ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ARTIFICIAL INTELLIGENCE BASED DETECTION SCHEMES FOR SECURE WIRELESS COMMUNICATION

M.Sc. THESIS

Selen GEÇGEL

Department of Electronics and Communication Engineering

Telecommunication Engineering Programme

May 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ARTIFICIAL INTELLIGENCE BASED DETECTION SCHEMES FOR SECURE WIRELESS COMMUNICATION

M.Sc. THESIS

Selen GEÇGEL
(504161330)

Department of Electronics and Communication Engineering

Telecommunication Engineering Programme

Thesis Advisor: Prof. Güneş Zeynep KARABULUT KURT

May 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİ TELSİZ İLETİŞİM SAĞLANMASINA YÖNELİK YAPAY ZEKA TABANLI SINIFLANDIRMA METOTLARI

YÜKSEK LİSANS TEZİ

Selen GEÇGEL
(504161330)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomunikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Güneş Zeynep KARABULUT KURT

MAYIS 2019

Selen GEÇGEL, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161330 successfully defended the thesis entitled "ARTIFICIAL INTELLIGENCE BASED DETECTION SCHEMES FOR SECURE WIRELESS COMMUNICATION", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Güneş Zeynep KARABULUT KURT** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Ender Mete EKŞİOĞLU** Istanbul Technical University	.....
	Assoc. Prof. Ertuğrul BAŞAR Koç University	.....

Date of Submission : **3 May 2019**
Date of Defense : **29 May 2019**

To my family,

FOREWORD

First of all, I would like to express my deep and sincere gratitude to my counselor Prof. Güneş Zeynep Kurt for her guidance. Throughout this period, she gave insights to build a road map for my own destination and to learn critical skills required. I am thankful to my family being there all the time. Additionally, I would like to thank my dear friends Ramazan Çetin, Ozan Alp Topal, Caner Göztepe, Mehmet Can and Halil Durmuş for their precious support.

May 2019

Selen GEÇGEL
(Electronics and Communication Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ADAM : Adaptive Moment Estimation
ANN : Artificial Neural Network
ALRT : Average Likelihood Ratio Test
AWGN : Additive White Gaussian Noise
BER : Bit Error Rate
BJ : Barrage Jamming
eNB : Base Station
CFO : Carrier Frequency Offset
CWT : Choi-Williams Transform
CV : Computer Vision
CP : Cyclic Prefix
DCNN : Deep Convolutional Neural Network
DRNN : Deep Recurrent Neural Network
DoS : Denial of Service
EVM : Error Vector Magnitude
FDD : Frequency Division Duplex
GLRT : Generalized Likelihood Ratio Test
GAN : Generative Adversarial Network
i.i.d : Independent Identically Distributed
k-NN : K Nearest Neighborhood
LSTM : Long Short-Term Memory
LTE : Long Term Evolution
LTE-A : Long Term Evolution Advanced
MAC : Media Access Control
MIMO : Multiple Input Multiple Output
ML : Maximum Likelihood
MN : Monitoring Node
NLOS : Nonline of Sight
NOMA : Nanorthogonal Multiple Access
OFDM : Orthogonal Frequency Division Multiplexing
OFDMA : Orthogonal Frequency Division Multiple Access
pdf : Probability Distribution Function
PDR : Packet Delivery Ratio
PSS : Primary Synchronization Signal
PSR : Packet Send Ratio
QPSK : Quadrature Phase Shift Keying
ReLu : Rectified Linear Unit
RSJ : Reference Signal Jamming

RSSI : Received Signal Strength Indicator
SSS : Secondary Synchronization Signal
STFT : Short Time Fourier Transform
SJR : Signal To Jamming Ratio
SINR : Signal To Interference Plus Noise Ratio
SNR : Signal To Noise Ratio
SVM : Support Vector Machine
SSJ : Synchronization Signal Jamming
TDD : Time Division Duplex
TFR : Time-frequency Representation
TFT : Time-frequency Transformation
UE : User Equipment
USRP : Universal Software Radio Peripheral
WVT : Wigner-Ville transformation
ZC : Zadoff Chu

