spot	Physical	Compressed sensing and Energy detection
[44] 2015	Spot	Physical	Compressed sensing and Cylcostationary features
[39] 2017	Spot	Physical	Spectral correlation and Neural Network
[45] 2019	Smart	Physical	Dynamic Bayesian approach
[23] 2019	Smart	Physical	Dynamic Bayesian approach
[46] 2020	Smart	Physical	Dynamic Bayesian approach

methods in CR network.

3.4 Security in CR-IoT

CR-IoT has acquired remarkable rendition in many applications ranging from home automation to smart cities due to the manifestation of AI and cognitive capabilities into IoT objects as introduced and discussed in **section.2.2**. CR-IoT covers a wide range of applications such as health care, environment control, farming monitoring, and many more. Even though CR-IoT is self-aware, intelligent, and more robust, it can be vulnerable to various malicious threats. Consequently, the detection of such attacks and then mitigating their effects during transmission is one of the fundamental objectives of the CR-IoT network. In this perspective, [47] presents an overview of security challenges in CR-IoT networks. The security requirements and their associated challenges have

Table 3-B: Attacks on different layers in IoT network.

Sr.no	***Layers***	***Attacks***
01	Physical	Battery Drained [55] Jammer Attacks [45] Eavesdropping [56] Hardware Malfunction [57]
02	MAC/NTW	Collision attacks [58] Channel congestion [59] Battery exhausted attacks [60] Injecting false devices [61] Message alteration attack [62]
03	Application	Malicious code [63] Brute force attacks [54] Cross-site scripting attacks [54]

been briefly discussed in [48]. The [49] describes a newly emerging framework based on Software-Defined-Network (SDN) for security in IoT scenarios. The security and privacy aspect of IoT networks have been investigated in [50] in which several attacks are discussed in terms of complications and computation basis for the 5G IoT network. A brief survey about the structure of malware attacks in CR-IoT networks has been thoroughly described in [51]. Due to the ubiquitous connection of objects in IoT for several applications, it is highlighted in [52] that the attacker can access and attack user accessories (Google glasses, smartwatch, etc.) to learn the behavior of people. Hardware vulnerabilities of IoT devices have been discussed and investigated in [53]. It is mentioned that IoT devices remain exposed to physical interface and boot process vulnerabilities, which can be manipulated remotely. Moreover, [54] discusses about the security requirements and challenges for IoT network. Table.3-B gives an overview of different attacks on layers in IoT network.

3.5 CR-IoT susceptible to Jamming attacks

Jamming attacks have been considered as most devastating and disrupting in the CR-IoT network [26]. Jammer attempts to interrupt normal signal transmission and eventually corrupt the signals. Furthermore, jammer depletes the transmission bandwidth and degrades overall network performance [64]. There has been relatively little work investigating and developing jammer detection techniques for the CR-IoT network in this regard. The paper [65] studies jamming attacks for time-sensitive wireless applications and present malicious attack detection based on jamming attack detection based on estimation (JADE) scheme with implementation for a wireless network. A reactive jammer detection scheme is presented in [66] for tactical wireless networks. A more recent work related to the jammer detection for IoT scenario is discussed in [67] in which a reactive jammer from another network access the transmission specification of a hidden terminal and can interrupt with its terminal emulation (HTE) attack. Where as in [68] context-aware hidden units attack for the CR-IoT network is presented. The effects of jamming attacks on the IEEE802.11 network performance is investigated in [69]. The paper also highlights the different types of jamming strategies. The authors propose a channel assignment algorithm for the CR-IoT network under jamming attacks while considering time-sensitive data traffic [70]. The work [71] describes jamming mitigation strategies for multi-input-multi-output CR-IoT network. The paper [72] presents jammer resistance technique that doesn't require jammer knowledge in a system.

A Bayesian framework is developed by learning dynamic models of the spectrum data based generalized state vector containing an OFDM modulation signals to detect jammer for the CR-IoT network [45]. The proposed work detect jammer inside the subcarrier of the OFDM modulation at two inferences levels of DBN model. Moreover, the work [23] presents two implementations of the dynamic Bayesian network, namely, single and bank-parallel DBNs for jammer detection in the CR-IoT network. The work proposes to learn a dynamic model for subcarriers of OFDM either by using one single DBN or deploying individual DBN (bank-parallel) for each sub-carrier of OFDM modulation. Table.3-C highlights abnormalities detection (jammer) for CR and CR-IoT network.

Table 3-C: Abnormalities (Jammer) detection in CR and CR-IoT.

Note-able work for Abnormality detection		
Ref and Year	Approach	Outcomes
[44] 2015	Cyclostationary Features	Jammer Detection
[73] 2016	Q-Learning Approach	Jammer Detection
[74] 2017	Neyman-Pearson Approach	Jammer Detection
[45] 2019	Bayesian Inference Approach	Jammer Detection
[23] 2019	Bayesian Inference Method	Jammer Detection
[75] 2020	GAN and Bayesian Inference	Abnormalities Detection
[76] 2020	Generative Model	Abnormalities Detection

3.6 Proposed method comparison with other work

We aim to develop a jammer detection method based on deep learning and probabilistic networks. Such a technique effectively handles higher dimension data and performs state estimation. The proposed method is different from the work presented [43],[73],[27] and [5]. In [39] jammer detection is achieved using cyclostationary feature extraction and compressing sensing method. Whereas, [73] describes jammer detection method based on Q-learning approach. The most recent approach for abnormality detection (most probably jammer attacks) is presented in [76] based on generative models and [23] presents jammer detection using dynamic Bayesian network. The generative models are not good in dealing with uncertainties, and DBN can't handle high dimensional data [75]. Nevertheless, the modern intelligent network generates and communicates a vast amount of wireless data. Therefore, a more robust and powerful method is presented in this work by exploiting deep learning and probabilistic models jointly to perform inference and prediction for high dimension signals and eventually detect abnormalities in the spectrum. We proposed the DBN model for low-dimensional signals to capture jammer signals inside OFDM modulated transmission in the CR-IoT network presented in **chapter.4**. **chapter.6** then describes deep learning, and the DBN method is jointly implemented to deal with high dimensional data to perform state estimation tasks and eventually capture abnormalities and detect abnormalities.

3.7 Summary

This chapter presents security threats in the CR network, precisely several kinds of attacks at the PHY-layer in the CR network. We discuss the limitations of the current work related to the jammer detection in the CR-IoT network and highlight the proposed method's motivation in this thesis work.

References

[1] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[2] A. Attar, H. Tang, A. V. Vasilakos, F. R. Yu, and V. C. M. Leung, "A survey of security challenges in cognitive radio networks: Solutions and future research directions," *Proceedings of the IEEE*, vol. 100, no. 12, pp. 3172–3186, 2012.

[3] N. Sureka and K. Gunaseelan, "Detection defense against primary user emulation attack in dynamic cognitive radio networks," in *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, vol. 1, 2019, pp. 505–510.

[4] M. Jia, X. Wang, Q. Guo, X. Gu, and Z. Yu, "A multi-bit decision cooperative spectrum sensing algorithm in mobile scenarios based on trust valuations in cognitive radio context," in *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2014, pp. 335–339.

[5] T. Nawaz, L. Marcenaro, and C. S. Regazzoni, "Stealthy jammer detection algorithm for wide-band radios: A physical layer approach," in *2017 IEEE 13th Inter-*

national Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2017, pp. 79–83.

[6] Wang Weifang, "Denial of service attacks in cognitive radio networks," in *2010 The 2nd Conference on Environmental Science and Information Application Technology*, vol. 2, 2010, pp. 530–533.

[7] S. Chatterjee and P. S. Chatterjee, "A comparison based clustering algorithm to counter ssdf attack in cwsn," in *2015 International Conference on Computational Intelligence and Networks*, 2015, pp. 194–195.

[8] V. Rajpoot and V. S. Tripathi, "A dedicated ccc based data channel selection scheme with proactive hand-off in cognitive radio network," in *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2018, pp. 1–6.

[9] S. Magotra and K. Kumar, "Detection of hello flood attack on leach protocol," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014, pp. 193–198.

[10] M. Guerroumi, A. Derhab, and K. Saleem, "Intrusion detection system against sink hole attack in wireless sensor networks with mobile sink," in *2015 12th International Conference on Information Technology - New Generations*, 2015, pp. 307–313.

[11] V. Nguyen, P. Lin, and R. Hwang, "Energy depletion attacks in low power wireless networks," *IEEE Access*, vol. 7, pp. 51 915–51 932, 2019.

[12] T. C. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, 2008, pp. 1–8.

[13] R. K. Sharma and D. B. Rawat, "Advances on security threats and countermeasures for cognitive radio networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1023–1043, 2015.

[14] R. Yu, Y. Zhang, Y. Liu, S. Gjessing, and M. Guizani, "Securing cognitive radio networks against primary user emulation attacks," *IEEE Network*, vol. 30, no. 6, pp. 62–69, 2016.

[15] D. Roy, T. Mukherjee, M. Chatterjee, and E. Pasiliao, "Defense against pue attacks

in dsa networks using gan based learning," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[16] R. Chen, J. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[17] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 286–301.

[18] A. A. Sharifi and M. J. Musevi Niya, "Defense against ssdf attack in cognitive radio networks: Attack-aware collaborative spectrum sensing approach," *IEEE Communications Letters*, vol. 20, no. 1, pp. 93–96, 2016.

[19] L. Li, F. Li, and J. Zhu, "A method to defense against cooperative ssdf attacks in cognitive radio networks," in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, 2013, pp. 1–6.

[20] Y. Fu and Z. He, "Bayesian-inference-based sliding window trust model against probabilistic ssdf attack in cognitive radio networks," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1764–1775, 2020.

[21] X. Zhang and C. Li, "Constructing secured cognitive wireless networks: Experiences and challenges," *Wireless Communications and Mobile Computing*, vol. 10, pp. 50–69, 01 2010.

[22] A. W. Min and K. G. Shin, "Robust tracking of small-scale mobile primary user in cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 778–788, 2013.

[23] A. Krayani, M. Farrukh, M. Baydoun, L. Marcenaro, Y. Gao, and C. S.Regazzoni, "Jammer detection in m-qam-ofdm by learning a dynamic bayesian model for the cognitive radio," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[24] A. Banerjee and S. P. Maity, "Cognitive radio networks with energy harvesting and eavesdropping-emulation resilience," in *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2020, pp. 873–875.

[25] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Communications Surveys Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[26] M. Aref, S. Jayaweera, and E. III, "A survey on cognitive anti-jamming communications," *IET Communications*, vol. 14, 05 2020.

[27] R. Muraleedharan and L. Osadciw, "Jamming attack detection and countermeasures in wireless sensor network using ant system," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6248, 06 2006.

[28] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1746–1759, 2014.

[29] M. Camilo, D. Moura, J. Galdino, and R. M. Salles, "Anti-jamming defense mechanism in cognitive radios networks," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–6.

[30] M. Pajic and R. Mangharam, "Anti-jamming for embedded wireless networks," in *2009 International Conference on Information Processing in Sensor Networks*, 2009, pp. 301–312.

[31] Y. Gwon, S. Dastangoo, C. Fossa, and H. T. Kung, "Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning," in *2013 IEEE Conference on Communications and Network Security (CNS)*, 2013, pp. 28–36.

[32] Wenjing Wang, M. Chatterjee, and K. Kwiat, "Collaborative jamming and collaborative defense in cognitive radio networks," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2011, pp. 1–6.

[33] M. Aref and S. Jayaweera, *Spectrum-Agile Cognitive Interference Avoidance Through Deep Reinforcement Learning*, 08 2019, pp. 218–231.

[34] W. Chen, Y. Zhang, and Y. Wei, "The feasibility of launching reduction of quality (roq) attacks in 802.11 wireless networks," in *2008 14th IEEE International Conference on Parallel and Distributed Systems*, 2008, pp. 517–524.

[35] S. Machuzak and S. K. Jayaweera, "Reinforcement learning based anti-jamming with

wideband autonomous cognitive radios," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, 2016, pp. 1–5.

[36] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attack strategies and network defense policies in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1119–1133, 2010.

[37] R. Bhojani and D. R. Joshi, "An integrated approach for jammer detection using software defined radio," *Procedia Computer Science*, vol. 79, pp. 809–816, 12 2016.

[38] J. Mohammadi, S. Stańczak, and M. Zheng, "Joint spectrum sensing and jamming detection with correlated channels in cognitive radio networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 889–894.

[39] T. Nawaz, D. Campo, M. O. Mughal, L. Marcenaro, and C. S. Regazzoni, "Jammer detection algorithm for wide-band radios using spectral correlation and neural networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 246–251.

[40] C. Sorrells, L. Qian, and H. Li, "Quickest detection of denial-of-service attacks in cognitive wireless networks," in *2012 IEEE Conference on Technologies for Homeland Security (HST)*, 2012, pp. 580–584.

[41] L. Qian, X. Li, and S. Wei, "Cross-layer detection of stealthy jammers in multihop cognitive radio networks," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, 2013, pp. 1026–1030.

[42] Z. M. Fadlullah, H. Nishiyama, N. Kato, and M. M. Fouda, "Intrusion detection system (ids) for combating attacks against cognitive radio networks," *IEEE Network*, vol. 27, no. 3, pp. 51–56, 2013.

[43] M. O. Mughal, K. Dabcevic, L. Marcenaro, and C. S. Regazzoni, "Compressed sensing based jammer detection algorithm for wide-band cognitive radio networks," in *2015 3rd International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa)*, 2015, pp. 119–123.

[44] M. O. Mughal, T. Nawaz, L. Marcenaro, and C. S. Regazzoni, "Cyclostationary-based jammer detection algorithm for wide-band radios using compressed sensing," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*,

2015, pp. 280–284.

[45] M. Farrukh, A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, “Learning a switching bayesian model for jammer detection in the cognitive-radio-based internet of things,” in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 380–385.

[46] A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, “Smart jammer detection for self-aware cognitive uav radios,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7.

[47] S. Hameed, F. Idris Khan, and B. Hameed, “Understanding security requirements and challenges in internet of things (iot): A review,” *Journal of Computer Networks and Communications*, vol. 2019, pp. 1–14, 01 2019.

[48] F. Khan and S. Hameed, “Understanding security requirements and challenges in internet of things (iots): A review,” *ArXiv*, vol. abs/1808.10529, 2019.

[49] Y. Liu, Y. Kuang, Y. Xiao, and G. Xu, “Sdn-based data transfer security for internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 257–268, 2018.

[50] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, “Iot survey: An sdn and fog computing perspective,” *Computer Networks*, vol. 143, pp. 221 – 246, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618305395

[51] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, “Anatomy of threats to the internet of things,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1636–1675, 2019.

[52] J. Deogirikar and A. Vidhate, “Security attacks in iot: A survey,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, pp. 32–37.

[53] G. Hernandez and D. Buentello, “Smart nest thermostat a smart spy in your home,” 2014.

[54] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, “An in-depth analysis of iot security requirements, challenges, and their countermeasures via

software-defined security," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 250–10 276, 2020.

[55] C. Gehrmann, M. Tiloca, and R. Höglund, "Smack: Short message authentication check against battery exhaustion in the internet of things," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2015, pp. 274–282.

[56] B. Ahuja, D. Mishra, and R. Bose, "Fairness-aware subcarrier allocation to combat full duplex eavesdropping and jamming attacks in iot," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[57] A. Sarthi, D. Gurjar, C. Sai, P. Pattanayak, and A. Bhardwaj, "Performance impact of hardware impairments on wireless powered cognitive radio sensor networks," *IEEE Sensors Letters*, vol. PP, pp. 1–1, 05 2020.

[58] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge computing-assisted internet of things," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[59] V. B. Misic, Jun Fang, and J. Misic, "Mac layer security of 802.15.4-compliant networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, 2005, pp. 8 pp.–854.

[60] A. Reziouk, E. Laurent, and J. Demay, "Practical security overview of ieee 802.15.4," in *2016 International Conference on Engineering MIS (ICEMIS)*, 2016, pp. 1–9.

[61] S. Suhail, C. S. Hong, M. A. Lodhi, F. Zafar, A. Khan, and F. Bashir, "Data trustworthiness in iot," in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 414–419.

[62] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Threats to networking cloud and edge datacenters in the internet of things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016.

[63] S. A. Hinai and A. V. Singh, "Internet of things: Architecture, security challenges and solutions," in *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, 2017, pp. 1–4.

[64] H. A. Bany Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Spectrum assignment

in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1904–1913, 2018.

[65] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1746–1759, 2014.

[66] A. Marttinen, A. M. Wyglinski, and R. Jäntti, "Statistics-based jamming detection algorithm for jamming attacks against tactical manets," in *2014 IEEE Military Communications Conference*, 2014, pp. 501–506.

[67] M. Hossain and J. Xie, "Detection of hidden terminal emulation attacks in cognitive radio-enabled iot networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[68] ——, "Third eye: Context-aware detection for hidden terminal emulation attacks in cognitive radio-enabled iot networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 214–228, 2020.

[69] I. Harjula, J. Pinola, and J. Prokkola, "Performance of ieee 802.11 based wlan devices under various jamming signals," in *2011 - MILCOM 2011 Military Communications Conference*, 2011, pp. 2129–2135.

[70] H. A. Bany Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1904–1913, 2018.

[71] L. Li and C. Chigan, "A virtual mimo based anti-jamming strategy for cognitive radio networks," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[72] P. Zhou, Q. Wang, W. Wang, Y. Hu, and D. Wu, "Near-optimal and practical jamming-resistant energy-efficient cognitive radio communications," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2807–2822, 2017.

[73] S. Machuzak and S. K. Jayaweera, "Reinforcement learning based anti-jamming with

wideband autonomous cognitive radios," in *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, 2016, pp. 1–5.

[74] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent reinforcement learning based cognitive anti-jamming," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[75] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Ai-based abnormality detection at the phy-layer of cognitive radio by learning generative models," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 21–34, 2020.

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

Chapter 4

Learning Dynamic Probabilistic Models for CR-IoT Spectrum

To develop autonomous CR radios, it is essential to perceive how CR behaves in the operating environment. In other words, understand how CR signals change as time evolves and predicting the dynamic state of the CR in the system. Such understanding facilitates estimating future states and assisting in modeling CR's interaction with the spectrum through dynamic models. This chapter introduces a framework based on the Bayesian network to exploit temporal relationships among random variables involved in a network. Dynamic Bayesian network is presented and proposed that model CR dynamic behavior inside the spectrum with time evolution. Moreover, the abnormality detection method is formulated and presented that provides a facility to detect abnormal signals in the CR-IoT network spectrum.

4.1 CR-an actor in Interactive and Cognitive Environment

A cognitive radio (CR) is an adaptive and intelligent radio capable of learning the radio environment and adjusting its operating parameters. An imperative feature of CR is the strength of autonomous learning and self-programming. CR incorporates AI techniques based on machine learning (ML) and neuroscience methods to develop intelligent radio devices. Such methods allow CR to tune its internal operating parameters according to the variations and transitions occurring in the environment. In this perspective, CR devices operate in a dynamic environment that imitates a spectrum evolving with time and the interaction of CR with the spectrum encompasses a dynamic process driven by a CR device. CR is an actor in the spectrum that imitates CR's operating environment to achieve specific goals (access channel for transmission). CR situates targets and struggles to get those objectives by interacting with the spectrum and perform some actions. Fig.4.1 shows CR as an agent which interacts with the radio environment and performs certain actions to meet desire goals. CR, with its cognitive capabilities, tries to maintain an equilibrium between its state and the environment. In this context, an Interactive and Cognitive Environment (ICE) can be explained as [1],

A typical physical ambient equipped with cognitive capabilities and facilities acquired from artificial intelligence endowment predicated on information and communication technologies..

CR should be aware of its radio environment to achieve cognitive tasks, and in this connection, spectrum sensing is considered a significant operation in CR. CR senses the spectrum by sending signals using either single or multiple antennas during the cognitive cycle and CR can be esteemed as an agent in the ICE, where the spectrum acts as an environment for a CR. As time evolves, CR learns different models from experiences and eventually becomes smarter by adapting the changes it has to apply to sustain dynamic equilibrium and stability with the operating and external environment. CR should be equipped with learning and reasoning abilities, and recent advancements in ML and Software-Defined Radio (SDR) have expedited CR to achieve its learning and reasoning objectives successfully. SDR facilitates radio parameter implementation in software that allows parameter tuning (transmit power, coding schemes, modulation schemes, and sensing policy) in software without hardware intervention. ML provides the facility to use learning algorithms that assist CR in regulating its parameters and adapt changes according to the environment. To introduce a higher level of cognition and intelligence, self-awareness (SA) is introduced in CR devices as mentioned in **section.2.3**. Dynamic environment evolves with time and encodes information about the dynamics of the system for every time instant. Therefore, time statistics is a major ambassador that mimics system dynamics. A time-frequency analysis tool can be deployed to extract dynamic features from the signal to capture time information precisely [2]. Time-frequency retains both time and frequency information of each signal inside the spectrum. Such features may include bandwidth, frequency, and transmitting power. In the dynamic model, a signal's characteristics provide a method to discursively infer the hidden or unobservable state for each entity. The spectrum represents a dynamic environment, while the signal in the spectrum exhibits the observation through which the entity's hidden state can be measured. The signals inside the spectrum exhibit observations from which an imperceptible state of the objects can be measured in a dynamic environment. The representation of the signal is the foundation that facilitates entities and interaction modeling. Signal representation in terms of extracted features is prestigious and discerning in learning dynamic models. Such features are heavily influential in the overall learning steps of dynamic models. Therefore, features extracted to represent signal behavior together with an associated model with optimal parameters are major steps to build a dynamic model representing changing and interacting entities inside a spectrum. Stand-alone models are used to represent single entities, while linked coupled models describe multiple entities in the ICE environment. Statistical and state estimation signal processing techniques can be deployed to make CR capable of estimating observed entities' current state (including CR itself) and predicting future state. Such methods can be considered as Bayesian filters to learn different models. Probabilistic Graphical Models (PGM) employ Bayesian filters that are adaptive to model non-stationary objects' actions and behaviors where coupled multiple PGMs can be deployed to model interacting among different entities.

PGM models have gained a lot of attention over the last decades to represent uncertainty in many fields such as speech processing, computer vision, signal processing, sequential data modeling, bioinformatics, probabilistic robotics, error-correcting coding theory, and

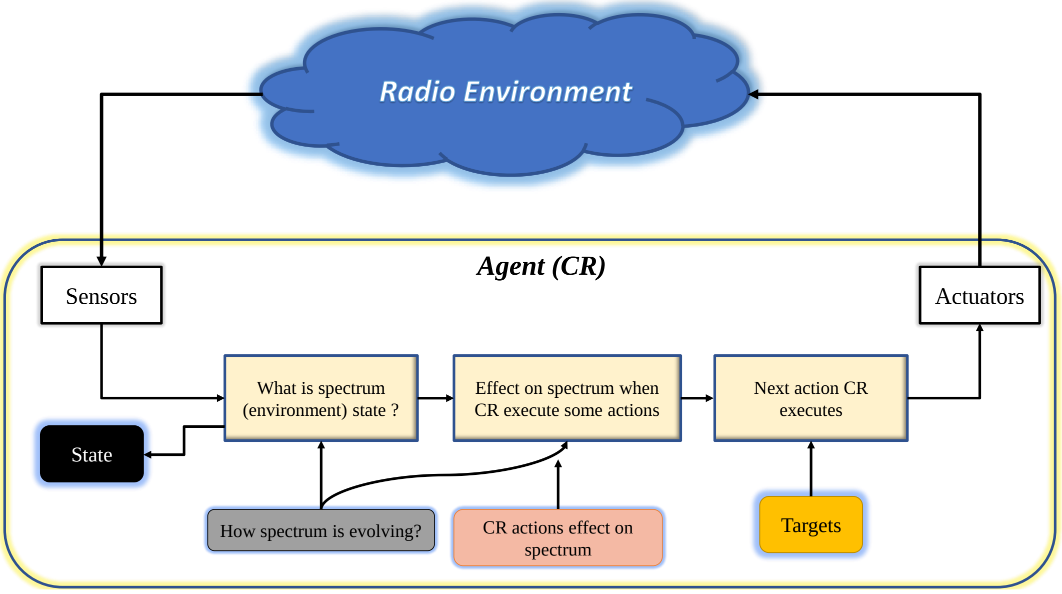

Radio Environment
Agent (CR)
Sensors
Actuators
What is spectrum (environment) state ?
Effect on spectrum when CR execute some actions
Next action CR executes
State
How spectrum is evolving?
CR actions effect on spectrum
Targets

Figure 4.1: CR as an agent in the operating radio environment [3].

artificial intelligence. PGM is a statistical model that captures the conditional independent relationship between interacting random variables and encodes complex joint multi-variant probability distribution using graphs. Inference and learning tasks can be accomplished easily once the graph structure of PGM is known. Inference includes computing the marginal distribution of one or more random variables, and learning covers estimating the parameters of probability functions [4].

4.2 Probabilistic Graphical Models

For intelligent radio systems, machine learning and self-awareness are the fundamental building blocks where the chief objective is to capture hidden patterns or trends in empirical data using statistical and computational techniques. Accordingly, the task is to automatically identify and recognize data structure and learn a model based on extracted features. Given new data samples, a learned model should generalize well to provide useful information. To account for this, ML and SA heavily rely on many fields such as statistical and probability theory, optimization methods, cognitive and neuroscience, and calculus. PGM models are essential in all three learning types (supervised, unsupervised, and reinforcement) and have emerged to be the method of choice for modeling uncertainty in many fields such as speech processing, computer vision, communication and signal processing, error coding, time-series modeling and etc. We can deal with uncertainty in two ways, i-e, extensionally, and intensionally. Extension systems, which are also called rule-based systems, are computationally efficient and robust, but they are semantically weak in measuring uncertainty. In contrast, the intensional system is computationally expensive and semantically strong.

Probability theory and graph theory have been integrated into PGM to provide a joint probability distribution in terms of a well-represented graph model by exploiting conditional independencies among the random variables. Graph models cope with uncertainty and complexity problems efficiently and adequately in the field of engineering by relaxing the computational burden of model learning and inference through conditional independence assumptions [5].

There are many versions of graphical statistical models for instant, Bayesian network (BN), dynamic Bayesian network (DBN), factor analysis (FA), Hidden Markov model (HMM), factorial HMM, Kalman Filter (KF), Boltzmann machines, and the Ising model. However, BNs present the most appropriate representation of relative influences among real world facts and thus, has become one the method for solving problems contains uncertainty [6]. BNs have been deployed in the field of AI to model complex interaction among causes and interactions. However, Bayesian reasoning and inference methods have recently gained attention and have become a method of choice in information fusion obtained from different sources. We will focus on BN and explain its different versions in the following sections.

4.3 A popular graph model defining relation among random variables in a network

A model based on a graphical illustration that defines conditional dependence among a group of random variables is known as BN. It is a special kind of model that is presented as a directed acyclic graph (DAG) [7]. In DAG, nodes show a graphical representation of objects and events of the real world and are called variables or states. To describe the causal relationship between nodes, an arc(edge) is used. For example, the causal relationship between variables $\widetilde{X_1}$ and $\widetilde{X_2}$ is represented by an edge leading from the cause variable directed toward the affected variable. Consider a graph model shown in Fig.4.2 in which $\widetilde{B}$ and $\widetilde{C}$ are conditionally independent given $\widetilde{A}$. A graphical model comprises of group of nodes $\widehat{n} = [1, 2, 3....\widehat{N}]$, probabilities distribution P for each variable and a set of dependencies (edges) between variables. Fig.4.2 depicts BN network where nodes show a random variable, and the absence of arc describes conditional independence assumption between variables. The probability of joint event $P(\widetilde{B}, \widetilde{C})$ is given as [8],

$$P(\widetilde{B}, \widetilde{C}) = P(\widetilde{B} \mid \widetilde{C})P(\widetilde{C}) \tag{4.1}$$

The joint probability in a graph model is expressed as the product of the conditional

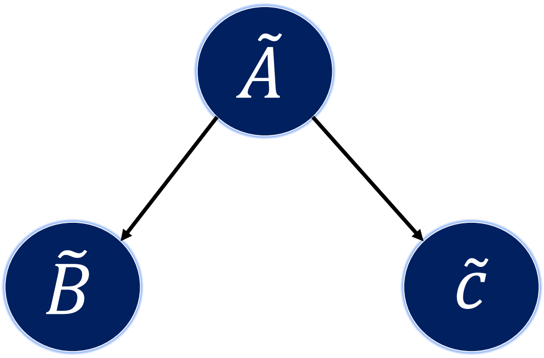


Figure 4.2: A simple BN network.

probabilities for each node in a network $[\widetilde{A_1}, \widetilde{A_2}, ...\widetilde{A_{1\check{n}}}]$ using the chain rule and is given as,

$$P\left(\widetilde{A_1}...., \widetilde{A_{\widehat{n}}}\right) = \prod_{i=1}^{\widehat{n}} P\left(\widetilde{A_i} \mid P_a(\widetilde{A_i})\right) \tag{4.2}$$

$P_a\left(\widetilde{A_i}\right)$ shows parent set of node. The joint probability distribution, the conditional probabilities, and the structure of the BN can be deployed to find the likelihood of each node holding one of its states. In the BN network, learning and inference are two main processes that are described in the following subsections.

4.3.1 Inference process in Bayesian Network

In BN, the inference is a process of finding the probability of each node's state when other variables are known. It is a process to infer the distribution adequately on a specific group of variables provided other random variables are known in a network. To explain inference, consider the partition $\widetilde{H_{\widehat{L}}} = \widetilde{F_{\widehat{N}}} \cup \widetilde{G_{\widehat{M}}}$ and let $\widetilde{F_{\widehat{N}}} = [f_0, f_1, f_2, ...f_{\widehat{N}-1}]$ shows set of hidden variables, and $\widetilde{G_{\widehat{M}}} = [g_0, g_1, g_2, ...g_{\widehat{M}-1}]$ represents visible variables, where $\widehat{L} = \widehat{M} + \widehat{N}$. Let $\widetilde{R_k}$ be a subgroup of $\widetilde{H_{\widehat{L}}}$. The inference process aims to determine the conditional distribution over $\widetilde{R}$ given the observed variables $\widetilde{G}$, which can be expressed as $P\left(\widetilde{R_k} \mid \widetilde{G}\right)$. if $\widetilde{R_k} \subseteq \widetilde{G}$, we find that probability distribution function is trivially equal to $P\left(\widetilde{R_k} \mid \widetilde{G}\right) = \prod_{k=1}^{K} \delta(r_k - g_k)$, where $\delta_f = 1$ for $f = 0$ and $\delta_f = 0$ otherwise. A nontrivial case arises when $\widetilde{R_k} \subseteq \widetilde{F}$. The Bayes rule is used to find the probability distribution function (pdf) as [9],

$$P(\widetilde{R_k} \mid \widetilde{G}) = \frac{P(\widetilde{R_k}, \widetilde{G})}{P(\widetilde{G})} \tag{4.3}$$

The probability distribution function over $\widetilde{R_k}$ and $\widetilde{G}$ is obtained by marginalizing $P(\widetilde{H_{\widehat{L}}})$ over the group of hidden variables as,

$$P(\widetilde{R_k} \mid \widetilde{G}) = \sum_{\widetilde{F}|\widetilde{R_k}} P(f, \widetilde{R_k}, \widetilde{G}) P(\widetilde{G}) \tag{4.4}$$

Inference in BN can achieved by exact probability propagation in a single connected network, and or by approximate inference such as Monte Carlo inference techniques, Helmholtz machine inference, Gibbs sampling, variational inference, and etc.