SYMBOLS

$\lvert\cdot\rvert$	**:** Absolute Value
$\mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{s}}^{\mathbf{2}})$	**:** Complex Gaussian Distribution with 0 mean and σ^2 variance
E[$\cdot$**]**	**:** Expectation Operator
Var[$\cdot$**]**	**:** Variance Operator
$\mathbf{H}$	**:** Channel Matrix
$\mathbf{\Gamma}$	**:** Gamma Function
mod $(\cdot, \cdot)$	**:** Modulo Operator
$\nabla(\cdot)$	**:** Partial Derivation Operator

LIST OF TABLES

Page

LIST OF FIGURES

Page

ARTIFICIAL INTELLIGENCE BASED DETECTION SCHEMES FOR SECURE WIRELESS COMMUNICATION

SUMMARY

In this thesis, machine learning algorithms, especially deep learning methods, is aimed for the solution of the signal detection and identification problems encountered frequently in communication systems. In this context, the detection and identification of attack signals were investigated to provide maintenance of secure communication systems.

In the first phase of the study, wide and narrow band jamming attacks are considered against to Long Term Evolution (LTE) systems. As a wide band jamming attack, Barrage jamming (BJ) signals which attack to whole band are generated. It is prevalent and optimal attack type. As a narrow band jamming attack, two different attack types are built: synchronization signal jamming (SSJ) and reference signal jamming (RSJ). They jam reference or synchronization symbols in the transmitted LTE frame by consuming low power so they are called as smart jammers. The dataset is expanded by including the case for the absence of attacks.

Choi-Williams, short time Fourier and Gabor wavelet time-frequency transform methods are applied to reveal the characteristic pattern of the different jamming signals. The attack signals are shown in the time-frequency domain. In this way, the new neighboring relations are caught between time and frequency components of signals. Deep convolutional neural networks (DCNN) are used to identify pre-processed jamming signals. The performance of the classifier based on deep learning methods is compared with the support vector machines (SVM) classifier which is training the features are extracted with principal component analysis.

DCNN classifier shows superior performance compared to the conventional methods based classifier. Also within the scope of the study, Gabor wavelet transform remarkably indicates characteristics of attack signals compared to other time-frequency transformations. According to obtained results, presented two classifiers determine the types of attack signals with high success rates when jamming signals are pre-processed with wavelet transform.

In the second phase of the study, the previous study is enlarged by considering different attack types, the presence of jammer and transmission. Also, the success of deep learning methods on real-time signal data is investigated. In this context, wide and narrow band jammers are implemented on software defined radios. Each data set has 6 different classes which are built by considering presence/absence of the signal transmission with/without the existence of the two types jamming signals. The data sets are extended for different position of the attacker and transmitter.

The complex jamming signals are transferred to the time-frequency domain with complex Gaussian wavelet and short-time frequency transformations. As a result of the pre-processing, the signal data are examined with different window widths in the

frequency domain and the characteristics of the attacks are extracted. DCNN has shown exceptionally better performance than the other shallow learning methods on simulation data set. Therefore, in this study, DCNN and deep recurrent neural networks (DRNN) are compared in terms of classification performance of signals. All models regarding these artificial neural network (ANN) architectures are trained using the data set which are gathered using software defined radios (SDRs). Models are optimized using different techniques and their parameters are tuned. As result of the study, the DCNN architecture shows higher performance than DRNN.

In this thesis, firstly, a detection scheme based on deep learning methods and time-frequency transformation for the detection and identification of jamming signals is proposed. The proposed deep convolutional neural network architecture is convenient to determine the type and existence of the jammer attacks on communication systems. It is shown using simulation and real-time signal dataset.

GÜVENLİ TELSİZ İLETİŞİM SAĞLANMASINA YÖNELİK YAPAY ZEKA TABANLI SINIFLANDIRMA METOTLARI

ÖZET

Sinyal tespiti ve sinyallerin türünün belirlenmesi çeşitli disiplinler altında yaygın olarak ele alınan önemli bir konudur. Haberleşme sistemlerinde, bu konu sistemlerin güvenliğinin sağlanması veya performansının iyileştirilmesi amacıyla incelenmektedir.

Literatürde ki çalışmalar incelendiğinde önerilen şema ve yöntemlerin çoğunlukla sinyalin istatiksel özelliklerine dayandırıldığı görülmektedir. Bu durum sinyalin türü ve istatiksel özellikleri ile ilgili varsayımlar yapma ihtiyacını doğurmaktadır ve sistemlerin değişen çevre koşullarına uyumlu olmasını güçleştirmektedir. İlaveten, önerilen çözümler amaçlanan sistemler üzerinde yüksek başarım göstersede başka sistemlerde uygulanabilirliği düşüktür.

Haberleşme sistemlerinde çevre koşulları hızla değişmektedir. Bu değişimi kimi zaman kasıtlı kimi zaman kasıtsız saldırılar sebebiyle oluşmaktadır. Yazılım tabanlı radyoların yaygınlaşması kasıtlı saldıraların artmasına neden olduğu gibi bu saldırıların çeşitlilik kazanmasına ve tespitinin zorlaşmasına sebep olmuştur. Bu durum problemlere mutlak çözüm önerileri sunmayı güçleştirmekte ve esnek çözüm şemalarını gerektirmektedir. Son yıllarda yüksek başarım sağlaması sebebiyle yapay zeka teknikleri pek çok disipline uyarlanmaktadır. Yapay zeka tekniklerinin literatürde bu kadar geniş yer tutmasının bir sebebide değişen koşullara karşı yeni çözümler sunmasıdır.

Bu tez çalışmasında, iletişim sistemlerinde sıklıkla karşılaşılan işaret tespit problemlerinin çözümüne yönelik makine öğrenmesi algoritmaları, özellikle derin öğrenme metotları araştırılmıştır. Bu kapsamda, iletişim sistemlerinin güvenliğini tehdit eden farklı sinyal bozucu saldırılarının tespiti ve türünün belirlenmesi için derin öğrenme tabanlı yapay sinir ağı mimarilerinin kullanımı önerilmiştir.

Çalışmanın ilk aşamasında, 4.nesil uzun vadeli evrim sistemlerine yönelik farklı işaret bozucu saldırıları göz önünde bulundurulmuştur. Literatürde baraj bozucu olarak yer alan, yaygın işaret bozucu türünün ve güvenli iletişimin olduğu durumlar değerlendirilmiştir. Akıllı işaret bozucular olarak da adlandırılan, uzun vadeli evrim teknolojisinin işaret çerçevesindeki referans veya senkronizasyon sembollerine saldıran dar bantlı bozucular dahil edilerek veri seti genişletilmiştir.

Farklı saldırılar altında alınan kompleks işaret verilerinin taşıdıklıkları karakteristik özellikleri ortaya çıkartmak maksadıyla önerilen şemanın ön işleme bloğunda Choi-Williams, kısa zamanlı Fourier ve Gabor dalgacık zaman-frekans dönüşüm yöntemleri uygulanmıştır. Bu sayede atak işaretleri zaman ve frekans eksenlerinde gösterilmiştir. Her iki domaine ilişkin özellikler ve bu özellikler arasındaki komşuluk ilişkileri ortaya çıkarılmıştır.

Zaman-frekans dönüşümleri ile işlenen verilerin tespiti ve ayırt edilmesi amacı ile ’derin evrişimsel sinir ağları’ ve ’destek vektör makinesi’ öğrenme metotları

kullanılmıştır. Destek vektör makinesi yönteminin kendiliğinden özellik çıkarma yetisi olmadığı için sınıflandırma işleminden önce 'temel bileşen analizi' ile özellik çıkarımı yapılmıştır. Derin öğrenme metotlarına dayalı sınıflandırıcının performansı, temel bileşen analizi ve destek vektör makineleri kullanılarak oluşturulan sınıflandırıcı ile karşılaştırılmıştır.

Derin evrişimsel sinir ağı sınıflandırıcısı geleneksel metotlarla oluşturulan sınıflandırıcıya nazaran üstün başarım göstermiştir. Ayrıca çalışma kapsamında dalgacık dönüşümlerinin atak işaretlerinin karekteristik özelliklerini diğer yöntemlere kıyasla daha iyi yansıttığı ve bu metotla oluşturulan veri setinde her iki sınıflandırıcının da atak işaretlerinin türlerini yüksek başarı ile belirlediği gözlenmiştir.

Çalışmanın ikinci aşamasında, atak işaret türünün yanı sıra atak işaretinin varlığının da tespiti göz önünde bulundurularak amaçlanan sistem genişletilmiştir. İlaveten, derin öğrenme metotlarının gerçek zamanlı saldırı işaretlerine ilişkin veriler üzerindeki başarısı araştırılmıştır.