4.3.2 Learning process in Bayesian Network

Learning includes the adjustment of the BN model parameters so that the pdfs defined by the network adequately describe the observed data's statistical behavior [9]. Learning techniques make it possible to complete the network's missing beliefs in a case in which conditional probabilities are unknown.

Consider $\breve{M}^i$ as a BN model associated with the parameters $\breve{\Phi}^i$ of some probability distribution for the i^{th} model with the given data $\breve{Z}^i_{\widehat{L}}$. Let $P(\breve{M})$ and $P(\breve{\Phi}^i \mid \breve{M}^i)$ be the prior distribution on the group of models and the space of parameters in these models, respectively. Consider model generates some data, we can estimate a parameter $\breve{\Phi}^i$ over a data $\breve{Z}^i_{\widehat{L}}$ as [9],

$$P(\breve{M}^i \mid \breve{Z}^i_{\widehat{L}}) = \frac{P(\breve{M}^i)}{P(\breve{Z}^i_{\widehat{L}})} \int_{\breve{\Phi}^i} P(\breve{Z}^i_{\widehat{L}} \mid \breve{\Phi}^i) P(\breve{\theta} \mid \breve{M}^i) d\breve{\Phi}^i \tag{4.5}$$

The maximum likelihood estimate $\breve{\theta}$ for a given model $\breve{M}$ is obtained from:

$$\breve{\Phi}^i = \underset{\breve{\Phi}^i}{argmax} log P(\breve{Z}^i_{\widehat{L}} \mid \breve{\Phi}^i) \tag{4.6}$$

We can define the goal of learning in scenario where all $\breve{Z}$ are not observable in a model as,

$$\breve{\Phi}^i = \underset{\breve{\Phi}^i}{argmax} \sum_{\chi} log P(\widetilde{Y}, \widetilde{X} \mid \breve{\Phi}^i) \tag{4.7}$$

P shows joint probability distribution specified by the model. The Expectation-Maximization (EM) algorithm can be used to minimized cost function as:

$$J(\breve{\Phi}^i) = -log \sum_{\chi} log P(\widetilde{Y}, \widetilde{X} \mid \breve{\Phi}^i) \tag{4.8}$$

4.4 Dynamic Bayesian Network

Dynamic Bayesian network (DBN) is preferable method in modeling the time-dependent process, and it falls under the category of BN [10]. In the DBN model, conditional dependencies between random variables are modeled across and within the time slots. The Markovian condition should be satisfied for the states of any system to define it as a DBN. We can express the hidden state variables $\widetilde{X}$ and measured variables $\widetilde{Y}$ over a time instant $\breve{T}$ as,

$$P(\widetilde{X}, \widetilde{Y}) = \prod_{t=1}^{\breve{T}-1} P(\breve{x}_t \mid \breve{x}_{t-1}) \prod_{t=1}^{\breve{T}-1} P(\breve{y}_t \mid \breve{x}_t) P(\breve{x}_0) \tag{4.9}$$

Three parameters are used to completely define DBN which are given as:

1 Transition pdfs that define temporal relativity between the occurrence of states.

2 Observation probability distributions that define observation node's dependencies with different nodes in a network at time t.

3 $P(\breve{x}_0)$ shows initial probability distribution.

The nature of state variables (hidden and observable) form a DBN structure, which can be continuous, discrete, or a mixture of both continuous and discrete. DBN undergoes the following four tasks:

1. Inference: Given some known observation and initial probability distribution, estimate pdf of hidden states.

2. Decoding: For the sequence of hidden states, determine the best-fitting probability values.

3. Learning: Estimate the DBN parameters that best fit the observed data and determine the best model for the system.

4. Pruning: Determine the most important nodes in the DBN and eliminate less or unimportant nodes from the network.

Some of the popular DBN models are now discussed in the following section.

4.4.1 The Markovian Model

HMM is a method for describing probability distribution on a sequence of observations and has been implemented in many applications such as speech processing, computer vision, time-series modeling, error correction and coding, and AI [11]. Fig.4.3 shows HMM model where $\widetilde{B}_{t_1}^1$ denotes observations over time t and $\widetilde{A}_{t_1}^1$ is the hidden or latent spaces from the observer. The probability distribution of a sequence of states and observation is expressed in a HMM model jointly as [12],

$$P(\widetilde{A}^K_{t_{1:T}}, \widetilde{B}^K_{t_{1:T}}) = P(\widetilde{A}^1_{t_1})P(\widetilde{B}^1_{t_1} \mid \widetilde{A}^1_{t_1}) \prod_{t=2}^{T} P(\widetilde{A}^1_{t_2} \mid \widetilde{A}^1_{t_1})P(\widetilde{B}^1_{t_1} \mid \widetilde{A}^1_{t_1}) \tag{4.10}$$

where, $t \in [t_1, t_2, ...T]$ shows time instants and $k \in [1, 2, 3.......K]$ represent particular state at time instant t. The state variable $\widetilde{A}^1_{t_1}$ is discrete in HMM and can take integer values. Given a sequence of observations in HMM we need to specify the followings: probability distribution over initial state $\widetilde{A}^1_{t_1}$, the $M \times M$ matrix defining state transition probabilities $P(\widetilde{A}^1_{t_2} \mid \widetilde{A}^1_{t_1})$ and the final network defining $P(\widetilde{B}^1_{t_1} \mid \widetilde{A}^1_{t_1})$. If the observations are discrete symbols taking on one of D values, the output model can be fully specified by a $M \times D$ observation matrix. $P(\widetilde{B}^1_{t_1} \mid \widetilde{A}^1_{t_1})$ can be modeled in many different form such as Gaussian, mixture of Gaussian and neural network for real-value observation vectors.

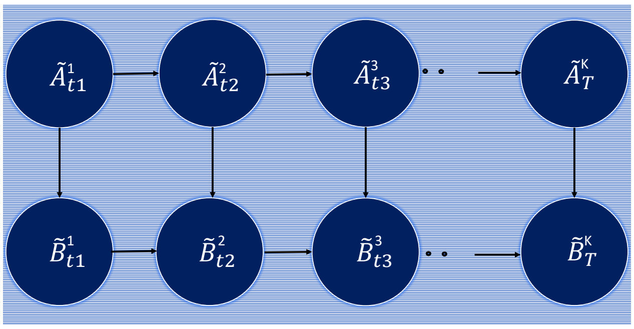


Figure 4.3: A simple HMM model.

4.4.2 Linear filtering-Kalman Filter (KF)

KF is a well-known filter that is used to provide an estimation of unknown variables provided some measurements observed over time. KF provides an estimation of a linear dynamic system under Gaussian noise $N \sim (0, 1)$. The set of state-space equations can define the state of an agent d to model its behavior as a linear dynamic system as [13],

$$\tilde{X}^d_k = F\tilde{X}^d_{k-1} + B\tilde{U}^d_{k-1} + w_{k-1} \tag{4.11}$$

where F is a matrix which describes transition from previous state $\tilde{X}^d_{k-1}$ to next state $\tilde{X}^d_k$, B is control input and w_{k-1} is noise which is $w_{k-1} \sim N(0, Q)$.

$$\tilde{Z}^d_k = H\tilde{X}^d_k + v_k \tag{4.12}$$

where $\tilde{Z}^d_k$ shows measurement or observation, H is a matrix containing measurement values, and v_k is measurement noise which is $v_k \sim N(0, R)$. equation.(4.11) and equation.(4.12) present process model and measurement model respectively.

KF filtering comprises of prediction/propagate and update/correction stages. The prediction stage is given as [13]:

Prediction

Prediction state estimation

$$\tilde{X}^d_k = F\tilde{X}^d_{k-1} + B\tilde{U}^d_{k-1} \tag{4.13}$$

Prediction error covariance

$$P^d_k = F\hat{P}^d_{k-1}F^T + Q \tag{4.14}$$

Update

Measurement value

$$\tilde{Y_k}^d = \tilde{Z}^d_k - H\tilde{X}^d_k \tag{4.15}$$

Kalman Gain

$$K_k = P^d_k H^T (R + HP^d_k H^T)^{-1} \tag{4.16}$$

Update state estimation

$$\tilde{X}^d_k = \tilde{X}^d_{k-1} + K_k \tilde{Y}^d_k \tag{4.17}$$

Update error covariance

$$P^d_k = (I - K_k H)P^d_{k-1} \tag{4.18}$$

The term P is a state error covariance. In the update stage, $\tilde{Y_k}$ is calculated which is measurement residual. $\tilde{Y_k}$ is also called as innovation and it is the difference between measurment $\tilde{Z}_k$ and the estimated measurement $H\hat{X}_k$.

To deal with non-linear systems, extended Kalman filter (EKF) [14] is a famous method that deals with a non-linearity. Unlike KF, which deals with the linear systems, EKF uses filtering based on Bayesian methods for the non-linear dynamic systems with Gaussian noise.

$$\tilde{X}_k^d = \mathfrak{F}(\tilde{X}_{k-1}^d, U_{k-1}^d) + w_{k-1} \tag{4.19}$$

$$\tilde{Z}^d = \mathfrak{H}(\tilde{X}_k^d) + v_k \tag{4.20}$$

$\mathfrak{F}$ and $\mathfrak{H}$ are the non-linear functions and the goal is to make function linear using Taylor series approximation. Given non-linear functions $\mathfrak{F}$ and $\mathfrak{H}$, prediction and innovation are computed respectively. To linearize the model about the current estimate, in each time step, the first-order partial derivative of a function f with respect to a vector of each model is determined to obtain Jacobian matrix as,

$$\mathfrak{F}_{k-1} = \frac{\partial f}{\partial x}|_{x_{k-1},u_{k-1}} \tag{4.21}$$

$$\mathfrak{H}_k = \frac{\partial h}{\partial x}|_{x_k} \tag{4.22}$$

To handle non-linear Gaussian problems, Cubature Kalman Filter (CKF), Unscented Kalman Filter (UKF) can also be deployed as a variant version of KF. Particle filter is also used in non-linear filtering problems.

4.5 Learning a Bayesian Model for Jammer Detection

CR can be considered as an agent of the IDE where spectrum represents an environment in which CR signal changes and evolves with time, carrying information in frequency bands. The interaction of CR with spectrum can be modeled as a dynamic process driven by CR cognitive cycle. In this context, a probabilistic framework is considered to devise and understand the dynamic relationship between CR and the spectrum. In a generalized coordinate system, the moving agent state is defined as a generalized state vector comprising its position information and l times derivatives and can be expressed as,

$$X_r = [x_r \dot{x_r} ....x_{(l)}^{(r)}]^T \tag{4.23}$$

$x \epsilon \mathbb{R}^d$ with $d - dimension$ space. We can exploit the temporal dependence of the agent by inferring current state dependence on the past values, i-e $P(X_{r,k}|X_{r,k-1})$. The $X_{r,k}$ describes the state of an $agent(r)$ in a dynamic system at a given time instant k, and $X_{r,k-1}$ shows agent state at previous time instant $k - 1$. Similarly, we can describe CR behavior inside a spectrum evolving with time using generalized states (GS) vector containing CR's state information and derivatives. As mentioned in **section.2.3.2**, we have considered OFDM modulation for PHY-layer transmission in a CR-IoT network. The OFDM modulated signal consists of a set of N sub-carriers:

$$C = \{C_1, C_2, \dots, C_N\}, \tag{4.24}$$

each sub-carrier is divided into M symbols in time domain, forming a $N \times M$ time-frequency grid respectively of the transmitted OFDM signal. For any given sub-carrier consisting of M symbols, there is a temporal evaluation between consecutive symbols that describe how amplitude and phase values are dynamically changing in a specific sub-carrier. We can form a GS containing FFT information obtained from the received OFDM signal in the receiver section. The motivation for choosing FFT output is explained as:

First, to analyze the signal statistically using amplitude and phase information.

Second, an anti-jamming technique can be implemented to detect jammer before the signal goes to demodulation and mitigate jammer effect at this level, thus reducing receiver complexity.

To evaluate the dynamics of the amplitudes and phases related to consecutive symbols and how they are evolving with time we consider the derivatives ($\dot{a}$, $\dot{p}$) of both amplitudes

(a) and phases (p), and the generalized state vector can be defined at each time instant k for a specific sub-carrier as,

$$X_{k,C_n} = [\, a \; p \; \dot{a} \; \dot{p} \,] \quad n = \{1, 2, \ldots, N\}, \quad C_n \in \boldsymbol{C} \tag{4.25}$$

Where a, p are amplitude and phase while $\dot{a}$, $\dot{p}$ are corresponding derivatives. A set of generalized state vectors corresponding to each sub-carrier is defined as:

$$X = \{X_{k,C_1}, X_{k,C_2}, \ldots, X_{k,C_N}\}, \tag{4.26}$$

After obtaining a set of state vectors describing the CR's behavior in the spectrum under normal situation (no abnormalities), the Switching Dynamic Bayesian Network (SDBN) model can be implemented to model spectrum evolution as a dynamic system and perform state estimation tasks. The proposed DBN is shown in Fig.4.4. DBN enables to include dependencies between involved random variables as time evolves and also facilitates the representation of different inference levels. Consequently, here the lowest level of inference corresponds to the observed received carrier amplitude and phase Z_k information from the spectrum. States, X_k, represent a medium inference

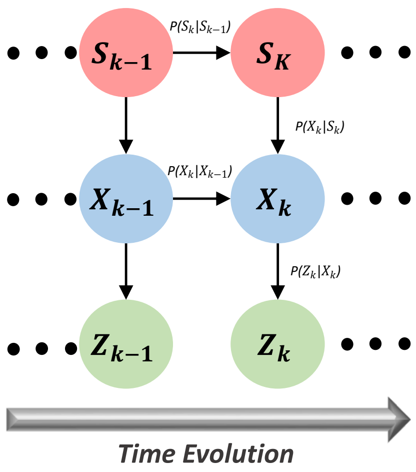


Figure 4.4: Proposed DBN model comprises of two parts: (Discrete and Continuous) to detect jammer in the Spectrum.

level which encodes continuous information. Super-states S_k correspond to the top level of inference which manifest the discretization of the continuous states. Additionally, arrows represent conditional probabilities between the involved variables. Vertical arrows facilitate to describe causalities between both, continuous and discrete levels of inference and observed measurements. Horizontal arrows explain temporal causalities between hidden variables. For each temporal slice, DBN defines three conditional dependencies.

- $P(Z_k|X_k)$ defines the probability of obtaining an observation based on CR current state inside the spectrum.
- $P(X_{k+1}|X_k)$ shows the probability of moving to next state of a CR given current state inside the spectrum.
- $P(X_k|S_k)$ represents probability of being in a certain state given the active superstate.

4.5.1 Steps to learn DBN model

Learning the switching DBN undergoes four essential steps. We now explain each step in detail.

A) Learning superstates. Unsupervised learning alogirthm Self Organizing Maps (SOMs) [15] is applied to obtain discrete regions of the spectrum named as superstates. Discrete states or superstates acts as a switching variables that enable the activation of associated dynamic models to predict future state (X_{k+1}). SOM receives X_k and produces a set of learned superstates $\boldsymbol{S}$ where similar information (quasi-constant derivatives) are valid, such that:

$$S = \{S_1, S_2, \dots, S_L\}, \tag{4.27}$$

where $S_k \in \boldsymbol{S}$ and L is the total number of superstates. SOM can miniature multiple dynamic patterns appearing on the same state-space coordinate.

B) Learn discrete transition models. By observing the activated superstate over time, it is possible to estimate a set of temporal transition matrices encoding the probabilities of passing from a current superstate to another one. Such matrices take into consideration the time spent in current superstate for encoding transition probabilities, facilitating the estimation of $P(S_k|S_{k-1}, t_k)$, where t_k encodes the time spent in the current superstate S_{k-1}.

C) Regions properties. A region S_k is represented by the variables ξ_{S_k}, Q_{S_k} and $\psi_{S_k^i}$ which encode the mean value, the covariance matrix of clustered states and a threshold value where linear models are valid, respectively. Such a threshold is defined as,

$$\psi_{S_k} = E(d_{S_k}) + 3\sqrt{V(d_{S_k})} \tag{4.28}$$

d_{S_k} contains distance between neighbouring super-states, $E(.)$ and $V(.)$ calculate mean and variance of a input data respectively. This threshold describes a boundary that determines where the model is valid.

D) Learn dynamic continuous models. Two models are required to analyze and make inferences about a dynamic system evolving with time. These models are measurement and dynamic models.

Measurement Model: This model defines the relation between observations Z_k and states X_k at time instant k and maps observations into a state such that:

$$Z_k = HX_k + v_k \tag{4.29}$$

where Z_k shows measurement vector, H is a measurement matrix, and v_k is measurement noise that is assumed to be Gaussian, i-e $v_k \sim N(0, R)$, and R is a covariance matrix.

Dynamic Model: The dynamic model describes the progression of the system state with time. Basically, it explains the temporal dependence and exploits the relationship between current and future state such that:

$$X_k = AX_{k-1} + BU_{S_{k-1}} + w_k, \tag{4.30}$$

where $A = [A_1 \;\; A_2]$ is a dynamic model matrix: $A_1 = [I_2 \;\; 0_{2,2}]^\intercal$ and $A_2 = 0_{4,2}$. I_n represents a square identity matrix of size n and $0_{l,m}$ is a $l \times m$ null matrix. $B = [I_2 \Delta k \;\; I_2]^\intercal$ is a control input model. w_k represents the prediction noise which is assumed to be Gaussian $w_k \sim N(0, Q)$ for all X_k along with a covariance matrix Q. The variable $U_{S_{k-1}}$ is a control vector that encodes the spectrum's action when it is inside a superstate S_k and it depends on the super-state S_k corresponds to the active cluster at a time k, such that:

$$U_{S_k} = [\dot{a}_{S_k} \quad \dot{p}_{S_k}]^\intercal, \tag{4.31}$$

Accordingly, it is possible to estimate the probability of obtaining a future spectrum's state given its present state $P(X_k|X_{k-1}, S_{k-1})$ for each superstate S_{k-1}.

4.5.2 Testing the learned DBN

To make inferences by employing the learned DBN, we proposed to use a probabilistic switching model called Markov Jump Particle filter (MJPF) [16]. MJPF makes use of Particle Filter (PF) with Sequential Important Resampling (SIR) algorithm along with the bank of Kalman Filters (KF). Such a realization expedites an inference of random variables at a discrete and continuous level in a coordinated fashion. Consequently, MJPF provides inference at two levels.

Level 1: Continuous Layer: At this layer, the inference is achieved based on measurements. Hence, it corresponds to the observed received spectrum information. Predictions $P(X_k|X_{k-1})$ are performed by using a bank of KF according to the active super-state S_k.

Level 2: Discrete Layer (Higher abstraction level): On this level, PF is used to perform inference tasks to predict the next super-state based on the current active super-state, i-e, $P(S_k|S_{k-1})$.

Applying MJPF makes it possible to detect abnormalities at two levels, i-e at continuous and discrete layers. Accordingly, two abnormality measurements are defined for detecting such abnormal behaviour inside the spectrum, based on the Bhattacharrya distance between prediction $p(X_k^*|X_{k-1}^*(S_k^*))$ and

- probability of being inside the predicted superstate of particle $p(X_k^*|S_k^*)$.

$$db1 = -\ln \int \sqrt{p(X_k^*|X_{k-1}^*(S_k^*))p(X_k^*|S_k^*)}dX_k^*; \tag{4.32}$$

- evidence $p(z_k|X_k^*)$ to have solutions near the measurement:

$$db2 = -\ln \int \sqrt{p(X_k^*|X_{k-1}^*(S_k^*))p(Z_k|X_k^*)}dX_k^*; \tag{4.33}$$

where, $(.)^*$ indicates the considered particle and (S_k^*) means that the prediction depends on the superstate. The value of $db1$ relates to the similarity between prediction of the state and the likelihood to be in the predicted superstate. The value of $db2$ relates to the similarity between the state prediction and the continuous state evidence related to the new observation in each superstate.

4.5.3 Performance evaluation of DBN under abnormal signals in CR-IoT spectrum

In the last section, we introduce the GS to encode spectrum information as a state and it's first order derivative in the context of CR, and presented learning of DBN based on GS. In the following section, we present the performance of DBN to detect the abnormalities (jammer attacks) in the CR-IoT spectrum. Since we mentioned, DBN provides anomalies at two levels; therefore, abnormalities are captured at two layers. For the experiments, OFDM signal based on IEEE 802.11ah configurations is assumed to be under consideration. The data is generated and modulated using 16-QAM, mapped onto 64 sub-carriers, followed by cyclic prefix (CP) addition and transformed into the time domain by using IFFT. The received signal is assumed to be affected by AWGN. After CP is stripped off and FFT is performed and we form a GS according to the equation.4.25. The output data is divided into two sets: one contains only the clean data (No jammer attack) for the training phase and the second set consists of the data affected by the jammer attack for the testing phase. The jammer attacks any of the sub-carriers in the OFDM signal according to the following scenarios.

- **Scenario 0 (***Normal Situation***):** This scenario is used to learn the DBN model as shown in Fig.4.5(a) for the normal behavior of the CR network by applying the clean data during the training phase. The learned DBN is utilized later on the other scenarios to determine the deviations of the new behavior from the normal situation. After learning the normal situation during the *training phase* we define different scenarios to test the proposed method (*testing phase*). The anomalies are detected based on the abnormality measurements $db1$ equation.4.32 and $db2$ equation.4.33.

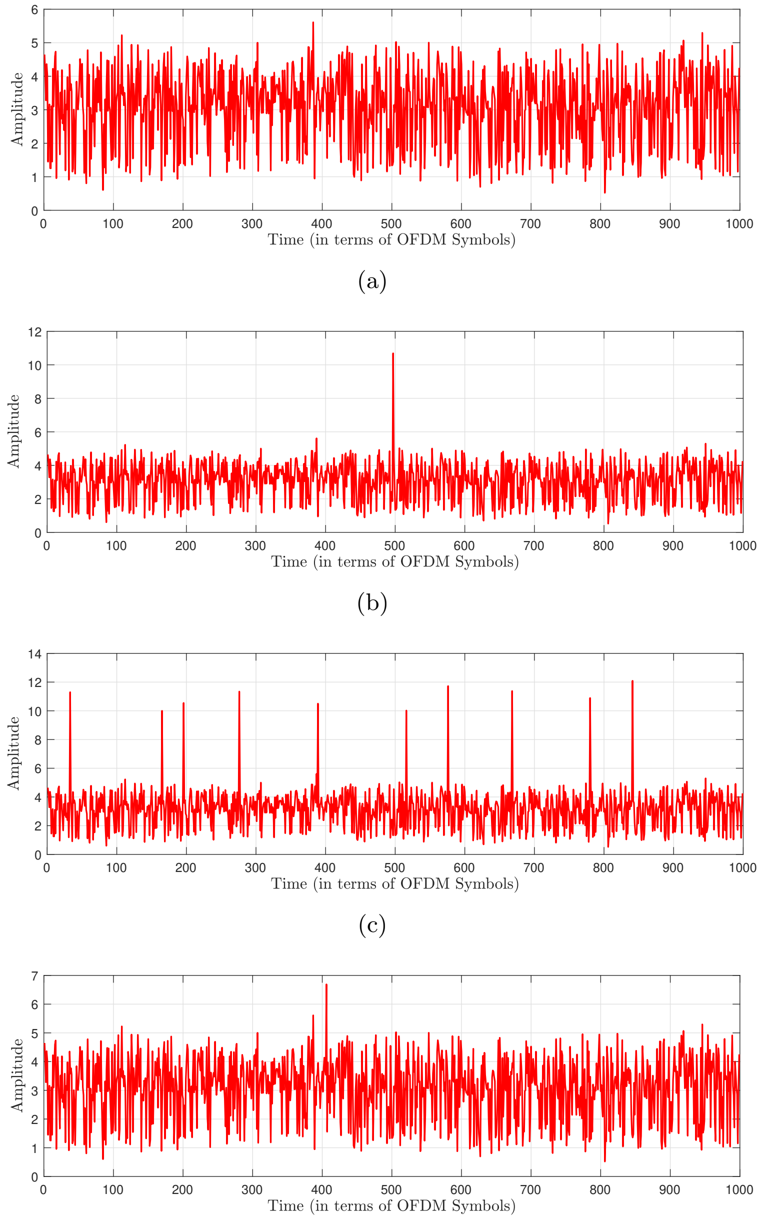
- **Scenario 1 (***Single Jammer attack***):** When the jammer attempts to disrupt the transmission of the primary user by attacking one symbol of the OFDM sub-carrier as shown in Fig.4.5(b).

- **Scenario 2 (***Multiple Jammer attacks***):** In this case, the jammer attacks different symbols of the OFDM sub-carrier, Fig.4.5(c).

- **Scenario 3 (***Jammer attack with low power***):** In this case, the jammer attacks single symbol of the OFDM sub-carrier with low power, Fig.4.5(d).

The detection of the abnormal situation is based on a calculated threshold for each abnormality measurement. Correspondingly, Fig.4.5(b) depicts the observation of **Scenario 1** where a single attack is present in the sub-carrier. By observing the abnormality measurements, it is possible to detect and locate the attacked symbol in that sub-carrier by comparing it with the threshold. The high peak means an abnormality and it is greater than threshold as displayed in Fig.4.6(a) and Fig.4.6(b). In this case we are able to detect the jammer at discrete and continuous levels.

In **Scenario 2** the jammer attacks multiple symbols. Our model is able to detect multiple attacks at discrete and continuous levels as shown in Fig.4.7(a) and Fig.4.7(b).



(a)

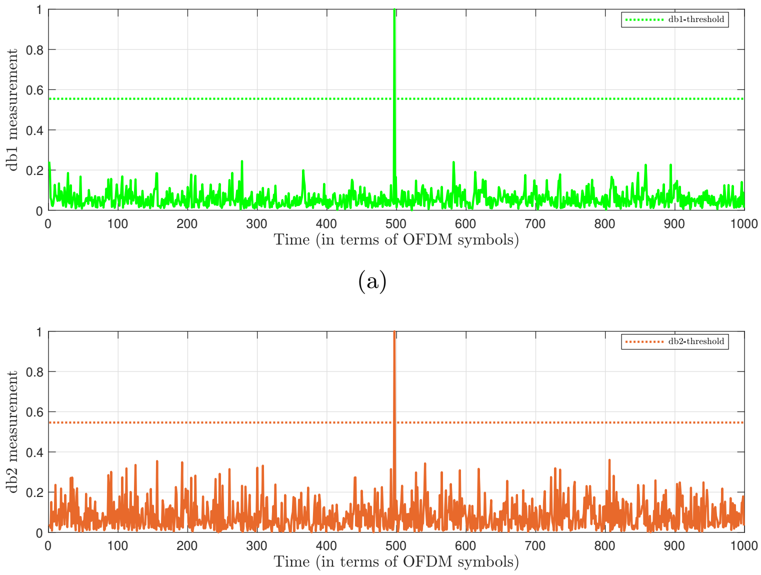
(b)

(c)

(d)

Figure 4.5: Different scenarios (a) Clean Data (b) Single Attack (c) Multiple Attack (d) Single attack with low power.

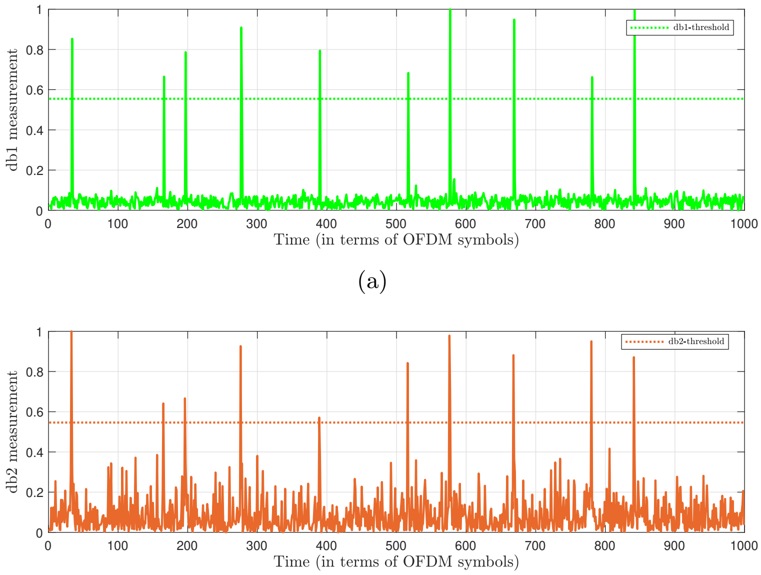
The jammer in **Scenario 3** attacks one symbol with lower power. In this situation, it is possible to detect jammer only at discrete level (db1) as indicated in Fig. 4.8(a), while at continuous level (db2) our method is not able to detect the attack, Fig.4.8(b).



(a)

(b)

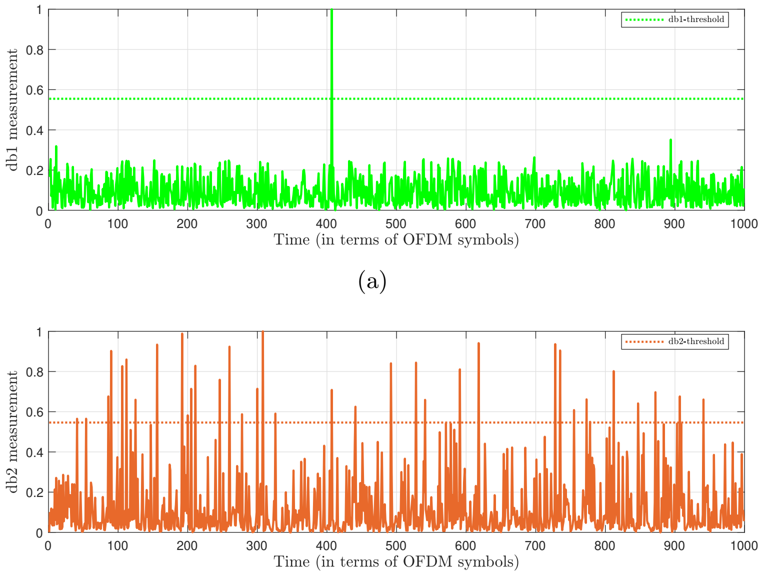
Figure 4.6: Scenario 1 Single attack (a) Abnormality detection at discrete level (db1) (b) Abnormality detection at continuous level (db2).



(a)

(b)

Figure 4.7: Scenario 2 Multiple attack (a) Abnormality detection at discrete level (db1) (b) Abnormality detection at continuous level (db2).



(a)

(b)

Figure 4.8: Scenario 3 Low power attack (a) Abnormality detection at discrete level (db1) (b) Abnormality detection at continuous level (db2)

4.5.4 DBN comparison with conventional detector

Conventional Energy Detector (ED) has been the most popular spectrum sensing method used in CR due to its simplicity. It compares the signal energy with a predefined threshold to decide if the spectrum is occupied or not. Subsequently, we use adaptive version of ED in order to provide a fair comparison with the proposed DBN. The detection is based on two hypotheses:

$$H_0 : \quad r(t) = s(t); \tag{4.34}$$

$$H_1 : \quad r(t) = s^J(t); \tag{4.35}$$

$r(t)$ is the received OFDM symbol. H_0 represents the hypothesis of a normal situation when the symbol is not attacked, while H_1 represents the hypothesis of an abnormal situation when the jammer has attacked the symbol. The decision of H_0 and H_1 is based on a predefined threshold T compared with the energy of each received sample. The performance of the ED is evaluated based on the probability of detection (P_d) which is calculated as follows:

$$P_d = P(E_{s_i} > T) \;\; i = 1, 2, ..., M; \tag{4.36}$$

where E_{s_i} is the energy of the detected symbol i. We adopt the traditional ED provided with a small memory, giving it a statistical knowledge of the symbol amplitude before and after jamming. This knowledge gives us an adaptive threshold which is able to detect the jammer. The threshold is obtained by calculating the difference values between the amplitude of the attacked symbols before and after jamming, such that:

$$D_i = ||s_i - s_i^J||; \tag{4.37}$$

where s_i represent the symbol before jamming and s_i^J after jamming. Accordingly, the result is a set of euclidean distance $\boldsymbol{D}$ related to the symbols under attack, such that,

$$D = \{D_1, D_2, \ldots, D_M\}; \tag{4.38}$$

The threshold (T) is calculated as follows:

$$T = |E(\boldsymbol{D})|^2; \tag{4.39}$$

Where E is the mean value of ($\boldsymbol{D}$). For ED we use the same data as for DBN in order to see the difference between the two systems. As discussed in the previous sections, our proposed DBN model is based on a statistical analysis of the sensed sub-carrier, it can predict and estimate future states and deals with the whole OFDM symbols. On the other hand, the conventional energy detector receives signal and performs energy test statics for a given time instant. To make a fair comparison between the two systems we provided the ED with a limited memory for memorizing previous and current state (before and after an attack) of the symbol. The scenarios mentioned before are done to see how our proposed DBN perform with different situations regarding the jammer (changing its power, increasing the number of attacks). We compare the adaptive ED with DBN by plotting the Probability of detection with respect to the number of attacks.

The performance of ED degrade as the attacks increase where DBN is always able to

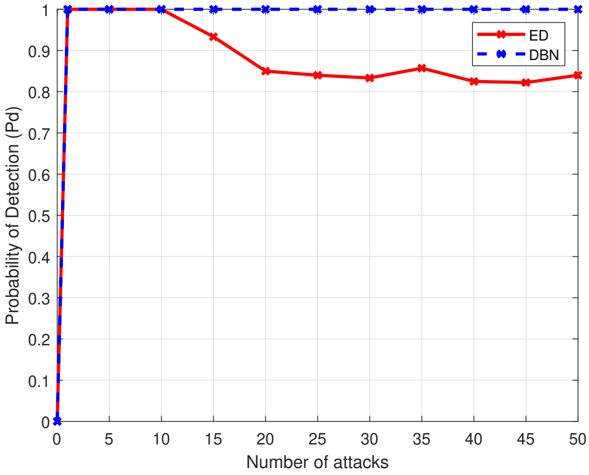


Figure 4.9: Performance of ED and DBN in terms of P_d as the number of attacks increases.

detect attacks with stable performance as shown in Fig.4.9. The advantage of our proposed DBN model with respect to the adaptive ED is that it is able to detect and locate attacked symbol in the sub-carrier, where the adaptive ED is not apt to identify affected symbols. This is due to the fact that ED has a limited memory which allows it to sample observed symbol at a given time instant.

4.6 Single and Bank-Parallel DBN implementation for N-Subcarriers in OFDM

The investigation on multiple sub-carriers modulated with different M-ary QAM in the OFDM signal under jammer attacks in the CR-IoT network was carried out and performed further. In this context, the general objective while considering multiple sub-carriers is to track the jammer's behavior and analyze how it is jumping between different sub-carriers, to detect the attacked frequency and predict the next sub-carrier that the jammer might attack in the next time instant. Jammer detection is achieved by implementing two proposed systems **Single**, and **Bank-Parallel**, and then performance evaluation of both systems are examined. Additionally, the effect of changing SOM size to QAM modulation is analyzed.

The proposed DBNs realize a Probabilistic Switching Models (PSMs) which provide agility to draw inference for each time slice about the spectrum at discrete and continuous levels by employing a combination of Particle Filter (PF) for discrete level and Kalman Filter (KF) for the continuous level as briefly introduced in **section.4.5**. DBNs are suitable for describing signals' dynamics due to their capability of modeling future instances based on observations in a probabilistic way. Such a characteristic is useful when performing tracking and recognizing abnormalities in CR.

Accordingly, we consider a CR-IoT network consisting of a group of Cognitive Radio Users (CRUs) and a jammer trying to disrupt the communication as shown in Fig. 4.10. CRUs sense the spectrum continuously and try to detect the abnormal situation. The radio spectrum contains OFDM waveforms based on IEEE 802.11ah standard, which is adopted in this work. OFDM divides the band channel into many narrower sub-carriers allowing different users to transmit simultaneously with different orthogonal frequencies. In the work presented and discussed in 4.5, only one sub-carrier is picked to employ the proposed method supposing that OFDM use 16-QAM for all the sub-carriers in the set equation.4.26. Instead, here we consider multiple sub-carriers modulated with different QAM (4, 16, 64, and 256-QAM according to the standard IEEE 802.11ah). Exploiting FFT output which consists of amplitude and phase of each symbol makes the spectrum sensing easier and less complex where CRUs can scan the entire grid. Moreover, by using the Amplitude and Phase information at this level, permits to implement a jammer detection technique before demodulation of the signal which reduces the receiver com-

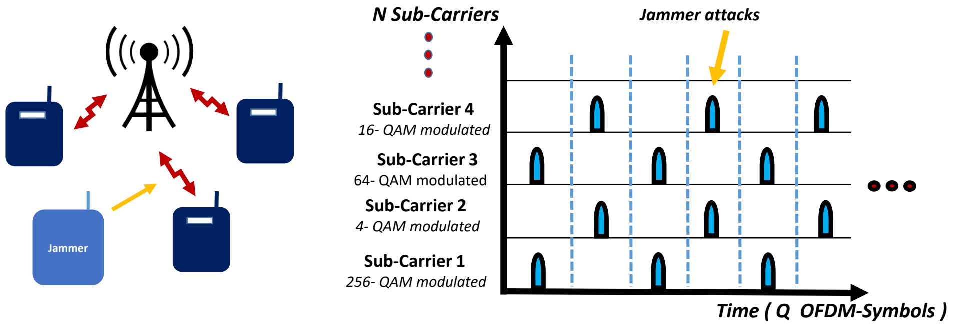


Figure 4.10: Spectrum of the M-ary QAM modulated OFDM users in the CR-IoT Network under jammer attacks.

plexity. The jammer attacks at different time instants by jumping from one frequency into another. We assume that there is perfect synchronization between the transmitter and receiver. To evaluate the dynamics of the amplitudes and phases related to consecutive symbols and how they are evolving with time, we consider the derivatives ($\dot{a}$, $\dot{p}$) of both amplitudes (a) and phases (p), and the generalized state vector is formed at each time instant k for a specific sub-carrier as given in equation.4.25. To infer and detect the jammer, we proposed to use the MJPF introduced in **section.4.5.2**. As mentioned before, the MJPF uses Particle filter to make inferences at discrete level. Additionally, each considered particle employs a Kalman Filter corresponding to the dynamic model learned for the corresponding value of the superstate at the continuous level.

4.6.1 Single Dynamic Bayesian Network

As shown in figure 4.11, we use the set of state vectors corresponding to each sub-carrier 4.26 in to learn a single DBN. During the Offline Learning Process, $\boldsymbol{X}$ is considered as input of the SOM which outputs a set of neuron $\boldsymbol{S}$. In this approach, $\boldsymbol{S}$ consists of the discretization of the entire spectrum. However, single DBN keeps a memory of the spectrum's behaviour in time and frequency domain. Additionally, a single abnormality indicator is provided during the online process.

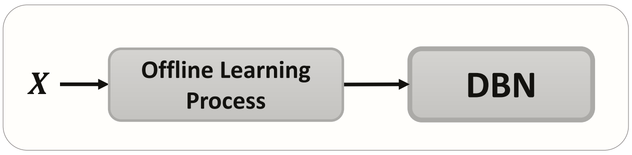


Figure 4.11: Single DBN.

4.6.2 Bank-Parallel Dynamic Bayesian Network

In this approach, we don't have any correlation between the sub-carriers, where the spectrum's behaviour at each sub-carrier X_{k,C_n} is processed individually (Fig.4.12). Accord-

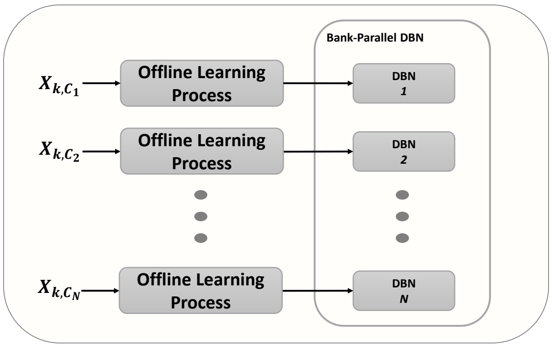


Figure 4.12: Bank-Parallel System.

ingly, for each X_{k,C_n} we learn a DBN, such as:

$$DBN = \{DBN_1, DBN_2, \dots, DBN_N\}, \tag{4.40}$$

In the online process, a MJPF is applied on each DBN_n providing an abnormality signal, such as:

$$db1 = \{db1_1, db1_2, \dots, db1_N\}, \tag{4.41}$$

4.6.3 Simulation Parameters

We use the OFDM system based on the IEEE 802.11ah standard. We use a simulated OFDM signal consists of $N = 64$ sub-carriers and $Q = 1000$ symbols. The source generates random independent data. Each sub-carrier of the OFDM signal is modulated with different QAM modulation. For our experiments, we pick four sub-carriers with different QAM modulation (4, 16, 64, 256). The received signal is assumed to be affected by additive white Gaussian noise (AWGN) with zero mean and power spectral density σ_w^2. Data is cleaved into two data sets: first set contains clean data (without jammer attacks) which is used during the training phase and the second one includes jammer's attacks which is used during testing, immediately after the cyclic prefix (CP) is removed and FFT is performed on received data. We consider that jammer launches attacks into multiple sub-carriers with equal power.

Table 4-A: Precision measurements for a Single-DBN

SOM size	4	16	64	128	256	512	1024
AUC (%)	98.98	99.75	99.05	99.89	99.71	99.93	**99.385**

4.6.4 Performance evaluation of Single and Bank-Parallel DBN

The performance of Single and Bank-Parallel DBN models are evaluated under multiple attacks and results are shown in terms of ROC curves which consist of Probability of Detection (P_d) and Probability of False Alarm (P_f), and Area Under Curve (AUC). The abnormality measurement ($db1$) is used to calculate the (P_d) and (P_f) respectively. (P_d) is the number of times where abnormalities (related to jammer attacks) are correctly identified, while (P_f) are the times where anomalies are wrongly assigned to normal symbols. 4.13 illustrates the ROC curve obtained from Single DBN when a different number of neurons is selected. It is evident from 4.13 and Tab 4-A. that 1024 neurons are the most appropriate for a Single DBN.

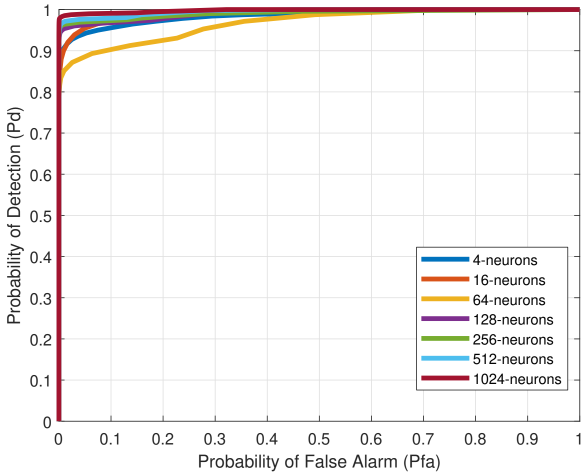
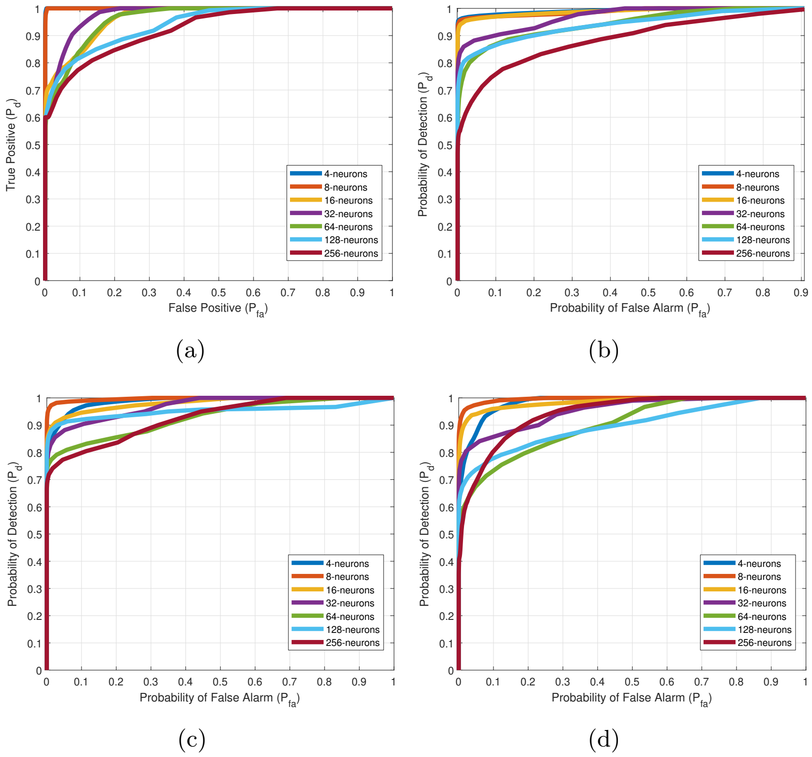


Figure 4.13: ROC for a Single-DBN under attacks while varying SOM size.

Whereas, Fig.4.14(a), 4.14(b), 4.14(c) and 4.14(d) present Bank-Parallel DBN ROC curves. For every ROC curve, each DBN deploys different QAM and optimum SOM size is analyzed. In case of 4-QAM, the optimum SOM size is 4 (see Fig. 4.14(a) and Table. 4-B). In 16-QAM is 4 (refer Fig. 4.14(b) and Table. 4-B). For 64-QAM, is 8 (see Fig.

4.14(c) and Table. 4-B), and for 256-QAM is 8 (refer Fig. 4.14(d) and Table. 4-B). We believe that the optimum number of neurons depend on the data and the number of symbols. For the simulated data used in our experiments and from the obtained results, we can notice that the Bank-Parallel system performs well for a small number of neurons, where the Single system performs well for a large number of neurons. This is due to the fact that Single-DBN uses the generalized state vector consisting of a large number of samples ($4Q$ symbols), which is 4 times the number of symbols used in Bank-Parallel system. After using the optimum number of neurons obtained previously to make a



(a)

(b)

(c)

(d)

Figure 4.14: ROC FOR for an individual DBN in Bank-Parallel-DBN employs different M-QAM under attacks while varying SOM size (a) 4-QAM (b) 16-QAM(c) 64-QAM (d) 256-QAM.

fair comparison between the two systems. The performance of both systems is somehow similar as shown in Fig. 4.15.

Table 4-B: Precision measurements for a Bank-Parallel-DBN

	SOM size						
	4	***8***	***16***	***32***	***64***	***128***	***256***
4-QAM	**99.99**	99.89	96.65	97.82	96.64	98.23	96.38
16-QAM	**99.86**	99.79	99.51	99.7	97.46	96.51	91.76
64-QAM	99.16	**99.89**	99.61	99.67	96.31	95.11	97.13
256-QAM	98.55	**99.83**	98.87	99.2	96.1	91.36	95.35

4.6.5 Discussion

We can deploy either of the proposed methods depending on the receiver complexity and specific task. For instant, Single DBN learns single vocabulary for all sub-carriers, whereas, Bank-Parallel DBN learns multiple vocabularies corresponds to each sub-carrier which increases complexity. Subsequently, implementing bank parallel DBN is suitable for the source characterization tasks. Tracking the jammer and keeping its profile history in the entire spectrum is much more convenient in Single DBN.

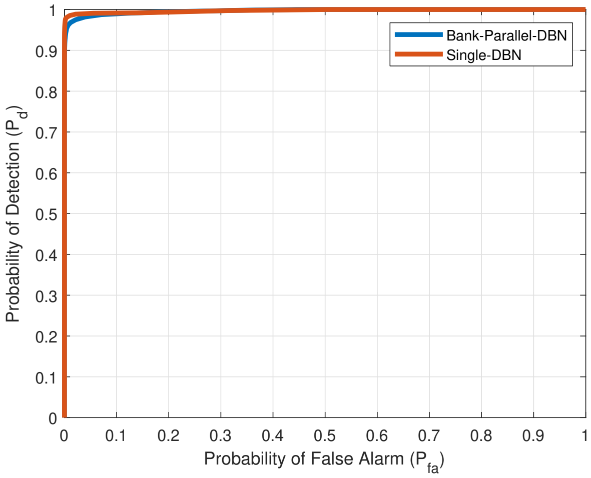


Figure 4.15: Performance comparison between Single-DBN and Bank-Parallel-DBN in terms of ROC.

4.7 Dynamic switching models in the context of incremental learning framework

To introduce AI capabilities into CR devices facilitating a higher SA level, an incremental learning process of dynamic switching models is presented that considers new experiences in the operating environment (i-e, abnormal signals, jammer signals, or change of transmitting scheme by the source). An incremental process allows the CR agent to learn a dynamic switching model from the stored data. The learned DBN model can predict and estimate situations that deviate from the previously known scenarios (yet learned models) and adopts to discover new experiences in an increment fashion. Such a procedure can be achieved through incrementally learning of switching dynamic models. A reference or initial model is esteemed that exhibits a dynamic equilibrium situation between an agent and the environment (CR and the spectrum), and increment learning is perceived as adding new knowledge to that standard reference model (initial model). Consequently, abnormality arises when there is a deviation from the dynamic equilibrium, and a new learned mode captures such an abnormal situation. An agent can exploit such a new scenario when experiencing changes in the reference model's dynamic equilibrium due to the addition of new learned models. Fig.4.16 depicts an overview of the incremental learning process. Such a scheme allows CR to learn models associated with new experiences incrementally. We can embed such knowledge in CR devices by using dynamic models to enhance intelligence, i-e, SA.

We can begin with a simple initial model $l_m = 0$, (where $l_m \in [0, 1, 2, ...L_M]$, and L_M) are total learned model), in which an agent remains immobile over time. Under this situation, an unmotivated KALMAN Filter (UKF) is deployed to track agent location. UKF is based on a random walk model such that [17]:

$$X_k = FX_{k-1} + w_k, \tag{4.42}$$

According to the equation, the agent remains in a quasi-static location, i-e $U_k \sim 0$. In this condition, only noise can affect an agent's state. It is a similar realization of null force filter (NFF). NFF [18] is an active filter that assumes a static agent behavior over time, leading $X_{k+1} = X_k$.

During the inference process, either a normal situation or an abnormal situation emerges. In the case of a typical problem, the model uses equation.4.30 to perform state estimation and there is no need to learn a new model as the situation is normal.

On the contrary, under an abnormal scenario, it is possible to learn a new model $m+1$ by utilizing anomalous data. The new model is learned using GS, which contains extracted abnormalities at time instant k and added to the learned model data-base. The overall process evolves with a time that facilitates introducing SA capabilities into the CR devices in a network and eventually incrementally detect abnormalities. We have described attributes of the SA module for CR devices in Table.4-C. Accordingly, it can be observed from the table that the SA module should be generative and discriminative, and the DBN model provides these two properties where generative includes learning of dynamic models based on generalized state, and the discriminating property involves

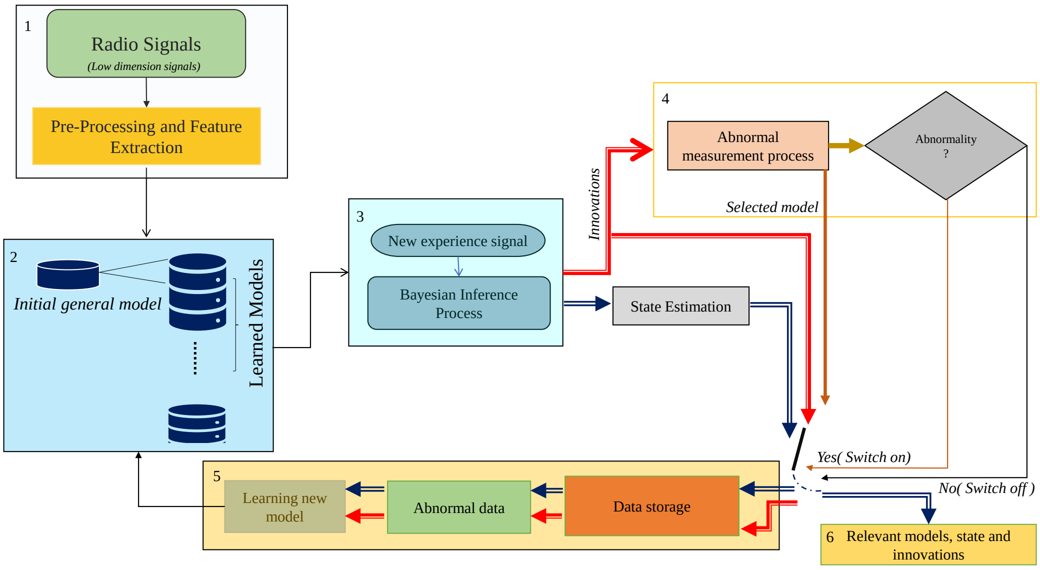


Figure 4.16: Incremental Learning Process when low dimensional radio signals are considered.

Table 4-C: Proposed SA model attributes

Self-Awareness Properties	Contribution
Generative	Learning models based on generalize state vector containing OFDM signal information
Discriminative	Abnormal signal measurements (Jammer Signals)
Interactive	Interaction between user and jammer signals present in a Networt
Hierarchical	Three levels DBN (Discrete state, Continuous state, Observation state)
Temporal reasoning	Future state prediction by using generalized state vector
Uncertain reasoning	DBN provides to make probabilistic reasoning

abnormality signal measurements. SA should provide a different level of inferences, and in this context, DBN gives inferences at three levels (continuous, discrete, and observation). The fundamental ability of SA is to predict the future state of an entity, and DBN hastens state estimation and indicates a future state of an object by using a probabilistic framework. Finally, DBN facilitates uncertain reasoning, and such ability is also a fundamental property of the SA model.

Undoubtedly, the proposed DBN expedites to acquaint SA abilities into CR-IoT devices, implemented in an increment fashion where the equilibrium between states is maintained, and some fluctuation arises, the abnormal situation is observed, and a new model is learned based on abnormal condition.

4.8 Discussion

In this chapter, we begin with the motivation to model CR behavior inside the spectrum as a dynamic process evolving with time. We present note-able state estimation methods such as PGM, BN, and DBN models, and highlight their role and impact in the signal processing domain. Followed by, jammer detection framework is presented for the

CR-IoT network spectrum. The proposed method uses DBN model comprised of discrete and continuous levels to perform inferences about spectrum evolution. In addition to this, Single and Bank-Parallel DBN systems have been introduced and discussed to detect jammer attacks inside OFDM modulated signal transmission.

The proposed DBN model captures abnormalities and can effectively handle low-dimension data (few subcarriers) of the OFDM signal transmission. Nevertheless, to deal with high-dimensional data (a higher number of subcarriers), we proceed in a direction to investigate deep generative models from the deep learning field. Consequently, deep learning models, specifically generative models based on probabilistic inference such as VAE, have been studied and analyzed to handle high dimensional data. In this context, **Chapter.5** highlights deep generative models and their application. **Chapter.6** presents a method based on the generative model and DBN that deal with high dimensional data efficiently and capture abnormalities in a CR-IoT spectrum. We also describe the realization of NFF in the deep learning model(VAE) by introducing an activation regularizer (AR). AR ensures learning of distinct latent spaces describing different situations. Such structure facilitates learning the model incrementally and detects abnormal scenarios.

4.9 Summary

Overall, this chapter presents a probabilistic framework to model CR behavior inside the CR-IoT network spectrum evolving with time. Popular probabilities models such as PGM, BN, DBN, HMM, and Kalman Filters are described. In this work, a switching DBN model is proposed, formulated, and implemented to capture jammer signals inside the CR-IoT spectrum based on generalized state vectors. Followed by this, Single and Bank-Parallel DBN models have been introduced and implemented for jammer detection in the OFDM modulated CR signal. The proposed method deals with low dimension data and detects jammer attacks. This chapter presents how DBN can facilitate SA ability by exhibiting the SA model's specific characteristics, implemented incrementally.

References

[1] A. Dore, A. F. Cattoni, and C. S. Regazzoni, "Interaction modeling and prediction in smart spaces: A bio-inspired approach based on autobiographical memory," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 6, pp. 1191–1205, 2010.

[2] A. Toma, C. Regazzoni, L. Marcenaro, and Y. Gao, *Learning Dynamic Jamming Models in Cognitive Radios*, 06 2018, pp. 1–37.

[3] S. C. Shapiro, *Artificial Intelligence (AI).* GBR: John Wiley and Sons Ltd., 2003, p. 89–93.

[4] L. E. Sucar, *Probabilistic Graphical Models: Principles and Applications.* Springer Publishing Company, Incorporated, 2015.

[5] F. Pernkopf, R. Peharz, and S. Tschiatschek, *Introduction to Probabilistic Graphical Models*, 12 2014, vol. 1, pp. 989–1064.

[6] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, 01 2009.

[7] D. Heckerman, "A tutorial on learning with bayesian networks," 2020.

[8] M. Ashcroft, "An introduction to bayesian networks in systems and control," in *18th International Conference on Automation and Computing (ICAC)*, 2012, pp. 1–6.

[9] V. Mihajlović and M. Petkovic, "Dynamic bayesian networks: A state of the art," *Europhysics Letters (epl)*, 01 2001.

[10] Y. Gu, J. Wang, and R. Han, "Dynamic bayesian network optimized by particle filtering in gene regulatory networks," in *2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, vol. 2, 2010, pp. 512–515.

[11] M. I. Mohd Yusoff, I. Mohamed, and M. R. Abu Bakar, "Hidden markov models: An insight," in *Proceedings of the 6th International Conference on Information Technology and Multimedia*, 2014, pp. 259–264.

[12] T. Bo, T. Xiaobin, and Y. Baoqun, "Continuous-time hidden markov models in network simulation," in *2008 IEEE International Symposium on Knowledge Acqui-*

sition and Modeling Workshop, 2008, pp. 667–670.

[13] P. J. Hargrave, "A tutorial introduction to kalman filtering," in *IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments*, 1989, pp. 1/1–1/6.

[14] G. A. Einicke and L. B. White, "Robust extended kalman filtering," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2596–2599, 1999.

[15] T. Kohonen, *Self-Organizing Maps, ser. Physics and astronomy online library.* Springer Berlin Heidelberg, 2001.

[16] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning Switching Models for Abnormality Detection for Autonomous Driving," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 2606–2613.

[17] R. Pelánek, T. Hanžl, I. Cerna, and L. Brim, "Enhancing random walk state space exploration," *FMICS'05 - Proceedings of the Tenth International Workshop on Formal Methods for Industrial Critical Systems*, 01 2005.

[18] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, "Static force field representation of environments based on agents' nonlinear motions," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, 01 2017.

Chapter 5

AI and Deep Generative Models for CR-IoT network

This chapter gives insight into the SA implementation in computing systems to develop an autonomous network. Various AI techniques for the CR network is described. Followed by this, deep learning models are presented along with the jammer signal classification framework. Moreover, the chapter also highlights popular generative models such as GAN, AE, and VAE.

5.1 Computing systems with Self-Awareness concept

Latest advancements in the field of Internet of Things (IoT), Wireless Sensor Network (WSN), Vehicle to Vehicle communication (V2V), UAV, and Software-Defined Networks (SDN) have made computing systems highly complex, heterogeneous, diverse, decentralized, and dynamic. Such a surge in the complexity of computing systems present obstacles in discerning and perceiving system behavior during run time and anticipate system response online. One of the emerging concepts to cope with this significant computing system issue is to incorporate self-awareness (SA) in a system to introduce autonomous characteristics. Such a facility allows the computing system to adjust its armature and tune parameters in the context of its functioning environment. In a nutshell, it enables systems to remain conscious of themselves. SA has drawn a lot of attention over the last couple of decades and has been penetrated in the field of computer vision and communication engineering. As a result, SA has become a fundamental ingredient of the Intelligent Computing and Communication System (ICCS). It can be defined as [1],

SA is a cognitive and intellectual characteristic of a human agent that observes and processes self-information.

In the research and scientific avenue, the term "Autonomic Computing" has been used to represent computing infrastructure that accustoms automatically to meet the application's requirements. Autonomic Computing has been inspired by the biological nervous system and was introduced in 2001 [2] to represent self-managing systems. Autonomic computing systems (ACS) can manage themselves without any human interaction to meet the objectives. Such systems can acclimate to changes triggered by a system's state or state of its operating environment, and are are self-configurable, self-managed, self-optimized, self-healing, and self-protected [3]. Fig.5.1 depicts basic ACS system which include automatic controller to adapt changes in a system during run-time [4]. SA has also been proposed and implemented in software engineering and in robotics [6].

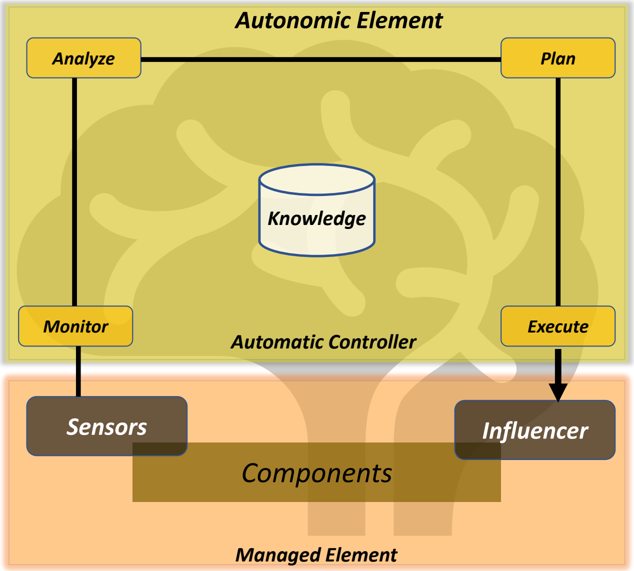


Figure 5.1: The basic architecture of Autonomic System [5].

Where in robotics, much of the work focuses on replicating forms of self-awareness as observed in humans. The other emanating applications such as cloud computing [7], interactive music devices [8], Internet of Things [9],[10], wireless sensor networks [11], automotive systems (self-driving cars) [12], mobile wireless network [13],and heterogeneous multi-core platforms on-the-fly computing [8], have employed SA functionalities to deal gracefully with the issues of run-time resource limitations.

Implementing SA in the emerging future 5G wireless network is quite apparent because of the resource constraints limitation, dynamic nature of the network, the high volume of data, and adversaries attacks in a network. Specifically, SA has been identified as a potential candidate in the CR-IoT network to deal with such challenges [14],[15] and [16]. For instance, devices in a network equipped with SA capabilities will perform self-monitoring in the operating environment to enhance spectral efficiency and adjust parameters to minimize interference with the adjacent devices. SA-enabled CR system can learn normal behavior inside the spectrum and detect any abnormal situation that deviates from normal operation. Fig 5.2 shows SA capabilities equipped in a cognitive device of CR-IoT network. In this context, refer **section.4.5** which describes method for low dimension data applications, and **section.6.2** discusses approach based on generative model and Bayesian network for high dimension data application.