Bu kapsamda, geniş ve dar bantlı işaret bozucular yazılım tabanlı radyolar üzerinde gerçeklenmiştir. İletimin ve farklı türde ki işaret bozucuların, olduğu ve olmadığı durumlar ayrı ayrı göz önünde bulundurularak 6 farklı veri sınıfı oluşturulmuştur. Veri seti işaret bozucunun alıcı ve vericiye olan uzaklığı, vericinin ve işaret bozucunun iletim gücü göz önünde bulundurularak genişletilmiştir.

İşaret verileri kompleks Gauss dalgacık ve kısa zamanlı frekans dönüşümleri ile zaman-frekans domainine taşınmış, işaret verilerine frekans domaninde farklı pencere genişlikleri ile bakılarak ataklara özgün nitelikler çıkarılmıştır. Sınıflandırıcı olarak derin evrişimsel ve derin yinelemeli sinir ağları tercih edilmiş, her iki yöntemin başarısı karşılaştırılmıştır. Çalışma sonucunda derin evrişimsel sinir ağı mimarisi daha yüksek performans göstermiştir.

Tez kapsamında gerçekleştirilen çalışmalarda, ilk olarak farklı işaret bozuculara ilişkin işaret türlerinin tespiti için yapay sinir ağları metotları ve zaman-frekans dönüşüm yöntemlerine dayanan bir yaklaşım önerilmiştir. Sunulan yöntem simülasyonlar ile ile oluşturulan veri setlerinde yüksek başarım göstermiştir.

Bir sonraki çalışmada önerilen derin evrişimsel sinir ağı mimarisinin, iletişim sistemlerine yönelik saldırıların türünün yanı sıra bu sinyallerin varlığının tespitine yönelik olarak da kullanılabileceği araştırılmıştır. İlaveten, bu çalışmada yazılım tabanlı radyolar ile oluşturulan bozucu işaretleri kullanılarak sistemin gerçek zamanlı sistemlere uygunluğu gösterilmiştir. Yapılan çalışmalar sonucunda, tasarlanan derin evrişimsel sinir ağı mimarisinin işaret bozucu tespitinde üstün performans gösterdiği deney ve benzetim ortamında gözlenmiştir.

Bu tez çalışmasının ilk bölümünde, sinyal tespit ve tanımlama problemi tanıtılmış ve problemin araştırılmasındaki motivasyonlardan söz edilmiştir. Ardından literatürdeki çalışmalar detaylı olarak ele alınmış, çalışmaların avantaj ve dezavantajları vurgulanmıştır. Son olarak tez çalışmasının literatüre katkıları açıklanmıştır.

İkinci bölümde, evrişimsel sinir ağları tanıtılmıştır. Evrişimsel sinir ağlarının temel fonksiyonları ve katmanları hakkında kapsamlı bilgi verilmiştir. Bunu takiben, evrişimsel sinir ağlarının eğitiminde kullanılmakta olan geri yayılım algoritması, tez çalışmasında başvurulan optimizasyon ve regülasyon tekniklerinden etraflıca bahsedilmiştir.

Üçüncü bölümde, uzun vadeli evrim sistemlerine farklı tip karıştırıcılar tarafından yapılecek saldırılar ele alınmıştır. İlk olarak uzun vadeli evrim sistemlerinin avantaj ve dezavantajlarından bahsedilip, 4.nesil haberleşme sistemlerinin çerçeve yapısı hakkında bilgi verilmiştir. Ayrıca konu edilen saldırı tipleri ve bu saldırıların temel ilkeleri tanıtılmıştır. Ardından sinyallerin taşıdığı karakteristik özellikleri ortaya çıkarmak amacıyla kullanılan farklı ön işleme metotları açıklannmıştır. Sinyallerin sınıflandırılması ve sistemin değişen çevre koşullarına göre daha dinamik yapıda olması istenerak iki makine öğrenmesi yaklaşımına başvurulmuştur. Bunlar derin evrişisel sinir ağı ve destek vektör makineleri yaklaşımdır. Her iki makine öğrenmesi yönteminin çalışma prensiplerine ilişkin gerekli bilgiler verilmiştir.

Özetle, tez çalışmasının bu bölümünde, verici ve alıcı arasında iletimin olduğu durumda karıştırıcı sinyallerinin varlığının ve türünün tespiti amaçlanıp, önerilen şemaya uygun tekniklerinin bulunması için çeitli metotlar incelenmiştir. Amaçlanan sistemin simülasyon ortamında gerçekleştirilmesine yönelik detaylı bilgilendirme yapılmıştır. Ardından elde edilen sonuçlar kapsamlı olarak yorumlanıp bölüm sonlandırılmıştır.