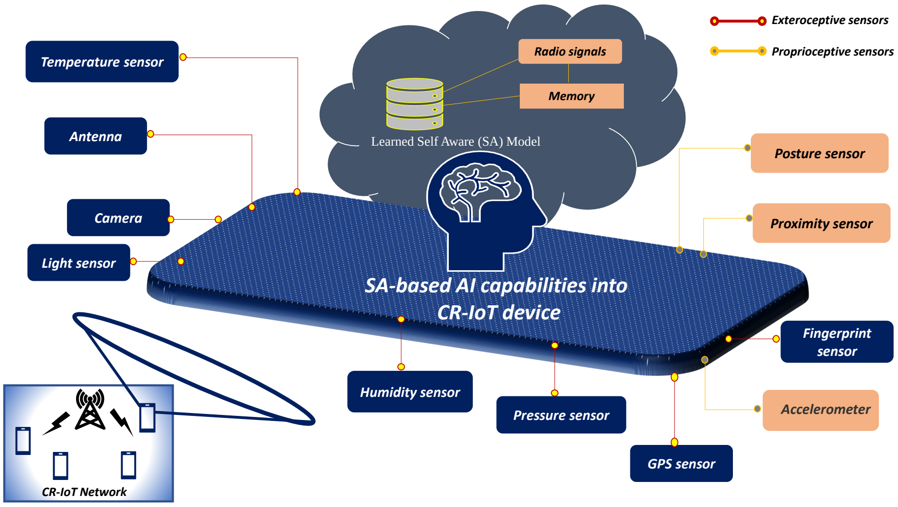


Figure 5.2: Bringing SA capabilities into cognitive mobile device in a CR-IoT network.

5.2 From Cognition to AI in CR-IoT network

The rapid proliferation and modern advancements in the Internet of Things (IoT) have driven towards the accessibility and provision of smart sensors, ubiquitous connectivity, and robust processing capabilities for many CR-IoT applications. Nevertheless, without the ubiety of cognitive capabilities, many significant IoT applications may endure limited or restricted. Consequently, it is imperative to equipped objects with cognitive capabilities, thereby proposing and presenting a CR-IoT network. CR was introduced in 1999 [17], and it is an emanating technology that has revolutionized the digital world. CR is an adaptive, self-aware radio and aware radio which can configure and change parameters in the pursuit of flexibility and adaptability according to the environment. It is equipped with dynamic spectrum access and self-configuring network capabilities. CR has been deployed in cognitive software defined networks [18], smart grid communication networks [19], intelligent cognitive vehicular communications and networks [20], IoT networks [21]. The basic CR characteristics are given as follows:

1) Awareness (Perceptions or Observations) It collects information about the radio environment and extracts parameters and their descriptions.

2) Adaptive (Reasoning or Reconfiguration) To optimize and improve network operation, it alters and modifies the radio parameters.

3) Cognition (Learning) It does the following:

a) Perceive and interpret the environment

b) Execute the decisions on the actions and eventually learn the consequence of such actions on the radio performance.

c) Assesses the performance of the network.

- Perceive and interpret the environment.
- Execute the decisions on the actions and eventually learn the consequence of such actions on the radio performance.
- Assesses the performance of the network.

In CR, Cognitive Engine (CE) integrates and incorporates the aforementioned attributes and is liable to execute cognitive tasks. CE is a cognitive envoy in a CR that inspect and evaluate the situation and find out the appropriate responses to the cognition and carry out decisions. Wireless networks have become more sophisticated, dense, and complicated because they offer various services (e-g text,images and videos) to users than ever before [22],[23]. Consequently, CR meets the following challenges in the emerging 5G and wireless networks:

1) CR devices will require time and cost (more hardware capacity) to learn the exhaustive and precise information about the radio environment of highly dense 5G and other wireless networks, consisting of several base stations, mobile devices, and other cognitive objects.

2) The nature of traffic volume in the 5G network is highly dynamic due to the users' demands for several services (e-g voice, audio, text, image, or videos). Such highly vigorous data brings a hurdle for CR to learn and predict accurately.

3) With the evolving concept of cloud computing and remote servers in various wireless networks, multi-dimensional resources (time, frequency, or spatial) at different layers (physical or network layers) coordinate to provide users' services. Hence, making the CR job more complicated and sophisticated.

As limitations mentioned above, there has been a magnificent consideration and substantial motivation to incorporate more robust and diverse capabilities into CR by introducing AI functionalities. Inspired by implementing AI techniques into computer vision and robotics fields, AI has become the best option for CR to achieve more assorted goals. AI brings a facility of self-awareness to the CR devices by enhancing the level of learning and reasoning. The main functions of AI in CR can be summarized as follows:

1) In a deep and divergent wireless network, it can learn a more rigorous environment pattern and execute appropriate actions yet with insufficient and inaccurate information of the environment.

2) Trace and track of CR devices in a network are accessible and possible because the AI agent can maintain a network's history and evolution profile. Therefore, it is easy to

track back the activities of devices in a system.
3) AI agent can evade complex mathematical formulation due to the efficient reasoning capability and hence, can instantly learn the impact of the action on the environment. Motivated by AI agent benefits, it has been realized and implemented into various filed of IoT networks. Following the discussion of SA into computing systems in this section, we now present various AI techniques for the CR-IoT network.

5.3 Various AI technique in CR-IoT Network

Various AI techniques have been proposed and implemented to achieve awareness, learning, and CR's reasoning functionalities. These AI techniques can be divided in to following categories according to the specific application and objectives.

5.3.1 Artificial Neural Network (ANN)

ANN is a parallel computing-structure comprised of non-linear functions with the tunable parameters to achieve a target output. ANN consists of several artificial neurons called *perceptrons*, developed in the early 1950s by the scientist Frank Rosenblat, inspired by the work of neuro-physiologist W.Mccollach in 1943. ANN consists of neurons interconnected and organized at different layers to perform specific tasks (such as prediction or classifications) in a network. The most popular ANN networks for CR-IoT networks are nonlinear perceptron networks (NPN) [24], radial basis functions network (RBFN) [25], and multi-layer perceptron networks (MLP) [26].
ANNs are robust models that learn network dynamics and extract patterns, features, and configurations from the objects. ANNs are highly adaptive and can be trained at any time. Due to such flexible nature of ANN, they have been deployed into various CR-IoT applications. For instance, [27], ANN model is used to hand off scheme in CR networks [27]. ANN has been deployed to accomplish spectrum sensing tasks for CR-IoT network [28]. Multi-dimensional spectrum sensing is presented for CR network using AI technique in [29]. Data-driven for quality improvements for spectrum sharing in CR-IoT

network [30].

5.3.2 Metaheuristic technique

A highly impenetrable and heterogeneous CR network contains search and optimization problems that involve complexities such as large dimensionality, non-linearity, group of mixed variables, and non-convexity. Therefore, classical algorithms based on a mathematical formulation can either become ineffective or inapplicable. To endure such problems, optimization algorithms have been proposed that can find an approximate solution while learning and establishing the relationship between various components in a CR network. Such algorithms are called *Metaheuristics* [31] introduced for the first time in 1986 [32]. Metaheuristics techniques (MT) don't explicitly provide the exact optimal solution of a problem; instead, it gives the near-optimal solution in a computationally efficient manner.

MT methods are implemented in the CR network, where finding an optimal solution with objective function aim to identify a set of rules that follow the training examples during learning. The overall objective is to find a hypothesis that maximizes the training example of the target concepts. MT methods such as genetic algorithm (GA), tabu searching and ant colony optimization (ACO) have been very famous in CR-IoT network to achieve several tasks. The work [33] describes spectrum allocation technique in CR network using genetic algorithm. ACO algorithm is implemented to achieve spectrum utilization and fairness in CR network [34], whereas, in [35] ACO is used to perform energy optimization in mobile IoT network. The paper [36] discusses task allocation strategies in CR-IoT network using ACO algorithm.

5.3.3 Rule-Based System

For a specific domain, rule-based systems (RBS) are implemented, which extract rules according to the applications and apply such rules to the decision-making process inside a system. RBS was introduced in early 1980 and comprised essentially two components, a rule base, an inference engine. The deployment of RBS in CR is quite tempting and obvious. Using RBS, a radio can instantly learn or deduce actions for a given input. The rule-base reasoning (RBR) system is designed and analyzed based on cognitive engine (CE) for wireless rural area network applications. RBR CE has been implemented and evaluated for cognitive radio networks [37].

5.3.4 Ontology-Based System

In the AI field, OBS has been used since the 1980s. OBS system provides format and explicit depiction of a set of conceptions in a domain [37]. It is used to reason about the attributes of the domain of interest. Following essential components constitute an OBS system 1) Classes 2) Instances 3) Attribute 4) Relations. To facilitate machine processing, ontology language has developed. In CR, a radio based on OBS can logically deduce facts to understand the characteristics of itself and other radios in a network [38].

5.3.5 Probabilistic Models

Probabilistic models comprise statistical networks and have been implemented to analyze the complex dynamic behavior of random processes [39]. PM contains observable and hidden states of an inevitable process to characterize the occurrence of the observable states. Such a model can be deployed to recognized observation with similar attributes by selecting a model that will likely produce the observed sequence. Hence, PM can be implemented as an observation model in the CR network to achieve awareness and recognized received sequences. The work [40] describes variational Bayesian inference method to achieve spectrum sensing in a cooperative manner for cognitive radio networks. Bayesian estimator is deployed to perform sensing operation in CR network [41]. Spectrum sensing in CR is accomplished using HMM model is presented in [42].

5.4 The principles of Deep Learning

Machine learning (ML) methods have reshaped human life and revolutionized many fields such as IoT, vehicular to vehicular communication, healthcare, transportation etc.

ML techniques are used for image classification, object detection, weather forecasting, abnormality detection from the spectrum of CR-IoT networks, time-series prediction, and many more. However, the ML technique meets certain limitations. For instance, precise features must be selected, extracted, and transformed into an appropriate representation vector from the raw data set. Such features are then used to train an ML model to perform either classification or prediction tasks depending on the applications [43]. On the contrary, deep learning methods have gained attention due to the ability to extract more complex and dense hidden features (spatial, temporal) from the raw data and possess powerful processing capabilities in generalizing the relationship of input data. Moreover, deep learning models exhibit good performance on large scale data while machine learning model may encounter over-fitting problem when dealing with huge amount of data. This is due to the deep and complex architecture of deep learning models.

Deep learning models are competent of erudition more complicated models and functions in the form of multiple-level representation structures composed of non-linear modules transforming the raw input from one level into another higher abstraction level. DL models predict and cluster objects/things using a neural network (NN) network comprised of many layers of neurons trained on a given data set. The intuition behind deep learning is an artificial neuron that mimics the human brain functionalities, as shown in Fig.5.3. DL network comprises several layers containing many artificial neurons, and the most common DL architectures consist of the following models: Multilayer Perceptrons (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Networks (RNN) [44]. In DL, feed-forward comprises several artificial neurons oriented across multiple layers inside a network that describes a process of mapping $f(x_0, \phi)$ $\mathbf{R}^{N0} \rightarrow \mathbf{R}^{NL}$ of input vector $x_0 \in \mathbf{R}^{N0}$ to an output vector $x_L \in \mathbf{R}^{NL}$ through L iterative processing steps.

$$x_i = f_i(x_{i-1}, \phi_i), i = 1, 2, 3..I \tag{5.1}$$

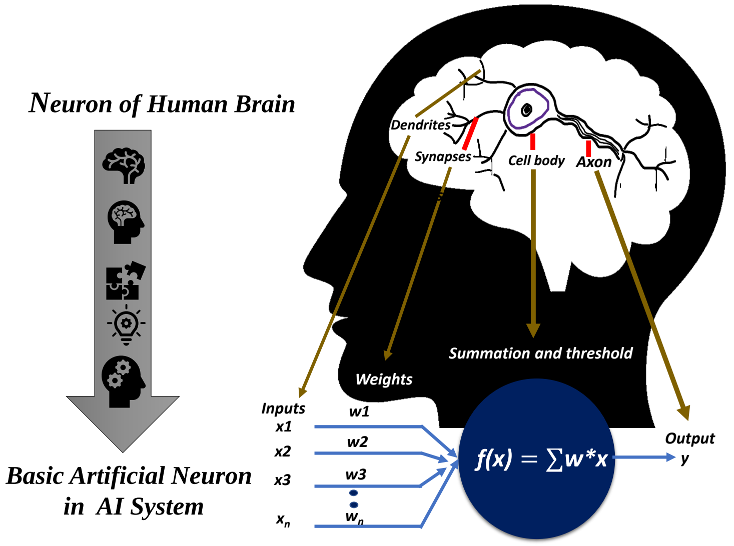


Figure 5.3: Human brain versus artificial neuron.

where, $x_i = f_i(x_{i-1}, \phi_i) : \mathbf{R}^{N_{i-1}} \rightarrow \mathbf{R}^{N_I}$ is essentially a mapping performed by the last I^{th} layer in the NN network. The f_i function may involve random variables, which makes mapping stochastic. The I^{th} is known as fully-connected or dense layer and it can be expressed as:

$$f_i(x_{i-1}, \phi_i) = \sigma(W_i x_i + b_i) \tag{5.2}$$

where, $b_i \in \mathbf{R}^{N_I}$, $W_i \in \mathbf{R}^{N_I} N_{I-1}$ and $\sigma(.)$ is an activation function. Weights and bias are the parameter of the fully connected layer. There are various kind of activation functions which can be used according to the given problem for a specific application. Such activation functions include: $Linear$, $ReLU$, Tan, $Sigmoid$ and $Softmax$ [45]. The training of NN network involves training set and label set which are deployed to train the model. The goal of training is to minimize cost function with respect to the parameter ϕ, which is expressed as,

$$L(\phi) = \frac{1}{M} \sum_{p=1}^{M} l(x^*_{I,P}, x_{I,P}) \tag{5.3}$$

Where, $L(\phi)$ is loss function and $x_{I,P}$ is an output of the NN.

5.4.1 Deep Learning for PHY-Layer CR-IoT

ML techniques have been implemented in communication systems to perform signal processing tasks at PHY-layer. Specifically, recent works also present many ML methods for the PHY-layer CR-IoT network. Concurrently, DL methods have also evolved as one of the potential techniques and have been deployed to attain cutting-edge results in various CR-IoT applications. Signal processing techniques at PHY-Layer for communication system possesses strong foundations in information theory and statistics. And such signal processing techniques use mathematical models that involve linearity and Gaussian statistics. However, many practical communication systems encounter non-linearities that arise due to devices in either transmitter or receiver. To address such challenges, it is imperative to deploy DL, which doesn't require an explicit mathematical model and can achieve the optimal solution of non-linear problems.

Moreover, due to the significant advancements in Graphics Processing Unit (GPUs), it is now possible to realized complex and dense NN network which are more energy efficient. NNs have demonstrated to achieve high efficiency in resource utilization using high-performance GPUs. Besides, DL methods are more robust in learning a full end-to-end communication system and can optimize the entire communication model for PHY-layer signal processing [46].

Various DL architectures such as CNNs, RNNs, and restricted Boltzmann machines have been implemented into multiple fields and obtained remarkable results. Specially, CNN network has gained significant attention in computer vision, image processing, and speech recognition. CCN networks have evolved from NN that perform convolution operations on a given input in any of its layers. CCN architecture essentially consists of several layers: pooling layer, convolution layer, and fully connected layer. The sparse connectivity and parameter sharing characteristics of the convolution layer have drastically improved the ML method's capabilities and performance. The kernel filter dimension is less than the input data dimension in sparse connectivity, which achieves sparsity connections between input and output data. Such compact kernel dimensions increase efficiency and lower memory demands. Parameter sharing makes use of the same parameters for several neurons inside a network. Fig.5.4 depicts CNN architecture comprises of several layers. Extensive learning of CNN can be helpful in training and testing. Every image inputted will go via a chain of convolution layers with filters (Kernals), pooling, and fully connected layers (FC) and eventually apply Soft-max function to measure an object ranging between 0 and 1 probabilistic values. The Fig.5.4 below depicts how CNN processes an input image and classifies the object based on its value.

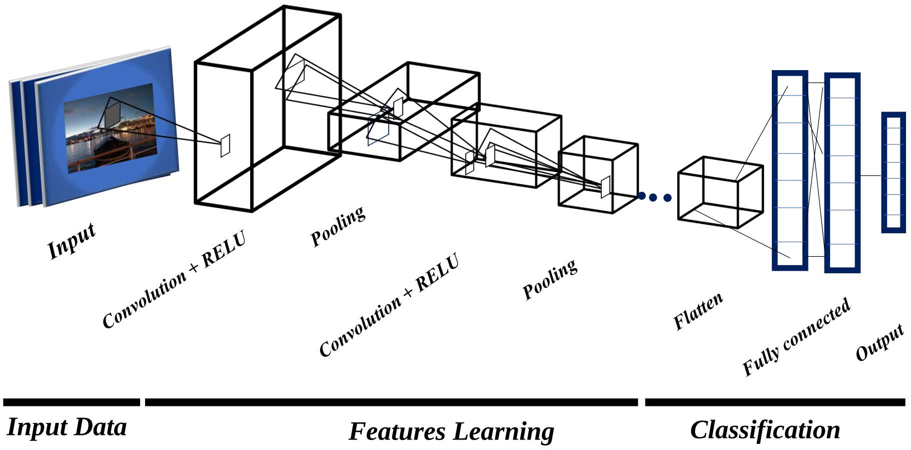


Figure 5.4: A simple CNN architecture consisting of several layers.

5.4.1.1 Automatic Jammer signal classification in the spectrum of CR-IoT network

Deep learning methods have recently gained attention for signal classification, automatic modulation recognition, and classifying normal and abnormal signals in the CR-IoT spectrum [47],[48]. In the radio domain covering a broad range of networks from Bluetooth to 5G networks, classification operations help infer signal identification, determine the modulation type of a received signal, and discover abnormalities in the spectrum. There has been a lot of work proposed and developed to perform radio signal classification by the researcher community. [49] presents end-to-end learning model for radio signal recognition. Signal identification for intelligent radios is demonstrated and discussed in [50]. Zheng et al.[51] describes signal classification method for cooperative radio classification. According to the recent advancements and developments in the signal classification domain, classification techniques are categorized into the following categories as follow:

1) Maximum likelihood estimation (MLE): MLE techniques exploit statistical knowledge and properties of the received signal to perform estimation based on Maximum-Likelihood (ML) [52]. ML methods are quite complicated, but their performance deteriorates in the dynamic radio environment [53].

2) Feature extraction based techniques: Feature extraction methods capture relevant and noteworthy features from the signals of the radio spectrum. Such attributes of the signal include (amplitude, phase, frequency, cyclo-stationary, and correlation coefficients). Such features are used to train the ML model for classification. Havryliuk et al. [54] presented a work which deploys wavelet transform and ANN classifier to track audio frequency in a network. In [55], cyclostationary features are used for detection and classification of OFDM signals. In [53] modulation techniques such as Binary Phase Shift Keying (BPSK), Quadrature Phase-Shift Keying (QPSK), Frequency-Shift Keying (FSK) and Minimum-Shift Keying (MSK) have been classified using S-transform based features. Also the comparison between different classifiers is presented under different range of Signal to Noise Ratio (SNR).

3) Deep learning models: The DL model's capability has also mesmerized the performance of signal classification output, and a lot of researchers have developed revolutionary methods that demonstrate promising results. The inherent delicacy of the DL method is its ability to learn automatically features from the data. Li et al.[56] demonstrated deep learning model for modulations identification. In [57], high order cumulants are used to learn deep learning model to perform modulation recognition. Tang et al.[58] describes the method which uses DL model (GAN) to automatically recognize modulation in cognitive radio network. [59] LSTM network is deployed as a deep learning model to perform wireless signals classification. Wang et al.[60] present data-driven approach which deploys CNN model for automatic modulation recognition in a CR. In [61] modulation classification is achieved using DL and signal constellation diagrams. Deep learning methods can be deployed to classify legitimate and jammer signals in the CR-IoT network spectrum.

In this context, we consider a CR-IoT network in which mobile devices are communicating with a base station (BS) using OFDM modulated signals. A smart jammer is also hypothesized to be present in a network which launches malicious attacks during the regular transmission. Such attacks disrupt normal communication and mislead the devices. Therefore, it is essential to perform classification operations to differentiate between normal and abnormal signals. Hence deep convolution neural networks (DCCN) have been used to accomplish classification tasks for the described scenario. In this work, two popular models (AlexNet and GoogLeNet) have been investigated and deployed.

Motivated by the promising results of DL methods, following work has been carried out:

1) Realize deep learning models **(AlexNet and GoogLeNet)** to classify normal user transmission and jammer attacks (with high and low power) inside the CR-IoT spectrum.

2) Collection of two features set of the OFDM modulated signal transmission. One set is obtained using continuous Wavelet Transform (CWT) in terms of scalograms representing spectrum contents in the form of an image. At the same time, the second set is obtained using Fast Fourier Transform (FFT), which gives complex data samples. These samples are then converted into the images to be used in the training of DL models.

3) Two models are trained based on two training sets and then tested. The performance is evaluated in terms of ROC curves.

CWT transforms OFDM signals into scalograms, which represent images encoding time and frequency information of the spectrum. Scalograms are obtained by taking absolute value of CWT signal coefficients. CWT filter bank is computed first to create the scalograms. Before acquiring the scalograms, the filter bank is deployed to obtain the CWT of the 1000 successive samples of the OFDM data. After that, scalograms are obtained from the coefficients. The generated scalogram are transformed into RGB images. Second data set is obtained using FFT transform. FFT operation provides complex samples of OFDM signals. Therefore, it is necessary to convert such data samples into RGB images. As mentioned, two deep learning models are used, which take images as an input. We now describe both model's configuration and their implementation setup. Fig.5.5 shows scalogram (left side) and FFT based images (right figure).

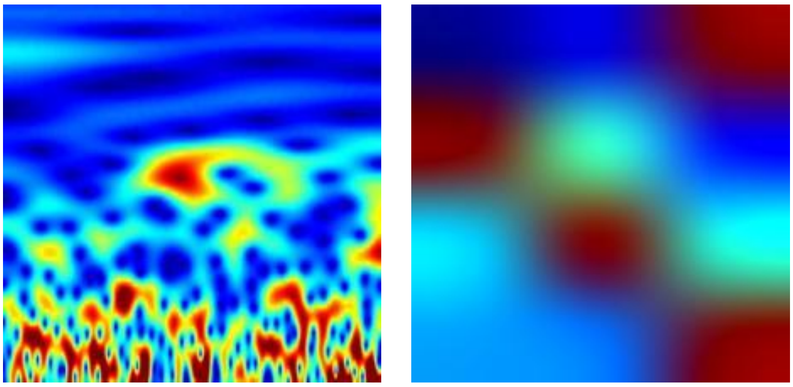


Figure 5.5: Scalogram (left image) and FFT images (right image).

- **AlexNet:** AlexNet played a significant role in making CNN popular in computer perception with its substantial and extensive architecture. The AlexNet structure contains eight layers. The initial five layers are convolutional, and the final three layers are fully-connected. Features are extracted by the first two convolutional layers that are connected to overlapping max-pooling layers. The 3^{rd} and 4^{th} convolutional layers are directly linked to the fully connected layers. ReLu activation function is applied to convolutional layers and fully connected layers. The final output layer uses the Soft-max function that produces 1000 different label classes [62]. AlexNet takes RGB images of size (256x256x3), where 256x256 shows image pixels and 3 is RGB channels. The overall architecture contains approximately 650,000 neurons along with other 60 million parameters. Dropout layers are also employed inside a network to avoid over-fitting during the training phase. Fig.5.6 depicts AlexNet architecture.

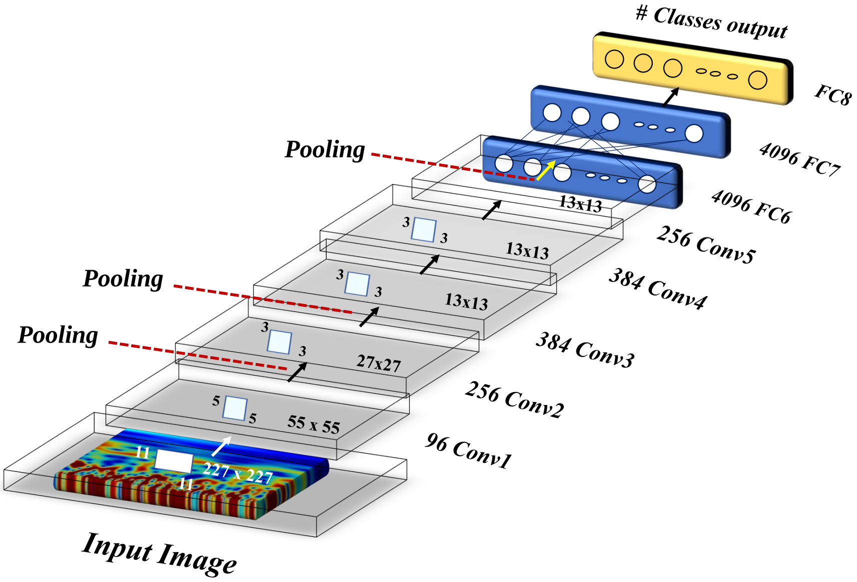


Figure 5.6: Alexnet network architecture.

- **GoogLeNet:** GoogLeNet has been evolved as a result of inception networks that are robust neural networks. Until now, three versions of inception networks or modules have been released. GoogLeNet is a robust model for detection and classification tasks. The overall structure is oriented as 22 deep layers and 27 pooling layers along with inception modules. The pooling layer is connected to each inception module at the end. There are a total of 7 million parameters in the GoogLeNet structure. It takes an input of size (227x227x3) [63]. Fig.5.7 describes the architecture of inception model of GoogLeNet.

5.4.1.2 Implementation

The implementation phase consists of training and testing phases, as shown in Fig.5.8. Parameters of both models are modified according to the given scenario and problem in this work. The models are capable of classifying spectrum signals into the following classes: reactive jammer with high power ($RJHP$), reactive jammer with low power ($RJLP$), and normal signal spectrum (NSS). Accordingly, for AlexNet, the number of the output layer is taken three instead of the default number, which is 1000. The other

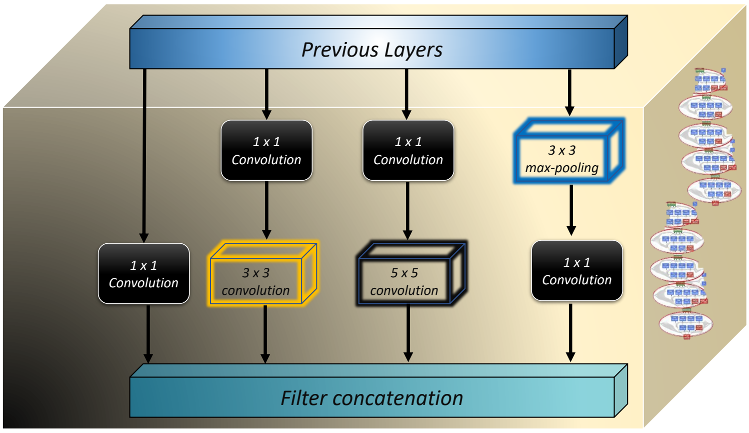


Figure 5.7: Inception model of GoogLeNet.

configuration parameters are set as, mini-batch size 64 with learning rate 0.0001, and 227 x 227 image resolution is taken for AlexNet. Adaptive Moment Estimation (ADAM) is used as a learning method for both models as it combines the benefits of RMSPro and momentum method and achieves high classification accuracy in comparison with Stochastic Gradient Descent Method (SGDM) and RMsProp. For GoogLeNet, input image resolution is 224 x 224, mini-batch size 64 with learning rate 0.0001 are selected The training of both models follow the given steps: **1)** Obtain the time-frequency representation of the OFDM modulated signals using CWT and FFT technique. And, convert the generated data into RGB images. **2)** Each generated image is labeled according to the classes. **3)** For training, 10000 images per signal category are collected, and for testing, 1000 labeled images per signal category. The training is done using the computing system NVIDIA GEFORCE CPU CORE i7. After the models have been trained, a test set is applied to infer the performance. The performance of models is analyzed in terms of the ROC curves.

5.4.1.3 Performance Evaluation

To test the performance of both models (AlexNet and GoogLeNet), two data sets, namely, FFT-based images, dataset, and CWT-based images dataset, are utilized to conduct the experiments. Both models' performance on two datasets is evaluated using

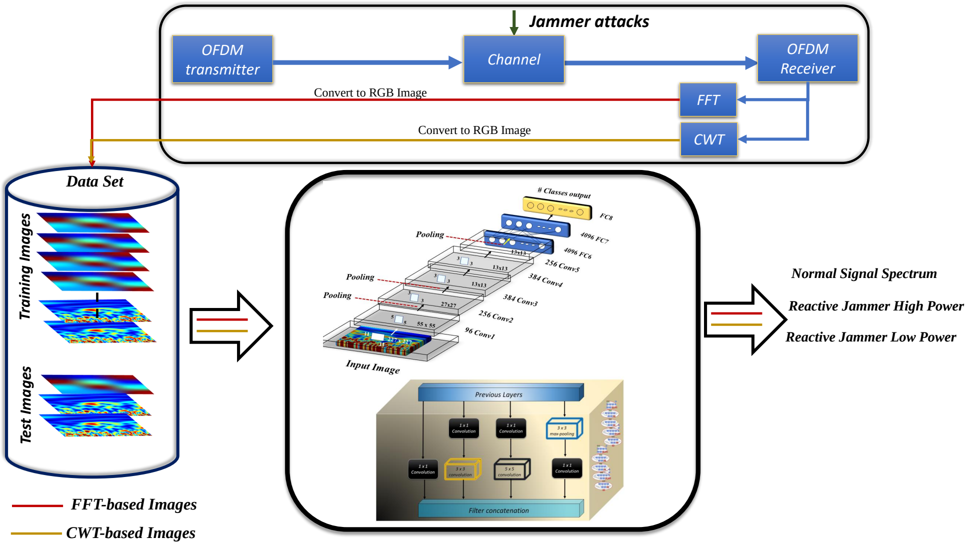
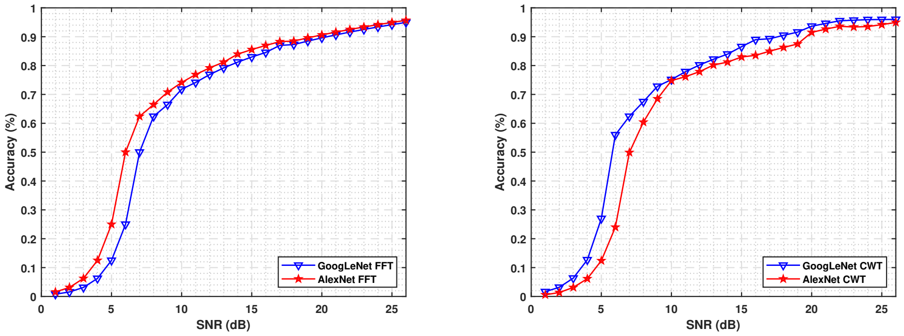


Figure 5.8: Implemented scenario of deep learning models using FFT and CWT based images to classify jammer signals.

a classification accuracy plotted against various SNR values as shown in 5.9(a) for FFT-based image dataset and 5.9(b) CWT-based image dataset. It can be observed from 5.9(a), the classification accuracy of AlexNet model slightly better than the GoogLeNet for FFT-based image dataset. On the other hand, for CWT-based image dataset, GoogLeNet classification accuracy is more nuanced than the AlexNet, as depicted in Fig.5.9(b). Fig.5.10 presents comparison analysis between deployed models and simple CNN model (trained on both datasets). It can be observed that both DL models outperform the simple CNN model. Classification accuracy comparison is plotted at various SNR values of all three models on both data sets. It can be analyzed that at 20dB SNR, AlexNet performance is better than the GoogLeNet on FFT-dataset, and GoogLeNet classifies better than the AlexNet on CWT-dataset. However, a simple CNN doesn't perform well on both datasets. Hence, we can deduce the following conclusion from the implemented method:

Classification of abnormal signals (jammer attacks) can be achieved using either of the deep learning models. However, the input data's pre-processing technique for training the DL model is crucial and must be selected according to the application. As presented in this work, there is a subtle difference in model performance on both datasets.

Therefore, it is exceptionally motivating to use the FFT-based images dataset. This is because most recent wireless technologies, such as WiFi, LTE, 5G, and CR-IoT networks, etc., deploy OFDM as a potential modulation technique due to its various advantages. Moreover, the OFDM communication system contains a built-in FFT-module. Thus, it is easy to extract FFT information in the OFDM system receiver without deploying any other transformation technique. Such information can then be converted into RGB images and use to train the DL model as presented in this work.



(a) (b)

Figure 5.9: Accuracy plots a) ALexNet and GoogLeNet models performance on FFT-based images b) ALexNet and GoogLeNet models performance on CWT-based images.

5.5 Deep Generator Model

CR-IoT network deploying data-driven SA capabilities based on AI techniques will achieve next-generation wireless networks requirements and eradicate prevailing issues of the current wireless network (such as spectrum access and utilization challenges). These wireless networks generate, exchange, and communicate diverse data (text, images, and videos) in a massive quantity; therefore, it is essential to learn and discover the data transmission pattern to develop autonomous and intelligent systems. In this context, discriminative models have been very famous and made remarkable developments in computer vision and smart systems domains. Discriminative models (DM) facilitate to

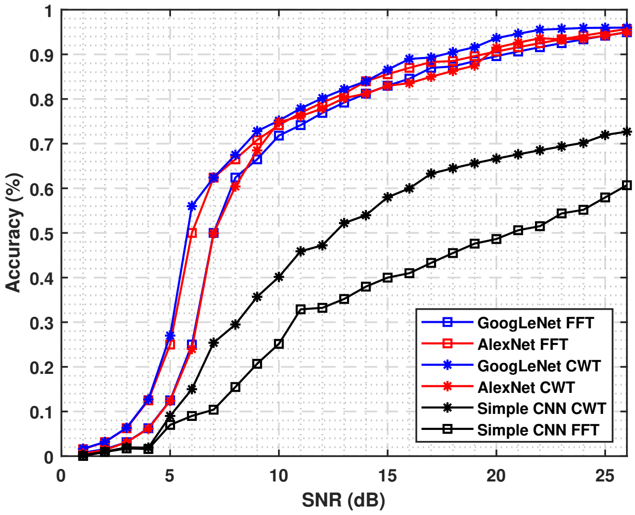


Figure 5.10: Performance comparision of AlexNet and GoogLeNet with conventional CCN model on FFT and CWT based images.

capture and discover hidden structure from the trove of data in an unsupervised fashion. However, DM becomes limited in a machine learning problem where direct learning a target is intractable. Moreover, DM models are not efficient in predicting out-of-samples. For a data-driven AI models with such an enormous amount of data in a network, the objective is to develop algorithms and build models to investigate and infer data in an unsupervised manner. That brings the need to implement a new class of models known as Generative Models (GM) [64]. GM models have been around for a long time and recently gained attention in data-driven SA based on AI applications. GM apprehends probability distribution from the input data samples and then samples from that probability distribution to generate output or target samples [65]. Such a generated output samples follow the input data samples closely. To summarize, it does:

Generated sample $p_{model}(\hat{x})$ wants to be similar to training data $p_{model}(\hat{x})$

DGM models have been implemented into robotics, speech recognition, and computer vision. GMs are good at exhibiting the following functionalities [64]:

1) Producing artificial yet realistic images.

2) Generating contents with predefined sentences and words.

3) Predicting and completing the missing or incomplete segments of data.

4) Capable of working with multi-modal targets or outputs.

5) GMs can manipulate original images based on specific features. They can change the image events (image-to-image translation).

GMs have been classified into two classes:

A) Energy-based models: In an energy-based model, the energy function is deployed to define the joint probability functions. Boltzmann machines and deep belief networks are examples of energy-based models.

B) Cost function model: In this model, the cost function is used to define loss between input and generated samples. It includes models such as GANs and AE.

The most popular deep GM models are now discussed.

5.5.1 Auto-encoder (AE)

AE is an outstanding GM class based on neural networks (NN) architecture that learns a concise and compact representation from high dimensional data and transforms it into low-dimensional data. AE encodes high dimension data into a low-dimension vector and decodes the data from that low dimension vector into a high dimension. AE is constructed and designed to learn data by extracting inherited regularities of a given input sample. AE was proposed in the early 1980s [66] and became famous in recent decade. The architecture of AE comprises of two networks [67]: Encoder and decoder, as shown in Fig.5.11.

Encoder: It is a multi-layer NN architecture that transforms input data into the low-dimension vector z. The low-dimension vector z is called a latent vector or bottleneck layer.

Decoder: It is consisting of NN layers that essentially perform the opposite job of the encoder, i-e, reconstructs output from the bottleneck layer.

The AE model is described by the encoder function $g(.)$ and parameter ϕ and decoder function $f(.)$ and parameter θ. The latent space learns $z = g_\phi(x)$ from input data and reconstructs $\hat{x} = f_\theta(g_\phi(x))$. The loss or object function of AE is the difference between

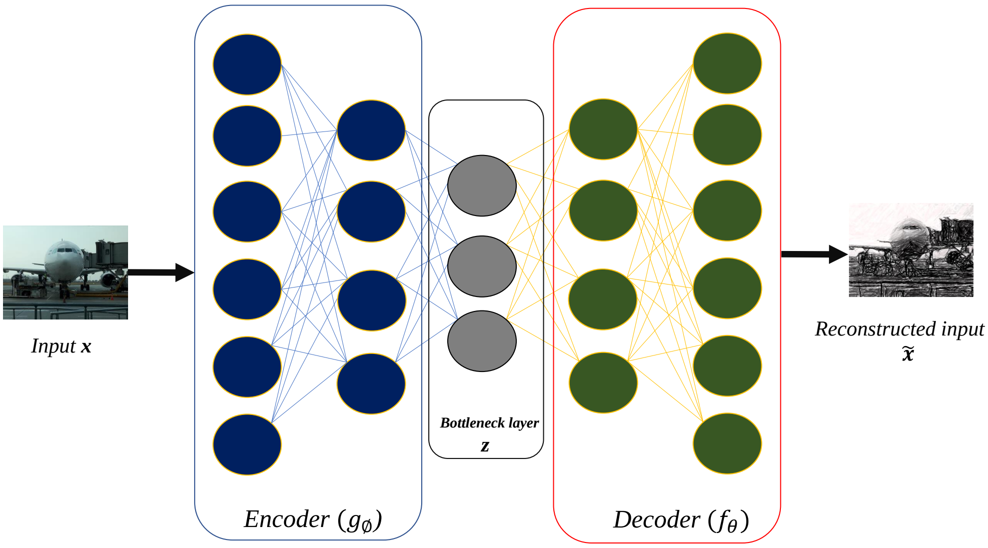


Figure 5.11: Autoencoder architecture comprise of encoder and decoder.

original and reconstructed output and is given as [68],

$$L_{AE}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(x^n - f_\theta(g_\phi(x^n)))^2 \tag{5.4}$$

There are different types of AE present in the literature, such as sparse AE, denoising AE, stacked AE, and variational AE. We shall be specifically describing VAE in the next section.

5.5.2 Variational Auto-encoder (VAE)

VAEs are GMs that have emerged as one of the powerful and compelling techniques to unsupervised learning of complex data distribution. VAEs are the type of AE based on Bayesian and variational inference methods. The intuition behind VAE is that it learns distribution in the form of z from the input data x, unlike traditional AE in which input data is transformed into a smaller latent vector, and it reconstructs output data $\hat{x}$ by sampling from the latent space distribution of z. VAE has been implemented in computer vision, speech processing, robotics, and IoT.

VAE comprises of encoder and decoder network as shown in Fig.5.12. W, W',b and b'

represents weight matrices and bias vectors of encoder and decoder respectively. Additionally, $\phi = (W, b)$ and $\theta = (W', b')$ are the training parameters use to train encoder and decoder of VAE respectively. The encoder parameter q_ϕ maps input data x^k to the latent vector z^k which represents data in a more compact form. The decoder p_θ attempts to project back the latent vector z^k to the input space and produce reconstructed data $\hat{x^k}$. Therefore,

$$z^k = q_\phi(z^k) = f(Wx^k + b), \tag{5.5}$$

$$\hat{x^k} = p_\theta(x^k) = g(W'z^k + b'), \tag{5.6}$$

where, $f(.)$ and $g(.)$ show activation functions of encoder and decoder respectively. We can define the relationship between input data x and the latent space z as **1) Prior** $p_\theta(z)$, **2) Likelihood** $p_\theta(x|z)$, **3) Posterior** $p_\theta(z|x)$. To reconstruct or generate a sample $\hat{x}$ that approximate or closely follow input data x. We follow the following steps by assuming that the real parameter θ^* for the distribution is known.

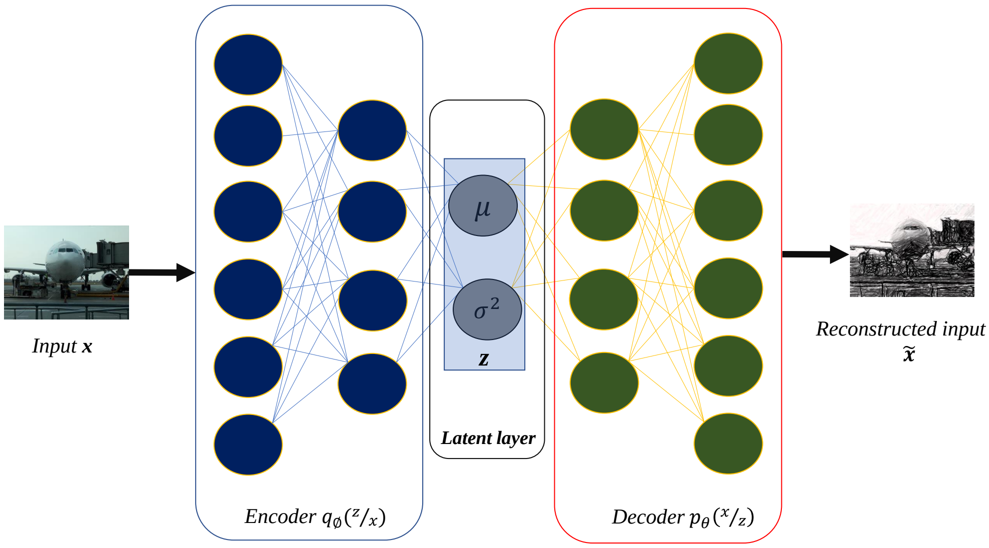


Figure 5.12: VAE architecture comprise of encoder and decoder.

1) From a prior distribution $p^*_\theta(z)$, sample z^i.

2) After that, x^i is produced from a distribution $p^*_\theta(x|z = z^i)$.

Parameter θ^* maximizes the probability of generating real data sample that is expressed as,

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1}^{n} logp_\theta(x^i) \tag{5.7}$$

And, data generation process which involve encoding vector,

$$p_\theta(x^i) = \int_\theta p_\theta(x^i|z)p_\theta(z)dz \tag{5.8}$$

Computing $p_\theta^*(x^i)$ is not traceable. Therefore, to narrow down value space to facilitate faster search, a new approximate function $p_\theta(z|x)$ in output is introduced parameterized by ϕ. Hence in VAE, $p_\theta(x|z)$ defines a probabilistic decoder of VAE, and $q_\phi(z|x)$ defines the encoder of VAE. Fig.5.13 show the graphical representation of VAE model.

The VAE loss is called as evidence lower bound (ELBO) loss which contains two part,

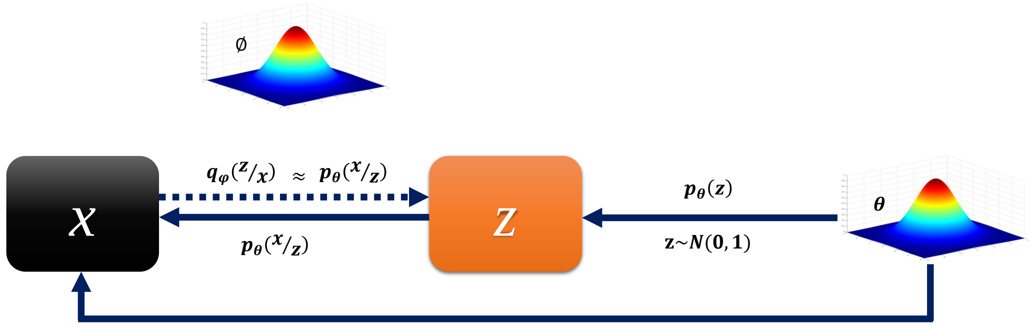


Figure 5.13: VAE graphical model.

i-e, reconstruction loss which determines similarity between decoder output and input, and Kullback–Leibler (KL) loss which defines the difference between two probability distributions. Moreover, K.L ensures that learned μ and σ are closed to the target distribution. The VAE loss is expressed as [69],

$$Loss_{VAE} = E_q(z/x^i)[Log_{p\theta}(x^i/z)] - D_{KL}(q_\phi(z/x^i)\|p_\theta(z)) \tag{5.9}$$

5.5.3 Generative Adversarial Network (GAN)

GANs have brought a state-of-the-art revolution to the deep generative models. GANs are the popular class of GM and have proven potential methods in many fields due to its remarkable data generation competence. GAN takes the advantages of unsupervised machine learning that automatically devise and learn the structure in input data so that model can be used to generate output probably could have been drawn from the original data set. Theoretically, GAN deploys a supervised learning approach to accomplish unsupervised learning by producing synthetic data. GANs are very effective in training a GM by considering a given problem with two sub-networks: Generator $G_{generator}$ and discriminator network by D [70].

The two models are trained concurrently and establish a min-max game between generator $G_{generator}$ and discriminator $D_{discriminator}$. The $G_{generator}$ attempts to deceive the $D_{discriminator}$ by providing real-world images close to the original from random noise vector z, whereas $D_{discriminator}$, gets better in distinguishing between real and synthetic data. Both networks amend themselves in the best feasible way to obtain targets. The GAN architecture is shown in Fig.5.14 [70]. The GAN model $G_{generator}$ takes noise vector z as an input defined by prior probability p_z and then train the model to learn

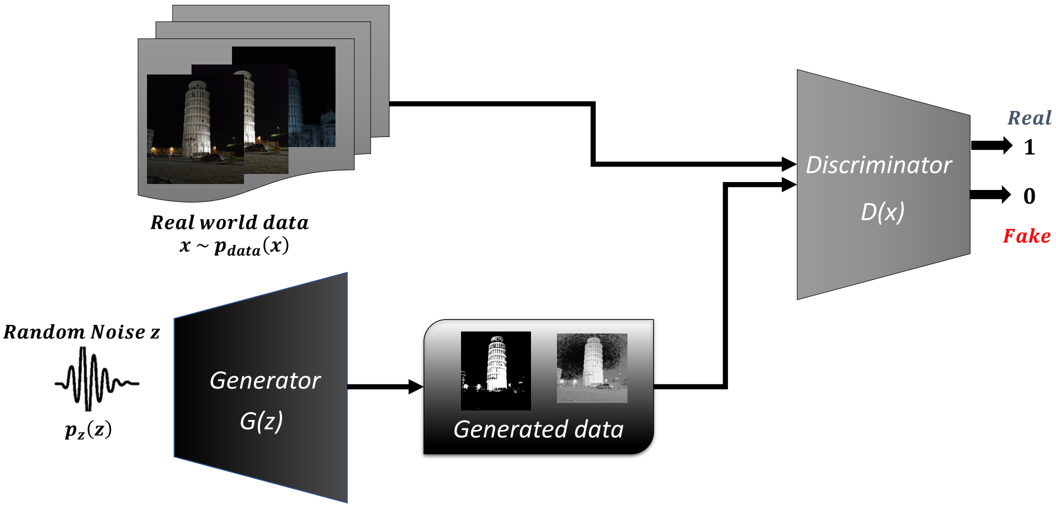


Figure 5.14: GAN model.

the distribution of generator p_g, by projecting back $G(z, \theta_g)$ from z to data space. The $D_{discriminator}$ takes an input image $\widetilde{X}$, then determines a projecting back $D(\widetilde{X}, \phi_d$ from $\widetilde{X})$ to a single scalar that is probability of the image $\widetilde{X}$,p_{data}. $D_{discriminator}$ output near to 1 if $\widetilde{X}$ is a real image from real data set p_{data}, whereas, it gives 0 means $\widetilde{X}$ is from p_g. $D_{discriminator}$ network aims to maximize its cost function and $G_{generator}$ attempts to minimize its cost value. The total object function is given as,

$$Loss_{GAN} = E_{p_{data}}[Log(D_{dis}(\widetilde{X})] + E_{z\ p_z(z)}[log(1 - D_{dis}(G_{gen}(z))] \tag{5.10}$$

5.5.4 Energy-based GM models

Energy-based GMs have been proposed and studied texture synthesis tasks and handwritten digits classifications [64]. Boltzmann Machines are such models that are based on energy functions, introduced in 1983 by Geoffrey Hinton et.al [71]. The energy function of a model describes the composition and configuration of the input data variables and provides a scalar value demonstrating the awfulness status of such configuration or composition. Therefore, the energy function corresponds to lower values to the correct configurations and high values for the incorrect one. Predictions are made by selecting the configurations that exhibit min values of energy.

The restricted Boltzmann machine based on binary Boltzmann machine is the foundation of recent generation powerful deep generative models, and the most popular models under this category are Deep Belief Networks (DBN) and Deep Boltzmann machine.

DBM comprises of a multi-layer network which is undirected. In this model, every unit is associated to every other unit in a multi-layer architecture. DBM models are trained in two steps: Pre-step in which each DBM is trained independently, and then a fine-tuning step in which the whole network is trained using back propagation. DBN networks are introduced in 2006 and are more complex and dense structure consisting of multi hidden layers. DBN is trained using a popular greedy-layer wise fast algorithm.

5.6 Summary

In this chapter, we briefly describe SA requirements in computing systems and the need to bring intelligence to CR devices. Various AI techniques have been presented for the CR-IoT network. We discuss deep learning principles and the motivation to use deep learning models to perform signal processing tasks. Jammer signal classification by using AlexNet and GoogLeNet models based on FFT and CWT based images have been proposed and presented. In this chapter, we also discuss popular generative models such as GAN, AE, and VAE.

References

[1] A. Morin, “Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views,” *Consciousness and cognition*, vol. 15, pp. 358–71, 07 2006.

[2] A. Ganek, “Autonomic computing: implementing the vision,” in *2003 Autonomic Computing Workshop*, 2003, pp. 1–1.

[3] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[4] P. R. Lewis, “Self-aware computing systems: From psychology to engineering,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, 2017, pp. 1044–1049.

[5] F. Krikava, “Domain-specific modeling language for self-adaptive software system architectures,” Ph.D. dissertation, 11 2013.

[6] M. Mathew, R. Sapra, and S. Majumder, “A learning based approach to self modeling robots,” in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 758–762.

[7] T. Chen and R. Bahsoon, “Toward a smarter cloud: Self-aware autoscaling of cloud configurations and resources,” *Computer*, vol. 48, no. 9, pp. 93–96, 2015.

[8] K. Nymoen, A. Chandra, and J. Torresen, *Self-awareness in Active Music Systems*, 07 2016, pp. 279–296.

[9] L. Esterle and B. Rinner, “An architecture for self -aware iot applications,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6588–6592.

[10] M. Möstl, J. Schlatow, R. Ernst, H. Hoffmann, A. Merchant, and A. Shraer, “Self-aware systems for the internet-of-things,” in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2016, pp. 1–9.

[11] B. Rinner, L. Esterle, J. Simonjan, G. Nebehay, R. Pflugfelder, G. Fernández Domínguez, and P. R. Lewis, “Self-aware and self-expressive camera networks,” *Computer*, vol. 48, no. 7, pp. 21–28, 2015.

[12] J. Schlatow, M. Moostl, R. Ernst, M. Nolte, I. Jatzkowski, M. Maurer, C. Herber, and A. Herkersdorf, “Self-awareness in autonomous automotive systems,” in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, 2017, pp. 1050–1055.

[13] M. Mordacchini, A. Passarella, M. J. Chorley, G. B. Colombo, and V. Tanasescu, “Making mobile users’ devices aware of the surrounding physical environment: An approach based on cognitive heuristics,” in *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, 2013, pp. 199–208.

[14] P. Sutton, L. E. Doyle, and K. E. Nolan, “A reconfigurable platform for cognitive networks,” in *2006 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, 2006, pp. 1–5.

[15] Y. Ding, J. Wu, H. Zhou, P. Feng, B. Liu, and L. Gui, “A self-awareness routing scheme with power control for underlay spectrum sharing networks,” in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2011, pp. 1–5.

[16] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni, “Ai-based abnormality detection at the phy-layer of cognitive radio by learning generative models,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 21–34, 2020.

[17] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[18] A. Zubow, M. Döring, M. Chwalisz, and A. Wolisz, "A sdn approach to spectrum brokerage in infrastructure-based cognitive radio networks," in *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2015, pp. 375–384.

[19] M. W. Khan, M. Zeeshan, and K. Shahzad, "On performance analysis of ieee 802.22 phy for cognitive radio based smart grid communications," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–4.

[20] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, "Internet of things meets brain–computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2084–2092, 2019.

[21] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[22] K. David and H. Berndt, "6g vision and requirements: Is there any need for beyond 5g?" *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, 2018.

[23] M. Patzold, "5g is coming around the corner [mobile radio]," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 4–10, 2019.

[24] H. Hu, Y. Wang, and J. Song, "Signal classification based on spectral correlation analysis and svm in cognitive radio," in *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*, 2008, pp. 883–887.

[25] S. Zhang, J. Hu, Z. Bao, and J. Wu, "Prediction of spectrum based on improved rbf neural network in cognitive radio," in *2013 International Conference on Wireless Information Networks and Systems (WINSYS)*, 2013, pp. 1–5.

[26] M. Xue, H. Wu, and Y. Zeng, "Multilayer perceptron for modulation recognition cognitive radio system," in *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2016, pp. 1–5.

[27] V. Agarwal, A. Kumar, D. Kumar, P. Tripathi, V. Rajpoot, and V. S. Tripathi, "A novel ann based efficient proactive handoff scheme for cognitive radio network," in *2019 International Conference on Computing, Power and Communication Tech-*

nologies (GUCON), 2019, pp. 400–404.

[28] Z. Wu, M. Luo, Z. Yin, and Y. Zhao, "Research of spectrum sensing based on ann algorithm," in *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control*, 2014, pp. 493–496.

[29] K. C. Sriharipriya and R. Sanju, "Artifical neural network based multi dimensional spectrum sensing in full duplex cognitive radio networks," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 307–312.

[30] X. Li, H. Ding, M. Pan, J. Wang, H. Zhang, and Y. Fang, "Statistical qos provisioning over uncertain shared spectrums in cognitive iot networks: A distributionally robust data-driven approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12 286–12 300, 2019.

[31] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, pp. 268–308, 01 2001.

[32] F. W. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, pp. 533–549, 1986.

[33] Y. El Morabit, F. Mrabti, and E. H. Abarkan, "Spectrum allocation using genetic algorithm in cognitive radio networks," in *2015 Third International Workshop on RFID And Adaptive Wireless Sensor Networks (RAWSN)*, 2015, pp. 90–93.

[34] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled internet of things: Network architecture and spectrum access," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[35] H. Zhao, J. Wang, X. Guan, Z. Wang, Y. He, and H. Xie, "Ant colony based energy consumption optimization for mobile iot networks," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2019, pp. 118–122.

[36] A. Zannou, A. Boulaaam, and E. H. Nfaoui, "A task allocation in iot using ant colony optimization," in *2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)*, 2019, pp. 1–6.

[37] A. He, K. K. Bae, T. R. Newman, J. Gaeddert, K. Kim, R. Menon, L. Morales-Tirado, J. . Neel, Y. Zhao, J. H. Reed, and W. H. Tranter, "A survey of artificial intelligence for cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1578–1592, 2010.

[38] M. M. Kokar and L. Lechowicz, "Language issues for cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 4, pp. 689–707, 2009.

[39] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[40] M. Wu, T. Song, L. Shen, and Z. Jia, "Variational bayesian inference based cooperative spectrum sensing in cognitive radio networks," in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 2014, pp. 108–109.

[41] C. H. A. Tavares and T. Abrão, "Bayesian estimators for cooperative spectrum sensing in cognitive radio networks," in *2017 IEEE URUCON*, 2017, pp. 1–4.

[42] M. S. El Din, M. El-Tarhuni, K. Assaleh, and S. Kiranyaz, "An hmm-based spectrum access algorithm for cognitive radio systems," in *2015 International Conference on Information and Communication Technology Research (ICTRC)*, 2015, pp. 116–119.

[43] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, and J. Liu, "A survey on deep learning empowered iot applications," *IEEE Access*, vol. 7, pp. 181 721–181 732, 2019.

[44] S. S. Alahmari, D. B. Goldgof, P. R. Mouton, and L. O. Hall, "Challenges for the repeatability of deep learning models," *IEEE Access*, vol. 8, pp. 211 860–211 868, 2020.

[45] A. Asperti, "Sparsity in variational autoencoders," *CoRR*, vol. abs/1812.07238, 2018. [Online]. Available: http://arxiv.org/abs/1812.07238

[46] L. Zhang, J. Tan, Y. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3281–3294, June 2019.

[47] H. Nishizaki and K. Makino, "Signal classification using deep learning," in *2019*

IEEE International Conference on Sensors and Nanotechnology, 2019, pp. 1–4.

[48] K. Bu, Y. He, X. Jing, and J. Han, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Processing Letters*, vol. 27, pp. 880–884, 2020.

[49] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.

[50] O. A. Dobre, "Signal identification for emerging intelligent radios: classical problems and new challenges," *IEEE Instrumentation Measurement Magazine*, vol. 18, no. 2, pp. 11–18, 2015.

[51] S. Zheng, S. Chen, and X. Yang, "Deep learning for cooperative radio signal classification," 2019.

[52] D. Boiteau and C. Le Martret, "A general maximum likelihood framework for modulation classification," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 4, 1998, pp. 2165–2168 vol.4.

[53] U. Satija, M. Mohanty, and B. Ramkumar, "Automatic modulation classification using s-transform based features," in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2015, pp. 708–712.

[54] V. Havryliuk, "Audio frequency track circuits monitoring based on wavelet transform and artificial neural network classifier," in *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2019, pp. 491–496.

[55] S. Vukotić and D. Vućić, "Detection and clasiffication of ofdm/qam and ofdm/oqam signals based on cyclostationary features," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, 2015, pp. 232–235.

[56] J. Li, L. Qi, and Y. Lin, "Research on modulation identification of digital signals based on deep learning," in *2016 IEEE International Conference on Electronic Information and Communication Technology (ICEICT)*, 2016, pp. 402–405.

[57] W. Xie, S. Hu, C. Yu, P. Zhu, X. Peng, and J. Ouyang, "Deep learning in digital modulation recognition using high order cumulants," *IEEE Access*, vol. 7, pp.

63 760–63 766, 2019.

[58] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, “Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks,” *IEEE Access*, vol. 6, pp. 15 713–15 722, 2018.

[59] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, “Deep learning models for wireless signal classification with distributed low-cost spectrum sensors,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

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

[61] W. P. Peng and Z. H. Ye, “Ofdm blending modulation systems in practical cognitive radio application,” in *2011 4th International Congress on Image and Signal Processing*, vol. 4, 2011, pp. 2237–2240.

[62] F. R. Mashrur, A. Dutta Roy, and D. K. Saha, “Automatic identification of arrhythmia from ecg using alexnet convolutional neural network,” in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 2019, pp. 1–5.

[63] P. Salavati and H. M. Mohammadi, “Obstacle detection using googlenet,” in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018, pp. 326–332.

[64] A. Oussidi and A. Elhassouny, “Deep generative models: Survey,” in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2018, pp. 1–8.

[65] C. G. Turhan and H. S. Bilge, “Recent trends in deep generative models: a review,” in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, 2018, pp. 574–579.

[66] M. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *Aiche Journal*, vol. 37, pp. 233–243, 1991.

[67] Y. Lin, C. Yen, and J. Wang, “Video popularity prediction: An autoencoder approach

with clustering," *IEEE Access*, vol. 8, pp. 129 285–129 299, 2020.

[68] V. L. Cao, M. Nicolau, and J. Mcdermott, "Learning neural representations for network anomaly detection," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–14, 06 2018.

[69] K. Han, H. Wen, J. Shi, K.-H. Lu, Y. Zhang, and Z. Liu, "Variational autoencoder: An unsupervised model for modeling and decoding fmri activity in visual cortex," 11 2017.

[70] K. Wang, C. Gou, Y. Duan, L. Yilun, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," vol. 4, pp. 588–598, 09 2017.

[71] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski, "Massively parallel architectures for ai: Netl, thistle, and boltzmann machines," in *Proceedings of the Third AAAI Conference on Artificial Intelligence*, ser. AAAI'83. AAAI Press, 1983, p. 109–113.

Chapter 6

Dynamic Deep Learning and Probabilistic Models to capture abnormalities in the CR-IoT Spectrum

As we described and presented jammer detection for low dimension signals by learning switching dynamic Bayesian model in chapter.4. This chapter describes the method based on VAE and DBN to capture abnormalities at the CR-IoT network's latent space level for high dimension data. An in-depth description of the proposed model architecture has been presented. The proposed method is assessed by using ROC curves and AUC metrics.

6.1 Abnormalities in the CR-IoT spectrum

The fleet-footed proliferation of emerging technologies has prompted the significance to incorporate AI techniques into an autonomous system and network because AI methods have achieved tremendous and remarkable results in performing several tasks such as object detection and tracking in the field of computer vision, and modulations classification, signal classification, anomalies detection in the radio communication domain. AI techniques have revolutionized many fields, including robotics, radar communication, self-driving cars, UAV, self-aware and autonomous radios, vehicle to vehicle communication and intelligent systems. Following the recent AI framework trends, modern communication systems have also witnessed the implementation of cognitive and self-aware capabilities into the new wireless technologies such as CR-IoT, V2V, and 5G and 6G mobile networks. Unprecedented wireless networks are more adaptive, dynamic than ever before due to such cognitive capabilities, incorporated not only at the network level but also the device level. Incredibly, extensive work has been carried to develop AI algorithms and models for the cognitive radio network at the physical layer. In this perspective, [1] describes a method using deep learning to achieve modulation classification tasks. The work [2] presents a method based on modulation classification automatically by deploying deep learning models ResNet-50 and Inception ResNet V2. The proposed framework possesses the ability to distinguish various modulation such as, Amplitude Shift Keying (ASK), Phase-Shift Keying (PSK), and Quadrature Amplitude Modulation (QAM). For 5G communication network, modulation classification is performed using CNN network in [3]. In [4], deep learning models are deployed to recognize OFDM based intelligent radios. Deep learning models have been used to perform spectrum sensing in cognitive radios in [5]. The paper [6] describes methods and techniques for developing context-aware radio systems based on deep learning. The work [7] and [8] present complex machine learning algorithms for a CR with in-depth reasoning and learning methods, and in [9] AI techniques are discussed for a CR. Reinforcement deep learning method is presented to perform coding and modulation selection tasks [10]. The paper discusses how secondary users start interfering with the primary user in cognitive radio and present a problem to PU select suitable modulation schemes. CNN is deployed which is data-driven method to perform spectrum sensing to detect legitimate user transmission in a cognitive radio [11]. The method doesn't need primary user signal profile history and SNR of the channel. Like other wireless networks, which are prone to malicious threats, CR-IoT is susceptible to jamming attacks as well. Therefore, making the CR-IoT network secure is yet a fundamental but challenging task. Hence, there have been several methods developed to autonomously detect attacks and intelligently combats threats highlighted in **section.3.5**. From this standpoint, the work [12], describes anomaly detection in a spectrum of the wireless network in an unsupervised manner using the generative network's approach. Anomalies detection in the crowd sourced sensors in the CR network is demonstrated in [13]. In [14], HMM model is deployed to detect abnormalities in a cognitive radio. Anomalies are detected using spectrum predictions method have been presented and described in [15]. GAN based anomaly detection for the radio spectrum is given in [16]. Signal anomalies detection using auto-encoder is presented in [17]. The paper [18] presents an abnormality detection method in which spectral contents and spectrograms of the radio signal have been extracted, and deep neural network are trained based on extracted features and abnormalities are detected.

6.2 Deep Learning and Probabilistic Bayesian framework to capture abnormalities in AI-enabled CR-IoT

Firstly, we begin our investigation to detect jammer inside the OFDM modulated transmission by considering a single subcarrier under attack **section.4.5**. The proposed method deploys DBN to perform state estimation at two levels, continuous and discrete, and eventually captured abnormalities (jammer attacks) in the CR-IoT spectrum. Followed by this, the succeeding work investigates single and bank-parallel DBN model implementation, which consider more subcarrier under adversary threats **section.4.6**. A more in-depth approach is carried out after this. Precisely, [19] describes a framework to detect abnormality in the CR spectrum based on data dimensionality. The approach investigates the DBN (Bayesian network) and GAN (Generative model) concurrently to present a self-aware module framework at two different CR levels. Specifically, the DBN approach addresses low dimension data, whereas; GAN handles high dimensional signals. DBN learns switching models from data series in the form of generalize state vectors where different linear models are described with the switching variables. DBN exhibits good performance when the data dimension is low, and several possible switching dynamic models in DBN is confined. Generative models felicitously address a large quantity of various dynamic models, but they are impotent to deal with uncertainties. Whereas, a recent work in the paper [20], the authors present deep learning-based spectrum anomaly detection method for cognitive mmwave radio where Conditional (C)-GAN, Auxiliary Classifier (AC)-GAN, and VAE are examined and investigated. However, the proposed approach does not provide a method to deal with uncertainties in the cognitive spectrum. Nevertheless, the modern intelligent network generates and communicates a vast amount of wireless data. Therefore, a more robust and powerful method is desirable, taking advantage of both deep learning and probabilistic network and presenting abnormalities detection framework. In this context, **VAE** from the DL domain facilitates the achievement of the data dimension reduction step, and the **DBN** from the Bayesian field fulfills the need for state estimation tasks effectively in the CR-IoT spectrum. We deploy VAE to transform high dimension data into low and compact representation. Then latent variables of VAE are clustered to learn temporal dependencies among them and constitute a probabilistic representation. We use **Apated-Markov Jump Particle Filter (A-MJPF)** to perform state estimation, which considers the uncertainties in the spectrum and consequently spot any malicious behavior that deviates from the standard etiquette in the spectrum at the continuous level. For the first time that generalized state vectors are explored and investigated containing latent spaces information obtained from the trained VAE of OFDM modulated transmission in CR-IoT to the best of our knowledge.

We deploy two VAEs, one for signal and another for signal derivative where the second VAE implementation is associated to the null-force filter implementation. Fig.6.1 presents an incremental learning process of the proposed approach, which describes a framework to introduce SA capabilities in the AI-enabled CR-IoT network. We present incremental learning processes from low-dimensional radio signals in **section.4.7** that describe model learning in an increment fashion. Under such a situation that deviates from normal, a new model is learned based on abnormal data. In Fig.6.1, we consider high dimension data, and we present steps involved in obtaining a low and compact representation of high dimensional data using the VAE model in Fig.6.2. In addition to that, Fig.6.2 describes steps that have been followed to learn the DBN model. Table.6-A describes SA model properties when high dimensional data is considered.

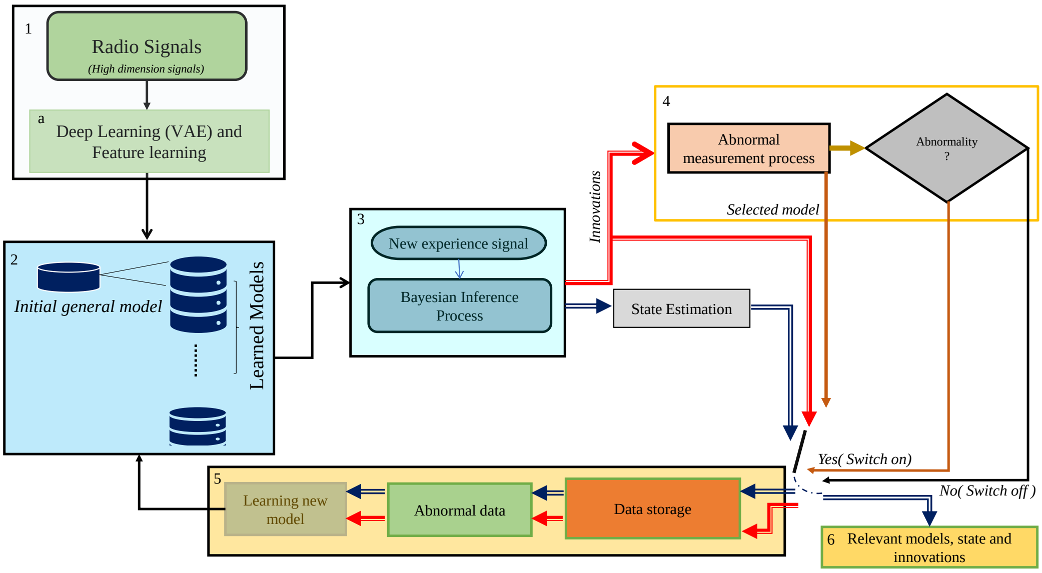


Figure 6.1: Incremental Learning Process from the high dimensional radio signals.

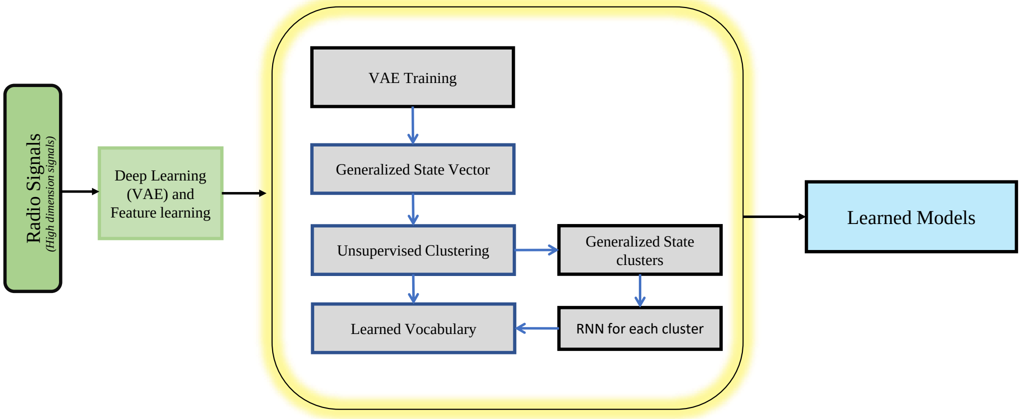


Figure 6.2: Step (a) of Fig.6.1: Transforming higher dimension data into low dimension using VAE and learning DBN model.

Table 6-A: Proposed SA model attributes for high dimensional data

Self-Awareness module Properties	**Contribution**
Generative	Learning models based on latent space (z) of high dimension data
Discriminative	Abnormal signal measurements (Jammer Signals)
Interactive	Interaction between user and jammer signals present in a Networt
Hierarchical	Three levels DBN (Discrete state, Continuous state, Observation state)
Temporal reasoning	Future state prediction by using generalized state vector
Uncertain reasoning	z latent space vector in VAE allows to make probabilistic reasoning

6.3 Dense and Heterogeneous CR-IoT network model

In this work, we esteem a CR-IoT system model shown in Fig.6.3 consisting of a base station B, two heterogeneous networks (HetNet1, HetNet2). The macro base station acts as a gateway between HetNet1 and HetNet2. Each HetNet consists of several R devices equipped with SA capabilities and a small tower ($b1$ for HetNet1 and $b2$ for HetNet2). Both HetNets are assumed to be at a distance of min interference, i-e, no co-channel interference. The base station B conducts OFDM modulated signal transmission, which encodes voice data. The cognitive devices in each HetNet are continuously sensing the environment and receive OFDM modulated signals from the respective towers ($b1$ or $b2$). Let $r(t)$ be the received OFDM signl at the terminal of any device r in a HetNet and can be expressed as:

$$r(t) = \sum_{p=0}^{L_p-1} h_l v_m(t - \delta_t - l)e^{-j2\pi fc(t-\delta_t)} + w_m(t) \tag{6.1}$$

where, $m \in [1, 2, 3...M]$ represents OFDM symbols, $w_m(t)$ is the complex additive white noise with $\mu = 0$ and $\sigma^2 = 1$, $(noise \sim N(0, 1))$, $v_m(t)$ shows transmitted OFDM signals which encode voice data, f_c shows carrier frequency, δ_t denotes time delay and h_l is the channel gain. We assume there is no time delay as transmitter and receiver are perfectly synchronized and h_l doesn't alter during sensing operation. The smart jammer is also hypothesized to be present in each HetNet. We assume that the jammer aware of the transmission protocol of the network and resource allocation scheme of the base station and attacks the radio spectrum of the CR-IoT network. OFDM data is extracted at the receiver section of the communication system and used as training and testing data for the proposed method, which is described in the following sections.

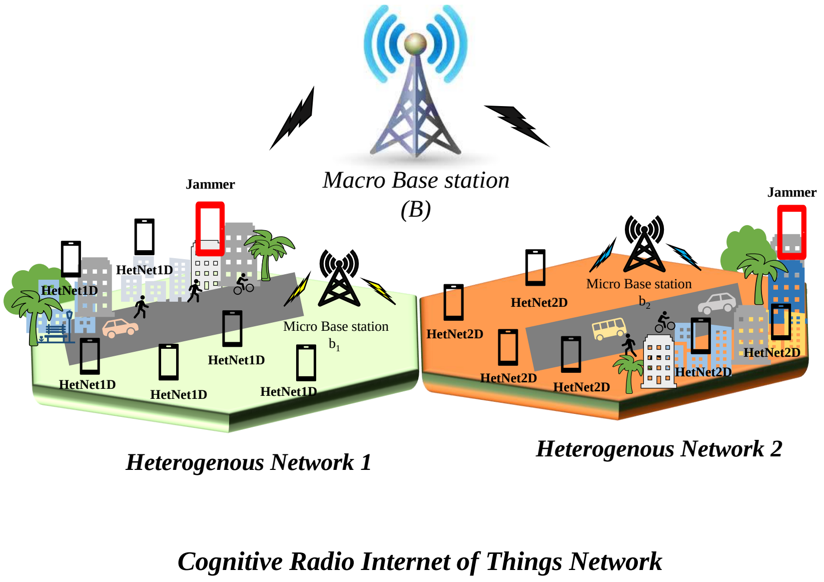
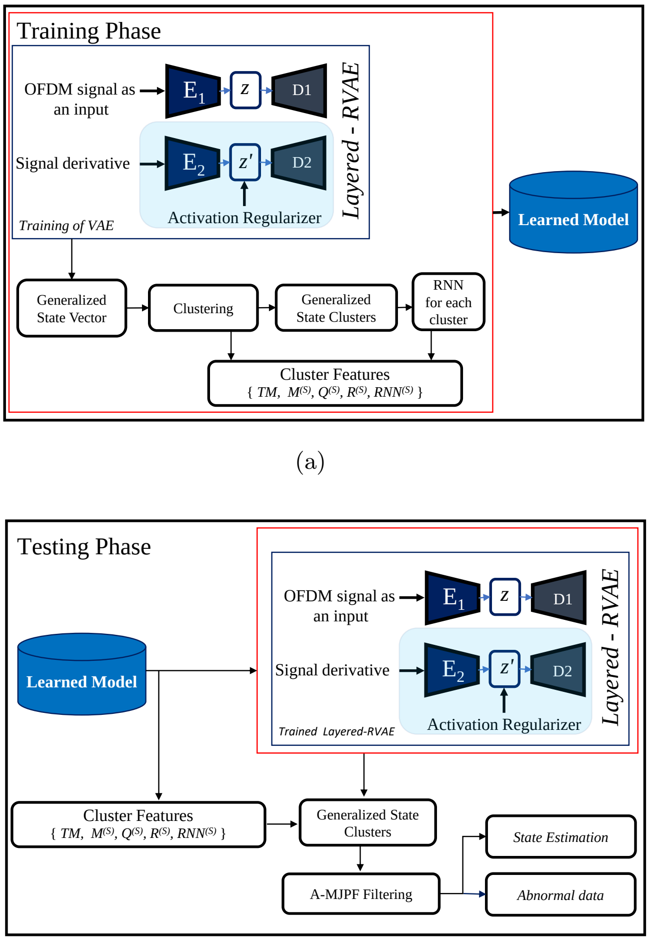


Figure 6.3: CR-IoT network composes of base station B, two heterogeneous networks (HetNet1 and HetNet2), and a jammer in each HetNet. Each HetNet consists of several cognitive devices and small towers (b1 for HetNet1 and b2 for HetNet2).

6.4 The Proposed framework

The proposed method consists of training and testing phase. Fig.6.4(a) and 6.4(b) depict the overview of the proposed framework for abnormality detection in the CR-IoT spectrum.



(a)

(b)

Figure 6.4: Proposed Model a) Training process of VAE and learning features of generalized state vector consisting of latent space information b) Testing process deploying trained VAE to obtain test latent information generalize state vector and performing state estimation using A-MJPF and detecting abnormalities.

6.4.1 Training Phase

The training phase consists of two stages. We first explain the motivation and implementation of the proposed model in the first stage. And then, we describe the learning process of the DBN network in the second stage.

Stage 1 - Deployment of VAE To obtain a reduced dimensionality representation of spectrum data, VAE is deployed. VAE learns a mapping between higher dimension into low dimension latent space vector $\tilde{z}$ and automatically extracts relevant features from the original received signal. This work considers OFDM modulated signals transmission, multi-carrier in-phase and quadrature components of transmitted OFDM symbols during communication in which jammer is absent are used to train VAE. The input to the VAE so contains OFDM subcarriers SC and symbols M representing portions of a spectrum and is expressed in terms of in-phase and quadrature components as,

$$\tilde{\gamma}_k = [I(SC_k^1), I(SC_k^2), ...I(SC_k^N), ...Q(SC_k^1), Q(SC_k^2).....Q(SC_k^N)]_{k=[1,2..M]}^T \tag{6.2}$$

where, N and M show subcarriers and symbols of OFDM signal respectively, and $\tilde{\gamma}_k$ is 1000-dimensional vector. The decoder learning the likelihood function assumes that the signal should be reconstructed as close to the input signal. In this variational approach, we use generalized coordinates in which signal changes as time evolves. Such evolution of a time is considered by providing the signal derivative of the original OFDM symbol vectors to a separate VAE. In this context, the input to the second VAE can be written as,

$$\dot{\tilde{\gamma}}_k = [I(\dot{SC}_k^1), I(\dot{SC}_k^2), ...I(\dot{SC}_k^N), ...Q(\dot{SC}_k^1), Q(\dot{SC}_k^2).....Q(\dot{SC}_k^N)]_{k=[1,2..M]}^T \tag{6.3}$$

where, $\tilde{\gamma}_k = \frac{\tilde{\gamma}_k^{L-1} - \tilde{\gamma}_{k-1}^{L-1}}{\Delta_k}$ and L shows L^{th} time derivative of the state. We consider only first order derivative (i-e, $L = 1$). This is because we deploy DBN with limited memory, which is a two-layered DBN so that dynamical models depend only on variables in the preceding slice.

The second VAE estimates another Gaussian vector of variables mean μ and variance σ^2, representing the signal derivative encoded in latent space $\tilde{\dot{z}}$. However, in this case, the decoder of the VAE is forced to reconstruct a null mean noise version of a derivative. A regularization parameter ρ is needed to allow VAE weights not to collapse, which puts constraints on the activation of the neurons in the latent layer [21]. Such constraints force neurons to learn proper representation from the input data wherein all neurons remain active and participate in the reconstruction of the signal in the decoder. This ensures the learning process to reconstruct a latent space derivative component by which the different dynamics of symbols can be compared to a common reference, i.e., the null mean noise. This scheme is inspired by the structure proposed for initial filters in incremental learning procedures when low dimensionality signals are considered. In that case, a particular type of Kalman filters, i.e., null force filter (NFF), is used that assumes an agent's behavior is locally static (i.e., its derivative should be null) [22]. For the training sequence, this can be violated, so the prediction errors of the NFF can be used to learn different modalities by which violations occur by unsupervised clustering of generalized states. In the case of null reconstructed output VAE, the null derivative hypothesis allows forcing in the loss function the constraint that each of the samples should have a comparable latent derivative depending on different characteristics of the derivative signal. We expect that latent variables differences back map deviations of derivatives in encoded variable and vice versa. The overall architecture is given a name as, *Layered-RVAE*, where R means regularized model. Both VAEs are individually trained, and after the training of both VAEs, we form a generalized state vector Z_{Train} containing the original signal and its derivatives; both encoded into the latent spaces and is expressed as,

$$Z_{Train} = [\tilde{z_k}, \tilde{\dot{z_k}}]^T \tag{6.4}$$

Stage 2 - Learning Probablistic Dynamic Bayesian Network After training of VAE and obtaining latent spaces in the form of the generalized state vector, which contains signal information describing normal behavior of the signal in the spectrum, the generalized state vector is used to learn the DBN network as shown in Fig.6.5. With the time evolution, DBN defines a dependency between random variables. Different level of inferences is extracted about the spectrum dynamics by using DBN. In this context, X_k shows measurement, which is related to the lowest inference level corresponding to the received OFDM signal. Continuous information of the spectrum is encoded into latent space by using state Z_k, associated with the medium level of inference. Superstates S_k associated with the highest level of inference, which comprises of continuous information discretization. Along with that, arrows in DBN represent conditional probabilities between involved variables. Vertical arrows expedite to define causalities between both continuous and discrete levels of inference and observed measurements. Horizontal arrows depict temporal antecedents between hidden variables. It is worth mentioning here that for a DBN model introduced in **section.4.5**, X_k represents a medium level of inference, and Z_k corresponds to the lowest level of inference. Here, in this high dimensional signals work, the medium level of inference is represented by Z_k, which is associated with the latent space of the VAE, and X_k is associated with the lowest inference level. An unsupervised clustering algorithm Growing Neural Gas (GNG) [23] is used to obtain clusters, which takes a generalized state vector as an input and provides a group of super-states S containing cognate data. The μ_k and $\dot{\mu_k}$ values are used in the clustering process to take into consideration the signal and its dynamics with respect to the next signal values. We can express a total number of clusters as;

$$S = \{S_1, S_2, \ldots, S_C\} \tag{6.5}$$

where $S_i \in \boldsymbol{S}$ and C is the total number of superstates.

Along with, mean $M^{(S)}$, covariance $Q^{(S)}$ and radius $R^{(S)}$ are learned as additional features of each clusters (refer **section4.5.1** for cluster features). A transition matrix TM is also obtained which encodes transition probabilities from each cluster to other clusters. To exploit the dynamics of GS which is corresponding to the continuous predictive model for each cluster S_i, a Recurrent Neural Network (RNN) network is deployed. For each cluster, RNN is trained by taking μ_k an input and $\dot{\mu_{k+1}}$ as an output where both $[\mu, \dot{\mu_{k+1}}]$ $\in S$. The VAE provides σ^2 of encoded information along with μ. To include uncertainty of latent space and completely define Gaussian N(μ,σ^2), $2D$ additional inputs and output are considered as well, where D is latent state dimension. Hence sigma-points which are associated with (μ_k, σ_k^2) can be expressed as [24],

$$\mu_k^j = \mu_k + (\sqrt{(D+\lambda)\Sigma_k})_j \ j = 1, 2.......D \tag{6.6}$$

$$\mu_k^j = \mu_k - (\sqrt{(D+\lambda)\Sigma_k})_j \ j = D+1..2D, \tag{6.7}$$

where λ is a scaling parameter and $\Sigma_k = I_D \sigma_k^2$, where I_D is identity matrix of dimension D. Based on specific spectrum information captured inside each cluster, RNN learns signal information in that particular spectrum portion and facilitates the prediction of the next signal values. This can assist when the model takes a new observation and that observation doesn't follow the previously learned scenario encoded inside each RNN network for every cluster; an abnormal situation will be emerged. This is due to the reason that RNN predictions are not following the observations. Therefore, a model should learn new situations and generate further information.

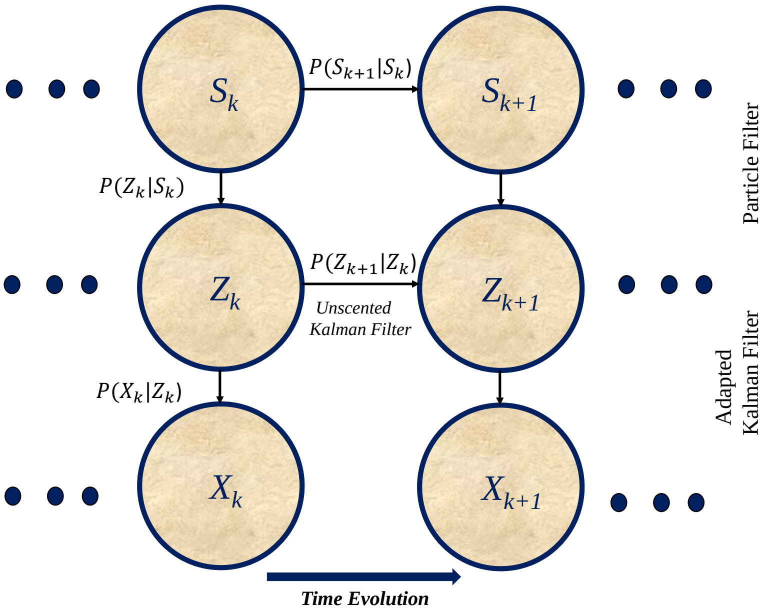


Figure 6.5: DBN model entails discrete and continuous layers to make inferences about spectrum evolution. At a discrete layer, a Particle filter is deployed, whereas, at the continuous level an adapted Kalman filter is employed.

6.4.2 Testing Phase

Fig.6.4(b) highlights the steps that have been followed in testing phase. At first, test signal derivative is determined and than given to the trained Layered-RVAE (both signal and it's derivative). Generalized state (GS) Z_{Test} is formed which contains latent space information encoding both state and derivative values for each time instant k. After obtaining, GS, A-MJPF [25] has been selected to perform inferences on DBN model as shown in Fig.6.5. Unlike MJPF [26] which deploys bank of KF at continuous level for prediction and detection of abnormalities, A-MJPF employs a modified version of KF which is Unscented Kalman Filter (UKF) to accomplish prediction tasks. Hence, at continuous level, bank of UKF is implemented for prediction. The necessity to employ a modified version of KF comes from the fact that a problem address in this work requires a non-linear model for prediction and non-linear observation model solved by RNN and VAE. Therefore, standard KF can't be implemented at the state level. A-MJPF performs prediction and update operation for each time instant k. A-MJPF essentially performs prediction and update steps that are described as follows:

- Prediction: The prediction of next super-state $P(S_{k+1}|S_k)$ at discrete level and next GS $(P(Z_{k+1}|Z_k)$ for each estimated particle at continuous level is done in the predicted step of A-MJPF. A-MJPF employs PF to predict super-states employing TM matrix information for each particle at discrete level. At continuous level for each selected super-state S_k, RNN performs predictions, and UKF is used to performs estimation as non-linear models are involved at continuous level.

- Update: When a new measurement is taken an update step is performed in A-MJPF. The particles are re-sampled at discrete level. The details of update process is given in [25].

6.4.3 Abnormality Measurements

At the continuous layer, predicted values are compared with the updates by determining probabilistic distance measurements that allow the A-MJPF to estimate the abnormality. In this work, we have used $db2$ as an abnormality measurement that defines the likeness between the predicted state and successive observation at a continuous level in each super-state, and it is expressed as,

$$db2 = -\ln \int \sqrt{p(Z_k^*|Z_{k-1}^*(S_k^*))p(X_k|Z_k^*)dZ_k^*}; \tag{6.8}$$

The prediction step aims to predict future signal values, and such predicted values compliance the same rules which had been learned by a model during the learning phase. If predictions don't pursue the learned rules, an abnormal situation emerges.

6.5 Deep VAE network architecture and configuration

We conducted experiments on simulated data. We consider IEEE802.11 ah configurations operating in 16 MHz channel BW, supporting 512 subcarriers with $64 - QAM$ modulation. Our CR-IoT network contains several devices in both HetNets. We use normalized power for both signal and jammer and evaluate the data at the receiver side of the cognitive device. We divide data into two sets (Training and Testing): 1) Training set contains standard data use to train Layered-RVAE model to get latent vectors for GS formation and than learn the DBN model. 2) The test data contains abnormalities and used during the testing phase. The VAE architecture comprises of an encoder network and decoder network where the encoder is modelled with fully connected convolutional neural network layers with LeakyReLU activation functions. For training, a learning rate of 0.00001 is used with mini-batch size of 32 over the data size of 8192000, and an adaptive moment estimation (ADAM) optimizer is deployed. The latent space dimension 30 is selected after performing several experiments on different latent size and 30 turned out to be an optimal value for our work. Similar configurations of the network and training parameters are adopted for second VAE (in case of derivative as an input) except regularizer parameter which we call as an activation regularizer ρ, and have selected values $0.1, 0.4, 0.6, 0.8, 1.0$. The motivation of implementing activation regularizer is introduced in the training process (refer **section.6.4.1**). We train our model in a normal situation (under no jammer). We use γ_{train} and $\gamma_{t\dot{r}ain}$ during the training process described in **section.6.4.1**) and, consequently, obtain trained Layered-VAE from which clusters and corresponding predictive models (RNN) are learned based on signal data under no jammer attacks. We perform cluster evaluation by considering the different ρ values and evaluating the impact of introducing the activation regularizer at the cluster level presented in the **section.6.5.1**. We now describe the considered scenarios in detail and present abnormality detection by the proposed framework, followed by an evaluation of the model in terms of the ROC curves and AUC tables.

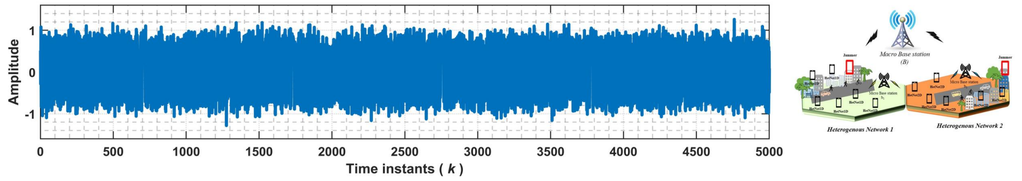


Figure 6.6: Reference situation describes transmission under jammer less and interference free environment. There are neither jamming attacks nor interfering source signals in the communication.

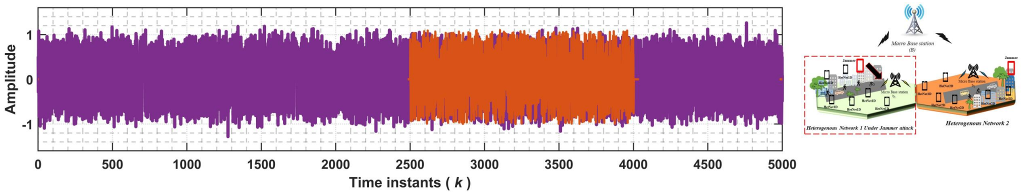


Figure 6.7: Jammer launches malicious attacks in the HetNet1 transmission.

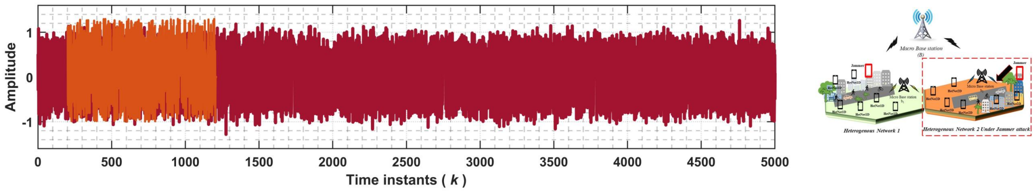


Figure 6.8: Jammer launches malicious attacks in the HetNet2 transmission.

- Scenario A (*Jammer less or interference free environment*): Fig.6.6 describes a normal situation with no jammer attacks or any other interfering source. The model is learned by applying normal data (no jamming signals) to the network during a training phase. Afterward, the trained model is deployed to test different abnormal situations, i-e, scenarios B,C,D and E, respectively.

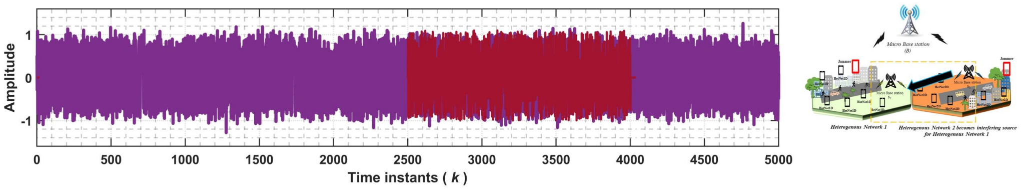


Figure 6.9: Devices in HetNet1 are receiving signals, and HetNet2 becomes an interfering source for the devices in HetNet1. Eventually, for the devices operating in HetNet1, source of transmission changes, which is an abnormal situation.

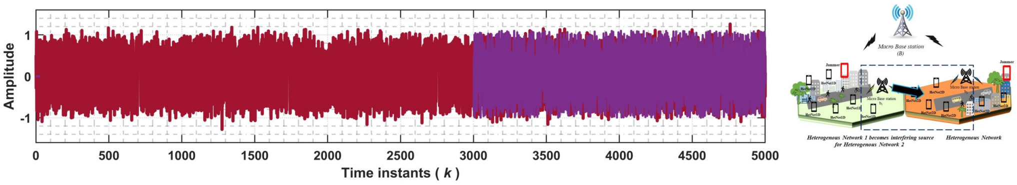


Figure 6.10: Devices in HetNet2 are receiving signals, and HetNet1 becomes an interfering source for the devices in HetNet2. Eventually, for the devices operating in HetNet2, source of transmission changes, which is an abnormal situation.

- Scenario B ($HetNet1$ *under attack*): Fig.6.7 presents HetNet1 under jamming attacks. The jammer injects disruptive signals starting from 2500 time instant and lasts up to 4000-time moments.

- Scenario C ($HetNet2$ *under attack*): This scenario illustrates the situation when HetNet2 comes under the jamming attacks. Correspondingly, Fig.6.8 depicts HetNet2 transmission where jammer corrupts the signals starting from 200-time instants and lasts until 1200-time instants.

- Scenario D ($HetNet2$ *as a abnormality for* $HetNet1$): Cognitive devices in HetNet1 receive signals from tower $b1$ and perceives it as a usual transmitting source for all devices that exist in a HetNet1. However, if any device receives signals from $b2$, which is the source transmitter of HetNet2, it appears as an abnormality for the device operating in HetNet1. This is due to the reason that devices in respective network (HetNet1) expect signals from source $b1$. This situation is shown in

Fig.6.9 where HetNet2 interference appears between 2500 and 4000-time instants in the HetNet1 transmission.

- Scenario E ($HetNet1\ as\ a\ abnormality\ for\ HetNet2$): A similar situation observes for HetNet2 when any device of the respective networks catches signals from $b1$; it appears as an abnormality for such device being operated inside HetNet2. This scenario is illustrated in Fig.6.10 where HetNet1 becomes intervene with HetNet2 communication and explicitly appears from 3000 time instants.

6.5.1 Cluster analysis of the learned latent space models

Clustering techniques discover distinct and meaningful patterns in data. It is a process that partition data into a group containing similar information [27]. Or in other words, we can define clustering as a method that assigns data points from the given data set $\mathfrak{D}$ to N clusters or groups while minimizing the distance between each data point and center of the cluster. Each cluster is also known as a node or a neuron. We can evaluate the clustering algorithm's performance based on different metrics such as minimum loss function or several clusters required for a specific system.

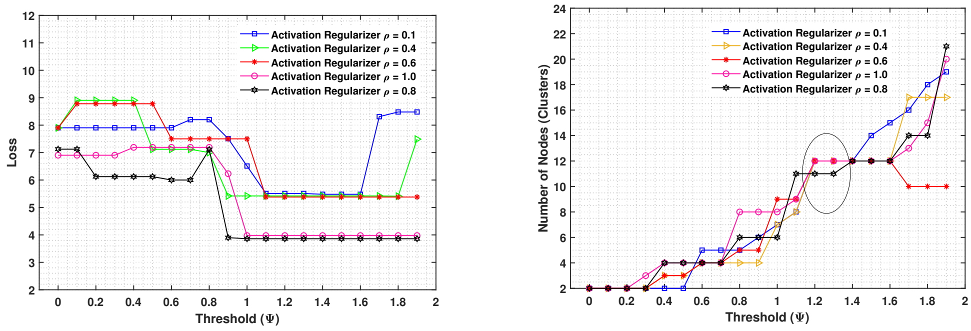
In our work for low dimensional data introduced in **section.4.5**, we deploy SOM to obtain discrete regions of a DBN during the learning phase. However, during learning, SOM exhibits some limitations: few dead neurons are produced that don't perform the system's inference task instead consume resources. To deal with such an issue, we employ the GNG clustering algorithm to cluster latent space vectors. We also perform optimization of the GNG algorithm to select the best system configuration and parameters. According to [28], we use utility function Ψ which is defined as ratio between min utility and max error and can be expressed as,

$$\frac{e^i_{max}}{u^i_{min}} > \Psi; \tag{6.9}$$

$\Delta u^i_0 = e^i_1 - e^i_2$ is the difference between the two closest data nodes of the current data point i. The threshold [29] removes irrelevant nodes frequently in a clustering algorithm.

We select value of Ψ with respect to the minimum loss function as defined in [28]. In Fig.6.11(a), we plot loss function over the range of threshold Ψ for the different ρ values. Note that the input to the clustering algorithm is a generalized state vector equation.6.4 obtained after the training of VAE. During training, we select distinct ρ values for the second VAE (for the derivative) and obtain generalized state vectors associated with different values of ρ. It can be noted in Fig.6.11(a), the optimal value of Ψ is 1.2, where the loss is minimum for all clustered latent spaces GS. It can be observed that the $\rho = 0.8$ has min loss among all at $\Psi = 1.2$. In Fig.6.11(b) we evaluate the effect of Ψ on network complexity. Given $\Psi = 1.2$, we select optimum number of cluster as 12 to perform clustering of latent space GS vectors.

We analyze the performance of clusters obtain for different ρ values given $\Psi = 1.2$ and $C = 12$. It can seen that the clusters in case when $\rho = 0.8$ is good. However, clusters are bad for $\rho = [0.1, 0.4, 0.6]$. Moreover, we can refer to transition matrices for different ρ values. In case when $\rho = 0.8$, transition matrix is quite improved. Therefore, we deduce the following conclusion that introducing activation regularizer during training of VAE (when input is derivative) has facilitated to capture meaning full information from the input data.



(a)

(b)

Figure 6.11: Cluster evaluation a) Loss function by changing threshold b) Network complexity verses cluster numbers.

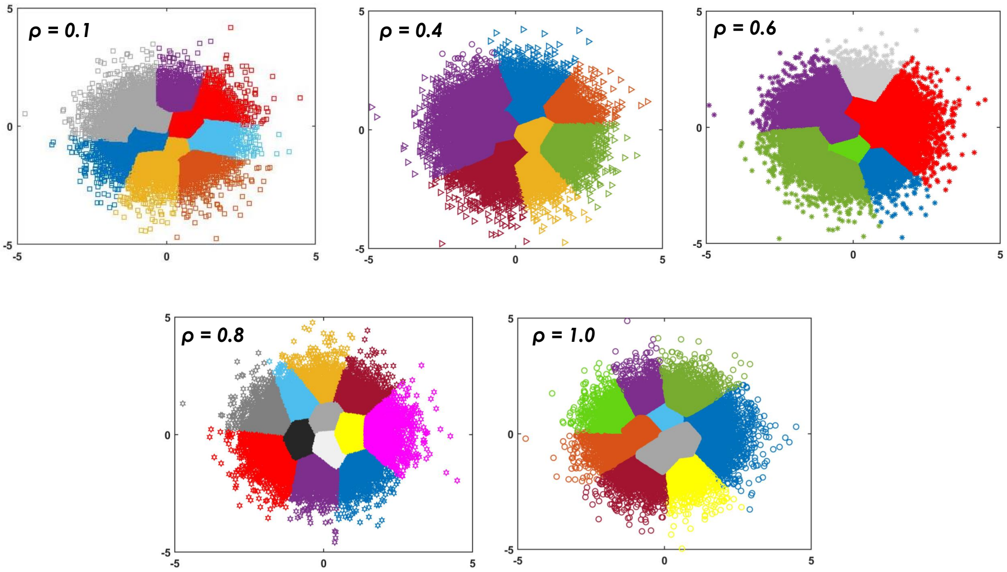


Figure 6.12: Cluster analysis of latent spaces for different ρ values.

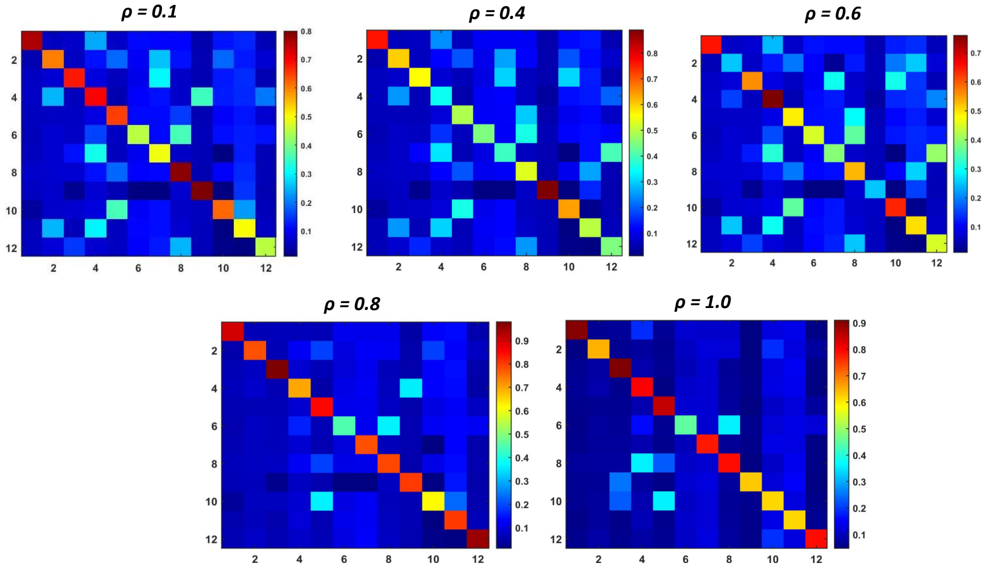


Figure 6.13: Transition Matrices comparison for different ρ values.

6.6 Testing the learned model

The learned model is tested with new measurements. The prediction step aims to predict future signal values, and such predicted values compliance the same rules which had been learned by a model during the learning phase. If predictions don't pursue the learned rules, an abnormal situation emerges. In this perspective, we use abnormality measurement (db2) at the continuous level of the DBN model to identify new observations are assenting with the learned rules or not.

It can be seen from Fig.6.14 and Fig.6.15 that the Layered-RVAE deep and probabilistic model can detect abnormal behavior (jammer attacks) for HetNet1 and HetNet2 networks, respectively. Consequently, figures show abnormality signals at the continuous inference layer. The abnormal signal is low when the probability of having a prediction is close to the measurements (Likelihood of how much prediction is compliance by the observation). On the contrary, the abnormality is high when predicted values are not close to the measurements. Fig.6.14 and Fig.6.15 indicate that the filtering process provides high abnormality signals for the time instants k where the jammer attacks are present. The signal above the threshold is considered abnormal and it is determined by calculating the mean of equation. (6.8) plus its standard deviation. Fig.6.16 presents an abnormality detection for scenario D. For a given situation, cognitive devices are receiving a transmission from HetNet1. However, during HetNet1 communication, HetNet2 starts interfering with HetNet1 communication, and therefore, an abnormality condition emerges. The proposed framework is competent to detect HetNet2 signals shown in Fig.6.16 (prevailing between 2000 and 4500 time-instants). Similarly, HetNet1 becomes a tampering source for the HetNet2 network and eventually detected by the model, as shown in Fig.6.17.

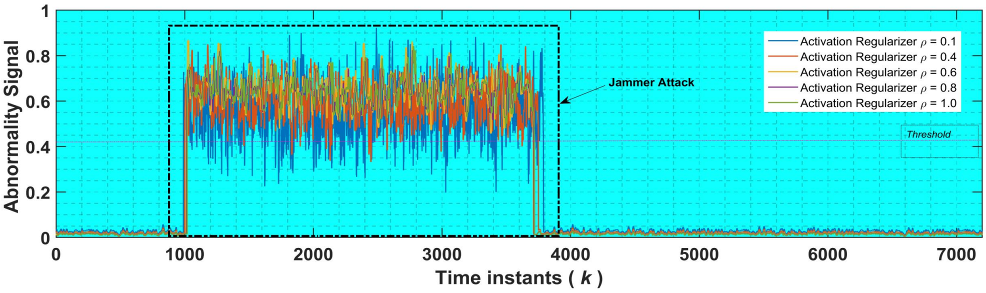


Figure 6.14: Abnormality detection by the Layered-RVAE model when HetNet1 under jammer attack.

In this work, we perform simulations using different ρ values to evaluate the effect of activation regularizer on the performance of second VAE (in case where input is the

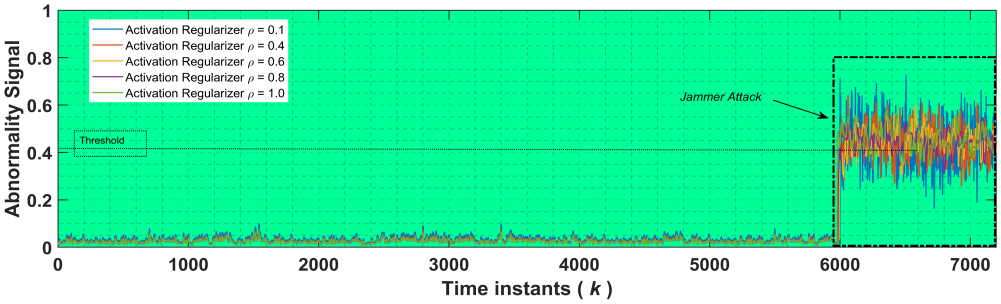


Figure 6.15: Abnormality detection by the Layered-RVAE model when HetNet2 under jammer attack.

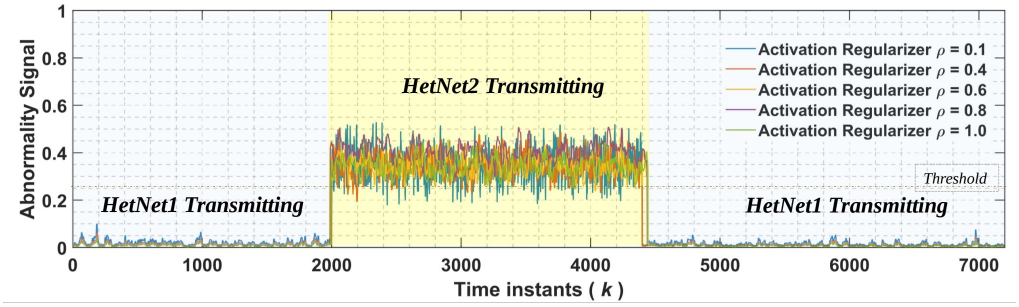


Figure 6.16: HetNet1 communication in which Hetnet2 becomes a source of interference and start transmission. Such interference is an abnormality for the HetNet1 and eventually detected by the Layered-RVAE model.

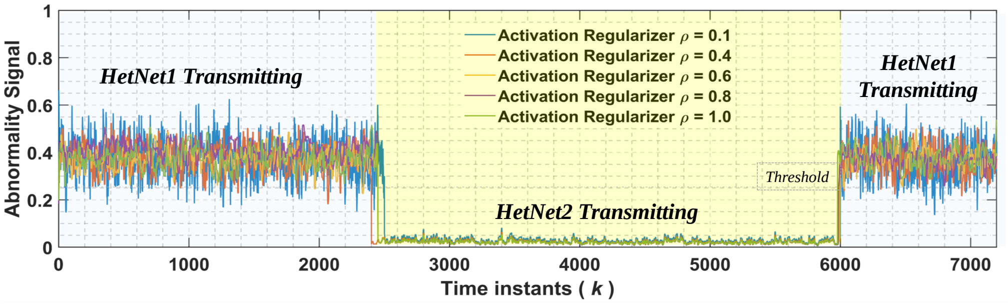


Figure 6.17: HetNet2 communication in which Hetnet1 becomes a source of interference and start transmission. Such interference is an abnormality for the HetNet2 and eventually detected by the Layered-RVAE model.

derivative of the signal) and, consequently, on the over-all Layered-RVAE model. Therefore, Figs.6.14, 6.15, 6.16 and 6.17 show abnormality signals using different values of activation regularizer, i-e $(0.1, 0.4, 0.6, 0.8, 1.0)$. We plot the accuracy scores against the various value of the activation regularizer. It is evident from the Fig.6.18 that $\rho = 0.8$ is the best value of activation regularization for all scenarios (B,C,D and E). There is a trade-off in the selection of the activation parameter. If we select $\rho = 1$, the model becomes standard VAE. Under this situation, VAE for the derivative as an input will neither exploit the hidden layer capacity nor put any constraints in decoding the signal. On the contrary, when select $\rho = 0$, the VAE turns into the vanilla VAE and serves all of its model capacity in sample generation. Subsequently, after extensive training of the proposed model, we discovered 0.8 is the optimal value of ρ on which model is demonstrating potential performance and capable of achieving high accuracy. It is worth to mention here that we have introduced the activation parameter for the 2^{nd} VAE that is being trained on a derivative of the OFDM signal as an input.

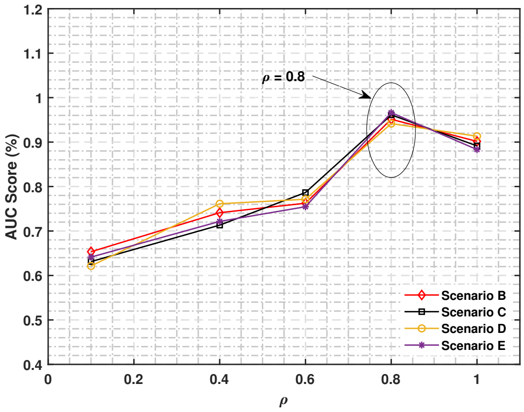
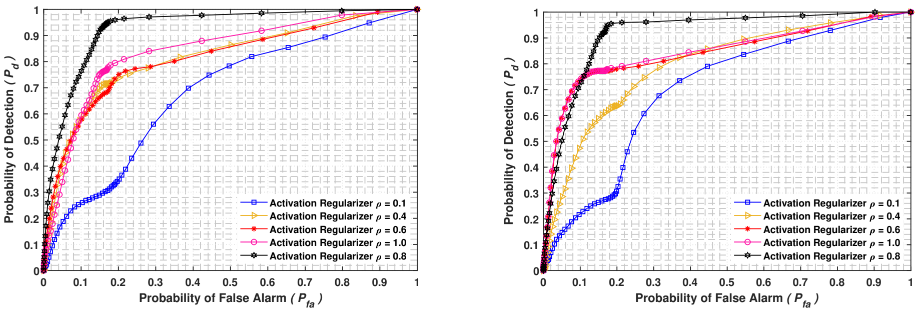


Figure 6.18: AUC scores for different value of activation regularizer under various scenarios.

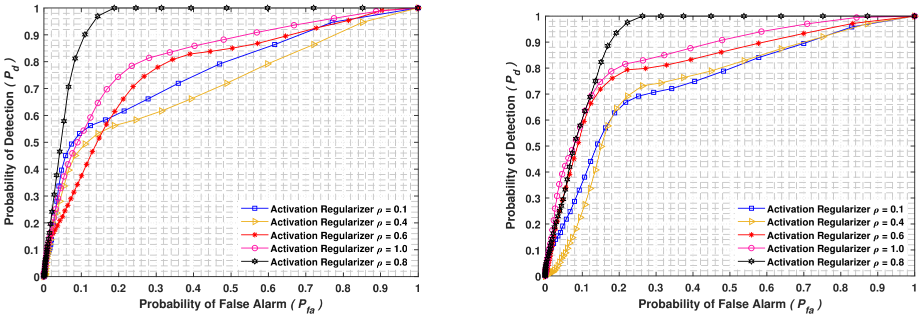
6.6.1 Performance evaluation using ROC along with AUC

We use ROC curves along with area under the curve (AUC) as evaluation metrics to assess the proposed model's performance. Fig.6.19, and Fig.6.20 confirm the viability of the proposed method and validate the excellent performance of the model. As mentioned earlier, we found $\rho = 0.8$ is an optimal value, and it is evident from ROC curves that the model achieves high detection with low P_{fa} for all scenarios when $\rho = 0.8$. Furthermore, we extract the AUC values, which are listed in Table.6-B.



(a) (b)

Figure 6.19: ROC curves a) Detection performance of the Layered-RVAE model when HetNet1 under jamming attacks b) Detection performance of the Layered-RVAE model when HetNet2 under jamming attacks.



(a) (b)

Figure 6.20: ROC curves a) Detection performance of the Layered-RVAE model for HetNet1 communication in which Hetnet2 becomes a source of interference b) Detection performance of the Layered-RVAE model for HetNet2 communication in which Hetnet1 becomes a source of interference.

6.7 Layered-RVAE and Layered-VAE implementations

We now analyze the two implementations (standard VAE and regularized VAE). We call the two realizations as **Layered-VAE** and **Layered-RVAE**, where RVAE indicates that we have activation regularization implementation in the second VAE when the input is OFDM signal derivative. We have selected the value of $\rho = 0.8$. We evaluate the perfor-

Table 6-B: AUC for all scenarios (B,C,D,E)

AUC (%)				
ρ	**Scenario B**	**Scenario C**	**Scenario D**	**Scenario E**
1.0	0.9016	0.8915	0.9132	0.8813
0.8	**0.9513**	**0.9612**	**0.9416**	**0.9663**
0.6	0.7625	0.7865	0.7614	0.7546
0.4	0.7412	0.7132	0.7613	0.7216
0.1	0.6534	0.6312	0.6214	0.6413

mance of both realizations by considering the following situations:

Situation A: HetNet1 communication under jamming threats. We have considered single jammer transmission (SJT) and multiple jammer transmission (MJT) in the accumulative transmission window of HetNet1.

Situation B: HetNet1 is transmitting, and HetNet2 becomes a source of interference for the devices in HeTNet1. Similarly, HetNet2 communication wherein HetNet1 transmission is an intervention for the HetNet2 devices.

Situation C: Changing SNR values for the communication for HetNet1 under single jammer attack transmission.

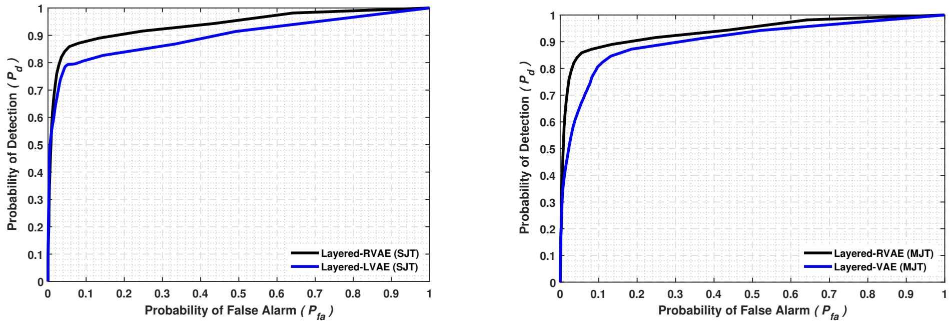
Situation D: Investigating different latent space vector sizes, i-e, 10, 20, and 30 for HetNet1 under single jammer jammer transmission (SJT).

Situation E: Jammer attacks with varying power, i-e, low, medium, and high power. The high jammer power means the jammer is attacking the transmission with considerably high power for HetNet1 communication and vice versa.

6.7.1 Performance evaluation

We analyse the performance in terms of ROC curves and AUC metrics. Accordingly, Fig.6.21(a) and Fig.6.21(b) depict the detection probability of both realizations for **situation A**. The Layered-RVAE outperforms the Layered-VAE for SJT and MJT for the HetNet1 network. The reason lies in the fact that Layered-RVAE deploys activation regularization, which facilitates to improve model performance as described in

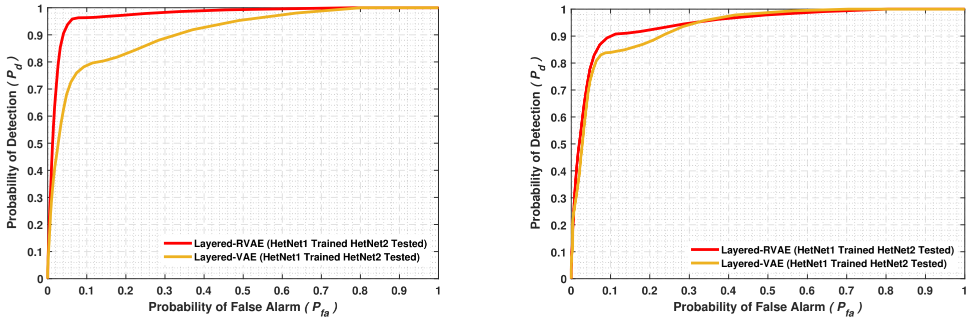
section.6.4.1. Correspondingly, Fig.6.22(a) and Fig.6.22(b) depict the models performance for the **situation B** in which HetNet1 is transmitting, and HetNet2 becomes a source of interference and appears as abnormality signals and vice versa.



(a)

(b)

Figure 6.21: ROC curves a) Detection performance of Layered-RVAE and Layered-VAE realizations for HetNet1 under single jammer transmission (SJT) b) Detection performance of Layered-RVAE and Layered-VAE realizations for HetNet1 under multiple jammer transmission (MJT).



(a)

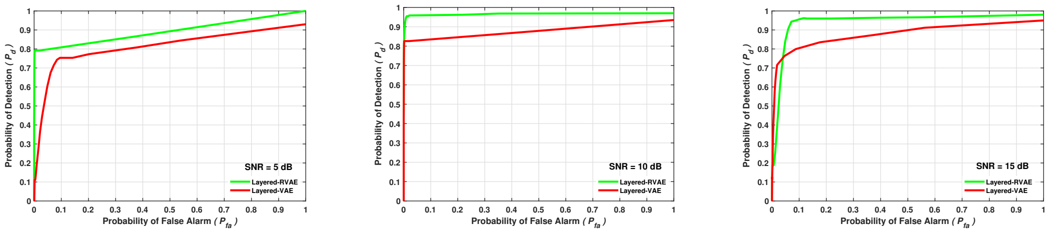
(b)

Figure 6.22: ROC curves a) Detection performance of Layered-RVAE and Layered-VAE realizations when HetNet1 transmitting while HetNet2 interfering b) Detection performance of Layered-RVAE and Layered-VAE realizations when HetNet2 transmitting while HetNet1 interfering.

We consider the different values of SNR to take into account the channel effects on the Layered-RVAE and Layered-VAE models performance. Consequently, 5,10, and 20 SNR values in (dB) have been selected **situation C**. Fig.6.23(a), (b) and (c) demonstrate both implementation (Layered-RVAE and Layered-VAE) performance whereas, Table.6-C gives AUC values on different SNR values.

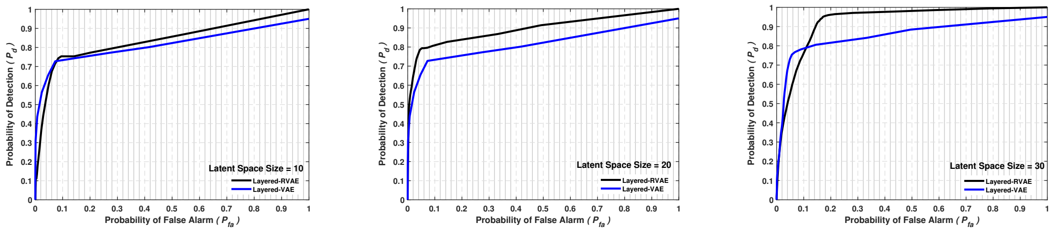
situation D work investigates the implication of taking different latent space sizes, i-e, 10, 20, and 30. We implement both models by using different latent vector sizes and evaluated the performance. consequently, Fig.6.24(a), (b) and (c) represent both model performance. For latent space $z = 10$ Layered-RVAE and Layered-VAE performance are comparable. Nonetheless, for $z = 20$ and 30, Layered-RVAE outperforms the Layered-VAE. Table.6-D shows AUC values for the different latent size.

We inspect Layered-RVAE and Layered-VAE in the **situation E** when jammer bombards attacks with varying strength. For simplicity, we have considered low, medium, and high power of jammer. The high power describes the status when jammer power is relatively higher than the signal power and low jammer power means jammer power is low as compared to the standard transmission. Fig.6.25 and Fig.6.26 illustrates model performance under low, medium, and high threats for single jammer transmission (SJT) and multiple jammer transmission (MJT) in HetNet1, respectively. In every case, the Layered-RVAE performance is better than the Layered-VAE model. Table.6-E and Table.6-F present AUC values for SJT and MJT, respectively.



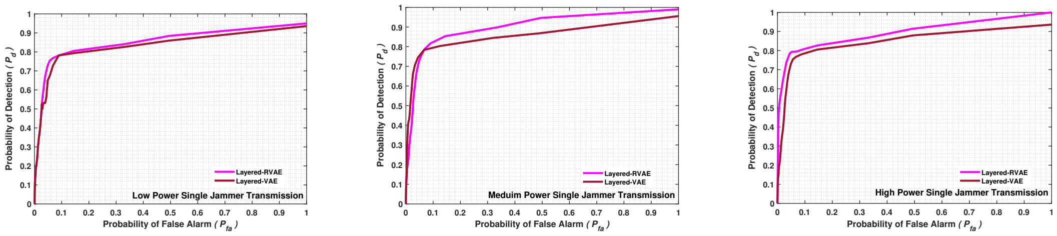
(a) (b) (c)

Figure 6.23: Accuracy plots a) ROC for SNR 5dB b) ROC for SNR 10dB c) ROC for SNR 15dB.



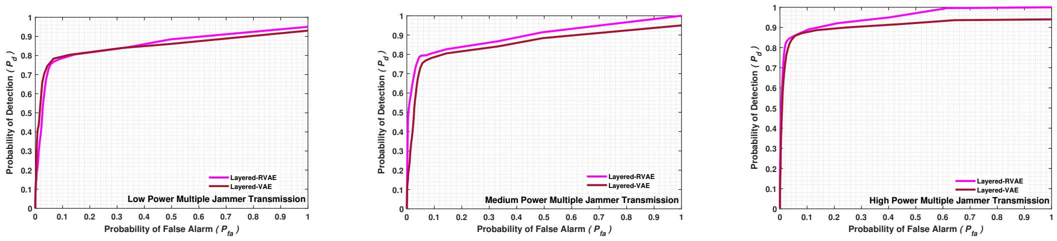
(a) (b) (c)

Figure 6.24: Accuracy plots a) ROC for latent space size 10) ROC for latent space size 20 c) ROC for latent space size 30.



(a) (b) (c)

Figure 6.25: Accuracy plots a) ROC for Low Power) ROC for medium power c) ROC for High power.



(a) (b) (c)

Figure 6.26: Accuracy plots a) ROC for Low Power) ROC for medium power c) ROC for High power.

Table 6-C: AUC Different SNR.

AUC(%)			
SNR(dB)	***5***	***10***	***15***
Layered-RVAE	0.8982	0.9423	0.9698
Layered-VAE	0.8235	0.8683	0.8886

Table 6-D: AUC Different Latent Size.

AUC(%)			
Latent Space Size	***10***	***20***	***30***
Layered-RVAE	0.8432	0.9057	0.9317
Layered- VAE	0.8225	0.8436	0.8713

Table 6-E: AUC SJT.

AUC(%)			
Single Jammer Transmission (SJT)			
Jammer Power	***Low***	***Medium***	***High***
Layered-RVAE	0.8714	0.9057	0.9260
Layered- VAE	0.8378	0.8499	0.8766

Table 6-F: AUC MJT.

AUC(%)			
Multiple Jammer Transmission (SJT)			
Jammer Power	***Low***	***Medium***	***High***
Layered-RVAE	0.8714	0.9257	0.9694
Layered- VAE	0.8406	0.8713	0.9142

6.8 Summary

This chapter presents a joint framework based on deep learning and a probabilistic model for abnormality detection in the CR-IoT spectrum. Accordingly, the proposed method deals with high dimensional data that take advantage of VAE to obtain low-dimensional latent spaces and then use DBN to perform state estimation and detect abnormalities. We introduce an activation regularizer in the VAE realization, and we call the proposed implementation as Layered-RVAE. We present cluster analysis of distinct latent spaces obtained by using different ρ values. This chapter also discusses introducing SA model properties by the proposed method for high dimensional data. The chapter also presents various scenario analyses of the Layered-RVAE implementation compared with the Layered-VAE model (with no activation regularizer).

References

[1] R. Yilmaz and A. E. Pusane, "Deep learning based automatic modulation classification in the case of carrier phase shift," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 354–357.

[2] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav, "Automatic modulation classification based on constellation density using deep learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1275–1278, 2020.

[3] A. P. Hermawan, R. R. Ginanjar, D. Kim, and J. Lee, "Cnn-based automatic modulation classification for beyond 5g communications," *IEEE Communications Letters*, vol. 24, no. 5, pp. 1038–1041, 2020.

[4] M. Fadul, D. Reising, and M. Sartipi, "Identification of ofdm-based radios under rayleigh fading using rf-dna and deep learning," *IEEE Access*, pp. 1–1, 2021.

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

[6] F. Paisana, A. Selim, M. Kist, P. Alvarez, J. Tallon, C. Bluemm, A. Puschmann, and L. DaSilva, "Context-aware cognitive radio using deep learning," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2017, pp. 1–2.

[7] L. Gavrilovska, V. Atanasovski, I. Macaluso, and L. A. DaSilva, "Learning and reasoning in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1761–1777, Fourth 2013.

[8] T. C. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *2008 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, 2008, pp. 1–8.

[9] N. Abbas, Y. Nasser, and K. E. Ahmad, "Recent advances on artificial intelligence and learning techniques in cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 174, Jun 2015. [Online]. Available: https://doi.org/10.1186/s13638-015-0381-7

[10] L. Zhang, J. Tan, Y. Liang, G. Feng, and D. Niyato, “Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3281–3294, June 2019.

[11] J. Xie, C. Liu, Y. Liang, and J. Fang, “Activity pattern aware spectrum sensing: A cnn-based deep learning approach,” *IEEE Communications Letters*, vol. 23, no. 6, pp. 1025–1028, June 2019.

[12] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, “Unsupervised wireless spectrum anomaly detection with interpretable features,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 637–647, 2019.

[13] S. Rajendran, V. Lenders, W. Meert, and S. Pollin, “Crowdsourced wireless spectrum anomaly detection,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 694–703, 2020.

[14] W. Honghao, J. Yunfeng, and W. Lei, “Spectrum anomalies autonomous detection in cognitive radio using hidden markov models,” in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2015, pp. 388–392.

[15] C. Ge, Z. Wang, and X. Zhang, “Robust long-term spectrum prediction with missing values and sparse anomalies,” *IEEE Access*, vol. 7, pp. 16 655–16 664, 2019.

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

[17] Q. Feng, Y. Zhang, C. Li, Z. Dou, and J. Wang, “Anomaly detection of spectrum in wireless communication via deep auto-encoders,” *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3161–3178, Jul 2017. [Online]. Available: https://doi.org/10.1007/s11227-017-2017-7

[18] N. Tandiya, A. Jauhar, V. Marojevic, and J. H. Reed, “Deep predictive coding neural network for rf anomaly detection in wireless networks,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018,

pp. 1–6.

[19] A. Toma, A. Krayani, M. Farrukh, H. Qi, L. Marcenaro, Y. Gao, and C. S. Regazzoni, “Ai-based abnormality detection at the phy-layer of cognitive radio by learning generative models,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 21–34, 2020.

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

[21] A. Asperti, “Sparsity in variational autoencoders,” *CoRR*, vol. abs/1812.07238, 2018. [Online]. Available: http://arxiv.org/abs/1812.07238

[22] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, “Static force field representation of environments based on agents' nonlinear motions,” *EURASIP Journal on Advances in Signal Processing*, vol. 2017, 01 2017.

[23] I. J. Sledge and J. M. Keller, “Growing neural gas for temporal clustering,” in *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[24] E. A. Wan and R. Van Der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[25] G. Slavic, D. Campo, M. Baydoun, P. Marin, D. Martin, L. Marcenaro, and C. Regazzoni, “Anomaly detection in video data based on probabilistic latent space models,” in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2020, pp. 1–8.

[26] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, “Learning Switching Models for Abnormality Detection for Autonomous Driving,” in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 2606–2613.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, p. 264–323, Sep. 1999. [Online]. Available: https://doi.org/10.1145/331499.331504

[28] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. Gomez, and C. Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," 10 2019.

[29] B. Fritzke, "A self-organizing network that can follow non-stationary distributions," in *Artificial Neural Networks — ICANN'97*, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 613–618.

Chapter 7

Discussion and Future Work

7.1 Introduction

In this research work, a joint framework based on deep learning and probabilistic models has been investigated, dissected, and developed to capture abnormalities in the CR-IoT network spectrum. The developed method unfolds the facility to introduce SA capabilities into the cognitive devices of a network. Accordingly, we proposed a data-driven scheme for the cognitive devices exposed to different data dimensionalities in the network to realize autonomous CR-IoT network and eventually detect abnormalities. Fig.7.1 depicts the overview of the implementation of the proposed methods in the case of low and high dimension data applications.

7.2 Low dimensions AI-enabled CR-IoT application

As mentioned and discussed in the **chapter.4**, we commenced to investigate jammer detection problem with low dimension data and proposed a DBN method to capture abnormalities in a CR-IoT Spectrum. DBN network is capable of modeling not only linear and non-linear system dynamics to the associated discrete switching variables, allowing filtering with the Bayesian framework at different levels by using either combi-

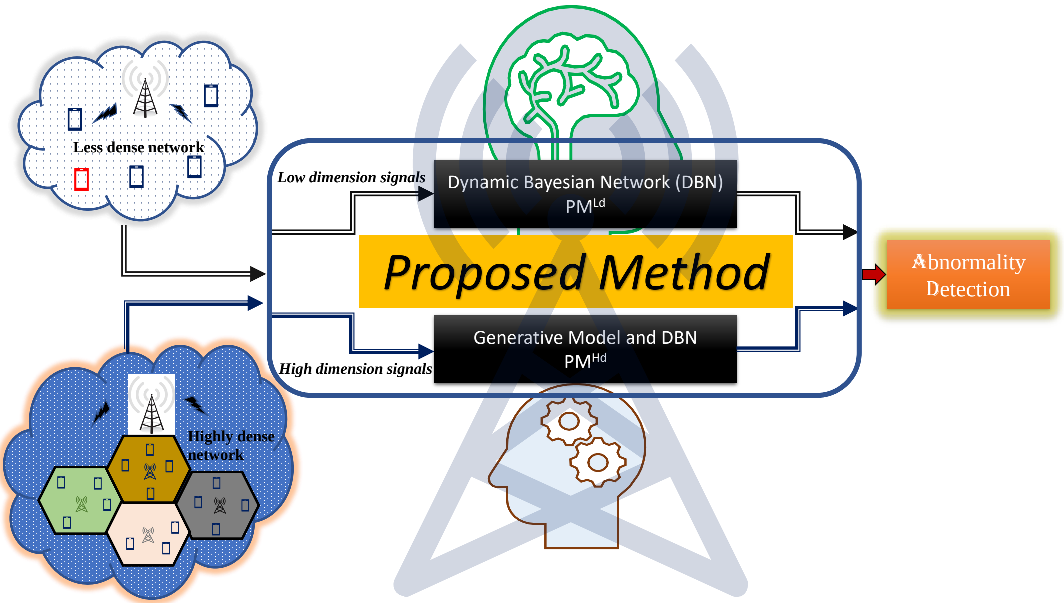


Figure 7.1: Method proposed in this thesis work.

nation of the Kalman filter and Particle filter (for linear problems) or modified Kalman filter together with Particle filter (for scenarios dealing with non-linear problems). The strength of a proposed approach lies in the fact that it provides inferences of the spectrum at a lower (continuous) level and a higher (discrete) level. Subsequently, jammer signals are detected at two levels. To explain the concept, refer to Fig.7.1. If a cognitive device is exposed to low-dimension data, i-e, operating in a less dense network using few subcarriers, it automatically switches to deploy the DBN model perform state estimation and detect abnormalities. We refer the low dimension method as PM^{Ld}, where Ld represents a low dimension.

We compare the performance of DBN with the conventional energy detector (ED) [1],[2] as shown in Fig.7.2 at discrete level and Fig.7.3 at continuous level under multiple jammer attacks. Therefore, it is quite evident that PM^{Ld} outperforms the traditional ED and significantly detect the attacks.

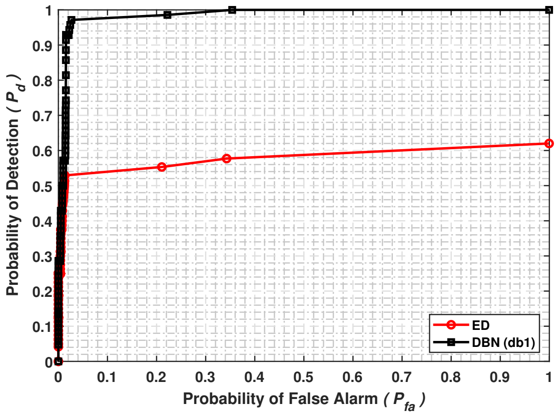


Figure 7.2: Performance evaluation of the proposed PM^{Ld} with ED at discrete level

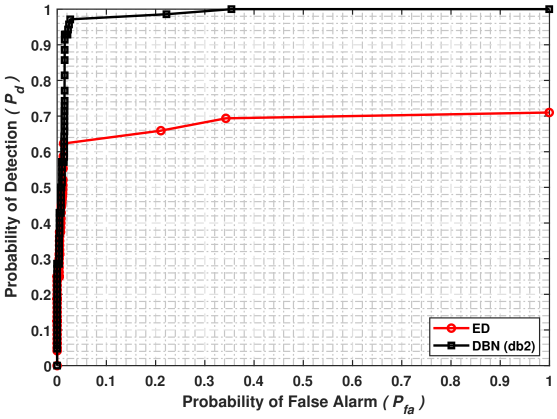


Figure 7.3: Performance evaluation of the proposed PM^{Ld} with ED at continuous level

7.2.1 Limitations

Undoubtedly, the PM^{Ld} has been performing well in detecting jammer attacks in a network. Nevertheless, DBN meets some limitations, such as the model becoming complex and intractable due to many switching variables generation to represent high dimension data. As a result, DBN learning will require a longer time and produce a vast amount of learned cluster features. Moreover, the parallel-DBN system also exhibits certain limitations for high-dimension data because several DBNs models will be involved in learning models for each carrier frequency and produce substantial cluster features. Hence, the parallel-DBN model requires a longer time for training and ultimately consume more resources of the system. Therefore, to handle high dimension data, we propose to deploy deep learning method to deal with high dimensional data and obtain compact latent representations of the spectrum signal. Such latent representations are then clustered, and the DBN model is learned based on low dimension latent space and eventually capture abnormalities in the CR-IoT network.

7.3 High dimensions AI-enabled CR-IoT application

When cognitive objects are being operated in a dense environment in CR-IoT networks, in this case, implementation of PM^{Ld} is somewhat limited as devices will be generating and exchanging massive volume of data as mentioned in **section.7.2.1**. Therefore, we take advantage of the DL (VAE) model discussed in **chapter.6** to obtain a low and compact representation of high dimensional data. VAE also facilitates in providing Bayesian inferences of input data. We deploy two VAEs (for signal and its derivative). We have also introduced an activation regularizer in the second VAE (when input is derivative) to provide distinct derivative latent spaces associated with different dynamic situations. After deploying VAE to acquire low and compact distinct latent spaces (using activation regularizer), we perform state estimation to predict the future state and capture any abnormal behavior that does not follow communication rules learned during the training process. Precisely, we deploy A-MJPF, which facilitates to infer the spectrum evolution by employing PF at a discrete level and modified KF at a continuous level. The abnormality (jammer attacks) is detected at the continuous layer by measuring the probabilistic distance between observation and predicted signal values. We refer to the higher dimension method as PM^{Hd}, where Hd describes high dimension as shown in Fig.7.1. We present a comparison of PM^{Hd} with the model-driven approaches such as conventional ED [2], maximum eigen value detection (MED) [3], and data-driven technique CNN [4] for HetNet1 and HetNet2 under jamming threats as shown in Fig.7.4 and Fig.7.5 respectively. We consider the same configuration and parameters to set-up a fair comparison among detecting techniques.

The data-driven methods exhibit good performance compared to the model-driven technique because data-driven approaches exploit and discover hidden patterns from the complex high dimension input data automatically by taking advantage of deep neural network architecture. Nearly data-driven models utilize the dense and deep neural network to train the model based on applied input data and provide promising results. It can be deduced from the Fig.7.4 and Fig.7.5 that the proposed method outperforms not only the data-driven models but also conventional model-driven approaches to detect jammer attacks in the high dimension OFDM modulated radio signals in the CR-IoT network.

The reason lies in the fact that the proposed method PM^{Hd} deploys a Layered-RVAE structure with an activation regularizer of 0.8 to learn distinct latent spaces. After that, latent space has been clustered using GNG clustering algorithm with optimized parameters selection, such as (threshold values Fig.6.11(a)and an optimal number of clusters Fig.6.11(b)). GNG clustering algorithm has been shown to perform better than SOM, as discussed in **section.6.5.1**. To improved and facilitate prediction for continuous predictive models, RRN is learned as well for each cluster. Finally, we use A-MJPF to perform state estimation and capture abnormalities during the testing phase. We have shown in **chapter.6** that PM^{Hd} capture abnormalities that either due to jammer attacks or unknown sources appear in a network.

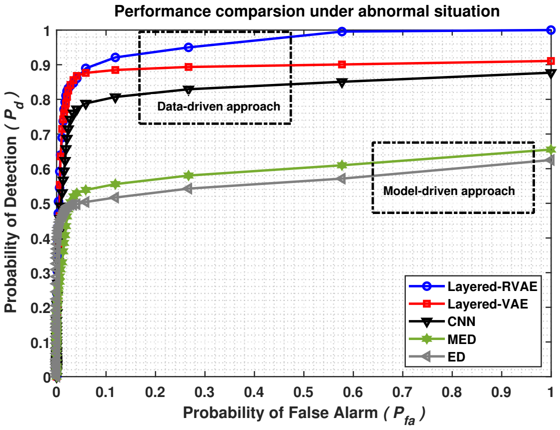


Figure 7.4: Performance comparison between data-driven and model-driven approaches when HetNet1 under jamming attacks.

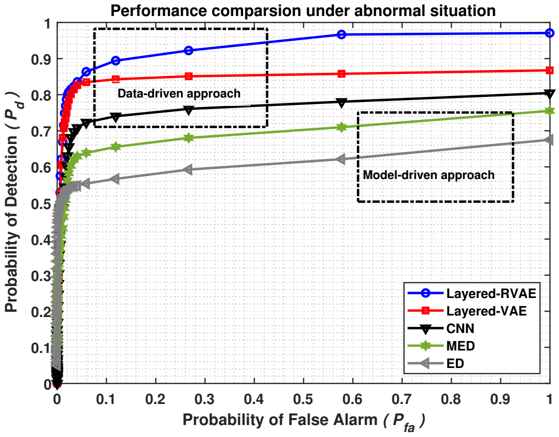


Figure 7.5: Performance comparison between data-driven and model-driven approaches when HetNet2 under jamming attacks.

7.3.1 Limitations

The PM^{Hd} deals with high dimensional data effectively, performs state estimation, and eventually captures abnormalities. However, the method is developed and investigated at the latent space level. Under anomalies, when high dimension data is transformed into a low dimension, we can't explicitly locate the exact location of jammer attacks in the spectrum. Therefore, to identify and trace jammer attacks, latent space needs to be projected back at the spectrum level or, in other words, reconstruct the decoder output and compare it with the input. Such work will facilitate navigating jammer attacks across individual subcarriers. This information can be used by the resource allocation unit in a transmitter to deactivate subcarriers jammed or change transmission strategies.

7.4 Future work and directions

The proposed work in this thesis work has provided a road map to build autonomous radio devices equipped with SA capabilities. The next-generation wireless network will exhibit a higher level of intelligence to provide services to end-users. More specifically, new emerging networks will be more smart, cognitive, self-aware, and intelligent. Moreover, devices in such a network will be more conscious. Undoubtedly, our work facilitates to bring intelligence with SA ability into CR devices to build autonomous CR-IoT network and furnish a way to make CR-IoT network secure by detecting abnormalities in the spectrum. The future work will address abnormality detection at the signal level by mapping latent spaces back to the full spectrum. Since we introduce activation regularizer to obtain distinct latent spaces corresponding to different scenarios in our work, more in-depth investigation will be carried out to evaluate the effect of ρ on signal reconstruction and, eventually, detecting and locating jammer attacks in the CR-IoT spectrum. Nevertheless, the investigated work in this research journey can be extended into several directions. We now highlight the remarkable ideas with some illustrations to carry out and extend the developed method.

- **Realizing coupled Layered-RVAE structure:** We can realize coupled Layered-RVAE implementation as shown in Fig.7.6 The latent space z' from the second VAE is coupled with the latent space z of the first VAE, and then the DBN model is learned based on a generalized state vector. After training, A-MJPF can be deployed to perform state estimation and detect abnormalities.

- **Incorporating LSTM into Layered-RVAE encoder and decoder:** Long-Short Term Memory (LSTM) has been a popular network to perform prediction for time-series data. Recently, the LSTM network is shown to achieve good performance when implemented inside the encoder and decoder architecture of the VAE [5]. Hence, we can modify the Layered-RVAE encoder and decoder and incorporate LSTM inside the encoder and decoder nets of the VAE as shown in Fig.7.7. Afterward, DBN can be deployed for spectrum inferences and capturing abnormal behaviors.

- **Abnormalities detection at higher abstract level:** Since DBN provides inferences at different levels, i-e, continuous and discrete layers, hence, abnormalities can be detected at a higher abstract level as well. As mentioned earlier in **chapter.4**, we obtain a discrete region of DBN by using a clustering algorithm. Therefore, the abnormality can be determined by measuring the data point's probabilistic distance from the centre of the cluster. If the distance is low, it indicates that the data sample belongs to the same cluster class, and if not, it demonstrates that abnormal sample appears in normal data.

- **Considering jammer with different modulated signals:** To address and deal with more realistic and dynamic situations in which jammer changes its transmission strategies such as modulation technique to launch its attack inside the spectrum, the Layered-RVAE can be implemented to discover special jammer attacks based on different modulation techniques. In this work, we have shown that the Layered-RVAE is capable of detecting jammer with varying power in **section.6.7.1**. Hence, more detailed characteristics of jammer signals can be detected with the proposed method at latent space level, and at signal level by projecting back latent spaces to full spectrum scale.

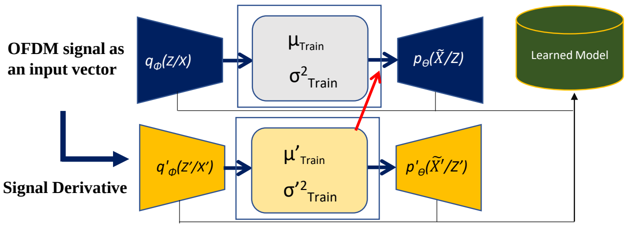


Figure 7.6: Coupled Layered-RAVE structure.

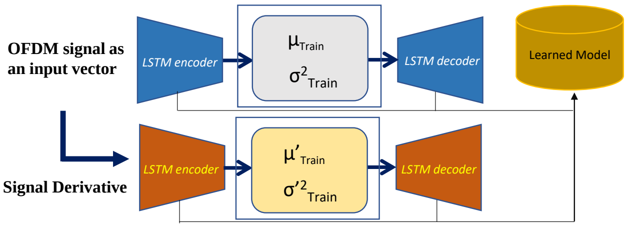


Figure 7.7: Layered-RAVE with LSTM structure.

Hence, the developed method in this work has opened directions to bring SA into the CR-IoT network. With the further in-depth realization of the proposed method, the CR-IoT network can be emerged as more autonomous, intelligent and cognitive. The developed method is not only limited to deployment in CR-IoT network, but rather it can be deployed into emerging technologies such as Unmanned Aerial Vehicle (UAVs), V2V, self-driving cars, and 6G mobile networks. For UAV applications, the proposed method can help develop secure transmission among multiple drones communicating with on-ground vehicles and several base stations. It can also facilitate V2V technology to provide reliable communication for navigation and particular guidance assistance purposes under emergencies to vehicles. Moreover, PM^{Hd} can enable the 6G mobile network to develop an intelligent and secure network and furnish various applications to the end-users.

References

[1] Z. Xuping and P. Jianguo, "Energy-detection based spectrum sensing for cognitive radio," in *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, 2007, pp. 944–947.

[2] M. Z. Alom, T. K. Godder, M. N. Morshed, and A. Maali, "Enhanced spectrum sensing based on energy detection in cognitive radio network using adaptive threshold," in *2017 International Conference on Networking, Systems and Security (NSysS)*, 2017, pp. 138–143.

[3] Y. Zeng, C. L. Koh, and Y.-C. Liang, "Maximum eigenvalue detection: Theory and application," in *2008 IEEE international conference on communications.* IEEE, 2008, pp. 4160–4164.

[4] D. Han, G. C. Sobabe, C. Zhang, X. Bai, Z. Wang, S. Liu, and B. Guo, "Spectrum sensing for cognitive radio based on convolution neural network," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–6.

[5] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using vae-lstm hybrid model," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4322–4326.

Appendix A

Algorithms

1. **Algorithm 1:** MJPF that is implemented in chapter.4 (section.4.5).

2. **Algorithm 2:** Describes the training of deep learning models using FFT and CWT images to classify jammer signals in the spectrum used on chapter.5 (section.5.4.1.1).

3. **Algorithm 3:** Presents testing steps followed after deep models have been trained and tested with test data set described in chapter.5 (section.5.4.1.1).

4. **Algorithm 4:** Describes the training of Layered-RVAE and DBN learning steps implemented in chapter.6 (section.6.4.1).

5. **Algorithm 5:** Describes the testing of Layered-RVAE by using A-MJPF implemented in chapter.6 (section.6.4.2).

Algorithm 1: MJPF Algorithm

```
Input $M^{(S))}$,$Q^{(S))}$,$R^{(S))}$ $TM$, $N$ Total particles ;
$Z_t \leftarrow$ Observation or measurements from the spectrum. t = 1,2,3..T
for $t$ = 1 to $T \leftarrow$ Time evolution do
    for $n$ = 1 to $N \leftarrow$ Particles do
        $w_n = \frac{1}{N} \leftarrow$ Particles weight
        if t==1 ← initial state then
            Sample $X_1$ from $P(X_1)$
            $X_t = X_1 \leftarrow$ current state
            estimate $S_t^*$ from $P(X_t \mid S_t)$
        else
            Predict $S_t^*$ by using $TM$
            $X_t = X_{t-1} \leftarrow$ current state
        Calculate $d(X_t, M^{(S_{t-1})}) \leftarrow$ euclidean distance
        if if $1 - \frac{d(X_t,M^{(S_{t-1})})}{R^{(S)}} < 0 \leftarrow$ outside the model then
            $U_{S_{t-1}} = 0$ and $P_{t-1|t-1} = R_t \leftarrow$ Process noise
        else
            $U_{S_{t-1}} = U_{S_{t-1}^*}$ and $P_{t-1|t-1} = Q_{S_{t-1}^*}$
        Prediction:
        $X_t = AX_{t-1} + BU_{S_{t-1}} \leftarrow$ state
        $P_{t|t-1} = AP_{t-1|t-1}A^t + \sigma_{t-1} \leftarrow$ covariance
        $Z_t = (Z_t - H_tX_t)$
        $K_t = P_{t|t-1}H_t^T(HP_{t|t-1} + R_t)^{-1}$
        Update
        $X_t = X_t + KZ_t \leftarrow$ updated state
        $P_{t|t} = 1 - K_tH_tP_{t|t-1} \leftarrow$ updated covariance
        Calculate abnormality signals $db_1$,$db_2$
        $w_n = \frac{w_n}{db_1+db_2} \leftarrow$ update weights
    SIR re-sampling
Output $db_1$,$db_2$
```

Algorithm 2: Training of Deep Learning Models

```
1. Begin;
2. Training of Deep Learning Models ;
3. Model Parameters: Learning Rate 0.0001, Batch Size=64, Learning Method
   ADAM, $\eta = [0, 1]$, 0 for FFT-based images and 1 for CWT-based images;
4. Receive OFDM modulated signal $Y$;
5. Obtain Time-Frequency representation of the $Y$ using:
   a) FFT transform to get complex data samples $Y_{FFT}$
   b) CWT transform to get scalogram coefficients $Y_{CWT}$ ;
6. Transform $Y_{FFT}$ and $Y_{CWT}$ data into RGB images and save as two separate
   Data sets D = [$Y_{FFT}$,$Y_{CWT}$];
7. Divide data set into training and testing set;
TRAINING ;
while for all train set images do
    while $\eta = 0$ do
        for $i$ = 1 to $I$ do
            -Feed the AlexNet and GoogLeNet model with FFT-based images
            - Train models
            Output: Trained models for FFT-based images
    while $\eta = 1$ do
        for $k$ = 1 to $N$ do
            -Feed the AlexNet and GoogLeNet model with CWT-based images
            -Train models
            Output: Trained models for CWT-based images
Result: Trained Models
```

```
Algorithm 3: Testing of trained deep learning models
1. Begin;
2. Jammer Signal Classification;
3. Trained AlexNet and GoogLeNet models, Test data set (containing FFT and
   CWT images);
TESTING;
while for all test set images do
    while $\eta = 0$ do
        for $i = m$ to $G$ do
            -Feed the FFT-based test images to the trained AlexNet and
             GoogLeNet models
            -Decision ← Classification Normal Signal, RJHP, RJLP
    while $\eta = 1$ do
        for $p = 1$ to $P$ do
            -Feed the CWT-based test images to the trained AlexNet and
             GoogLeNet models
            -Decision ← Classification Normal Signal, RJHP, RJLP
Result: Jammer signal classification
```

Algorithm 4: Training of Layered-RVAE and DBN learning

```
1. Initialize: Learning Rate 0.00001, Batch size = 32, VAE $\in [VAE_1,VAE_2]$
   ADAM optimizer,activation regularizer $\rho$ =[0.1,0.4,0.6,0.8,1.0], latent space =
   30, number of clusters = 12 ;
2. From the received high dimension OFDM signal r(t) Obtain:
     a) $\widetilde{\gamma}$ an input vector for the $VAE_1$    b) $\tilde{\widetilde{\gamma}}$ a derivative input vector for the
   $VAE_2$ ;
3.Train $VAE_1$ on input vector $\widetilde{\gamma}$;
$\phi_{Encoder^1}$, $\theta_{Decoder^1}$ $\leftarrow$ Initialize network parameter
while $itr < Max\ iteration$ do
    $\chi \leftarrow$ random-mini bact from Dataset, where $\chi \in \widetilde{\gamma}$
    $z \leftarrow Encoder^1_\phi(\chi)$
    $loss_{prior} \leftarrow D_{kl}(q(z \mid x) \parallel p(z))$
    $\widetilde{\chi} \leftarrow Decoder^1(z)$
    Update parameters
    $\phi_{Encoder^1} \leftarrow \bigtriangledown\phi_{Encoder^1}$ : using ADAM optimizer
    $\theta_{Decoder^1} \leftarrow \bigtriangledown\theta_{Decoder^1}$ : using ADAM optimizer
4.Train $VAE_2$ on input vector $\tilde{\widetilde{\gamma}}$;
$\phi_{Encoder^2}$, $\theta_{Decoder^2}$ $\leftarrow$ Initialize network parameter
while $itr < Max\ iteration$ do
    $\chi \leftarrow$ random-mini bact from Dataset, where $\chi \in \tilde{\widetilde{\gamma}}$
    $z' \leftarrow Encoder^2_\phi(\chi) \leftarrow \rho$
    $loss_{prior} \leftarrow D_{kl}(q($z' $\mid x) \parallel p(z))$
    $\widetilde{\chi} \leftarrow Decoder^2(z')$
    Update parameters
    $\phi_{Encoder^2} \leftarrow \bigtriangledown\phi_{Encoder^2}$ : using ADAM optimizer
    $\theta_{Decoder^2} \leftarrow \bigtriangledown\theta_{Decoder^2}$ : using ADAM optimizer
5.$Z_{Train} = [z_k, \dot{z_k}]^T \leftarrow$ Generalize state vector
6.$M^{(S)}$,$Q^{(S)}$,$R^{(S)}$ and $RNN^{(S)} \leftarrow$ Learned cluster features
Output: Trained Layered-RVAE model and learned features of $Z_{Train}$
```

Algorithm 5: Layered-RVAE testing and A-MJPF

```
1. Input Trained Layered-RVAE, and Learned DBN features
   $M^{(S))}$,$Q^{(S))}$,$R^{(S))}$,$RNN^{(S))}$, $TM$, $N$ Total particles;
$Z_{Test} = [z_k, \dot{z_k}]^T \leftarrow$ From trained Layered-RVAE
for $k$ = 1 to $T \leftarrow$ Time evolution do
    for $n$ = 1 to $N \leftarrow$ Particles do
        $w_n = \frac{1}{N} \leftarrow$ Particles weight
        if $k$==1 $\leftarrow$ initial state then
            Sample $Z_1$ from $P(Z_1)$
            $Z_k = Z_1 \leftarrow$ current state
            estimate $S_k^*$ from $P(Z_k \mid S_k)$
        else
            Predict $S_k^*$ by using $TM$
            $Z_k = Z_{k-1} \leftarrow$ current state
        Calculate $d(Z_k, M^{(S_{k-1})}) \leftarrow$ euclidean distance
        if if $1 - \frac{d(Z_k, M^{(S_{k-1})})}{R^{(S)}} < 0 \leftarrow$ outside the model then
            $U_{S_{k-1}} = 0$ and $P_{k-1|k-1} = R_k \leftarrow$ Process noise
        else
            $U_{S_{k-1}} = U_{S^*_{k-1}}$ and $P_{k-1|k-1} = Q_{S^*_{k-1}}$
        Prediction:
        $Z^j_{k+1} = AZ^j_k + BRNN^{(S)}(U^j_k) + w^j_k \leftarrow$ state
        $Z_{k+1|k} = \sum_{j=0}^{2D} W^{j,m} Z_{k+1|k}$
        $P_{k+1|k} = \sum_{j=0}^{2D} W^{j,c} \left\{ Z^j_{k+1} - Z^j_{k+1|k} \right\} \left\{ Z^j_{k+1} - Z_{k+1|k} \right\}^T$
        $P^D_{k+1|k} = P_{k+1|k}$ Update:
        $K_{k+1} = [P^D_{k+1}; I_D]_t (P^D_{t|k-1} + \Sigma_{k+1})^{-1}$
        $Z_{k+1|k+1} = Z_{k+1|k} + K_{k+1}(u_{k+1} - u_{k+1|k})$
        $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}(P^D_{k+1|k} + \Sigma_{K+1})K^T_{k+1}$
        $db2 = -\ln \int \sqrt{p(Z^*_k|Z^*_{k-1}(S^*_k))p(X_k|Z^*_k)}dZ^*_k \leftarrow$ Abnormality
         Measurement
        $w_n = \frac{w_n}{db_1 + db_2}$
    SIR resampling
Output Abnormality signals $db_2$
```