Tez çalışmasının dördüncü bölümünde, dik frekans bölmeli çoğullama tekniğinin kullanıldığı bir haberleşme sistemine yapılan saldırı tipleri incelenmiştir. Haberleşme sistemi ve karıştırıcı sistemleri, yazılım tabanlı radyolar kullanılarak gerçek zamanlı ortamda gerçeklenmiştir. Alıcı, veric ve karıştırıcı arasında ki farklı uzaklıklar ve farkli sinyal-gürültü oranları için önerilen şemaların başarımı analiz edilmiştir. Bir önceki bölümde detaylı olarak açıklanmakta olan çalışmadan farklı olarak, bu çalışmada verici ile alıcı arasındaki iletimin varlığı da araştırılmıştır. İlaveten, farklı bir dalgacık dönüşümü yönteminin ve makine öğrenmesi metodunun başarımı araştırılmıştır.

Evrişimsel sinir ağları bir başka gelişmiş derin öğrenme yöntemi olan tekrarlayan sinir ağları ile performans bakımından karşılaştırılmıştır. Çalışmada başvurulan tüm yöntemler ve bu yöntemlerin uygulanmasına ilişkin detaylı bilgiler verilmiştir. Son olarak önerilen sistem üzerinden elde edilen sonuçlar paylaşılmıştır. Sonuçlar göstermiştir ki gerçek zamanlı sistemler üzerinden alde edilen sinyallerin tespiti ve ayırt edilmesi hususnda dalgacık dönüşümü yöntemi derin evrişimsel ssinir ağı mimarisi daha başarılıdır.

Sonuç bölümünde, tez kapsamında ele alınan probleme ilişkin önerilen şema ve yöntemler kısaca açıklanmıştır. Bu yöntemlerin uygunluğu ve biarada kullanımına ilişkin genel bir değerlendirme yapılmıştır.

1. INTRODUCTION

Signal detection is the fundamental issue for several research areas such as speech processing, communication systems, control systems, radar and sonar systems, bio-medicine, image processing, and seismology. In a radar system, the important thing is determining the presence or absence of an aircraft. On the other side, in communication systems, the definition of the problem changes as detecting of a transmitted signal. The main objectives of all are sometimes only realizing the presence of the specific signal or identifying the signal, sometimes all of them.

Signal identification approaches can be summarized in two categories. One of these is maximum likelihood (ML) estimation based approaches which are the average likelihood ratio test (ALRT), the generalized likelihood ratio test (GLRT) and hybrid of the two tests. Most of the ML approaches that have appeared in the literature are based on the ALRT due to the analytical difficulties in obtaining an exact solution using this test, only approximations to optimal solutions are possible [1]. These approaches based on decision theory are generally considered for multiple hypothesis test problems and they are proposed to maximize the probability of correct identification. At this point, there are several challenges such as the need for the probability density function, the assumption of independent and identically distributed signals, depending on the hypothesis. We can say that ML approach is sensitive to model mismatches such as phase, frequency, timing offset. As a result of these, the identification performance can degrade significantly because of the imperfections [2].

1.1 Motivations

Artificial neural network (ANN) based solutions show superior performance and surpass human efforts at a variety of disciplines [3–5] during the last years. Especially, deep learning techniques forge ahead of other shallow learning techniques such as support vector machines (SVM), k nearest neighborhood (k-NN) algorithms and they are seen as state of the art technologies. As a striking advantage, its

variations have wide application areas. The most important and prevalent ones are computer vision (CV), voice and natural language processing. The main problem of obsolescent techniques is the addiction to extract a lot of features and a requirement of pre-processing. On the other hand, mostly deep learning techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs) offer critical advantages in terms of these issues besides a higher performance.

DCNN is the most common deep learning algorithm for interdisciplinary research areas under the CV. As in other deep learning techniques, CNNs process the data in like manner to the human brain. When the working principle of CNN is examined more carefully, it extracts the low and high-level features from the data sample as similar to the interaction between humans and eyes. In 1998, Le-Cun et al. proposed LeNet to recognize hand-written numbers on grayscale images. Almost 15 years later in 2012, another milestone for DCNN, AlexNet was proposed with the deeper architecture which has stacked convolutional layers, pooling layers, rectified linear units (ReLu), dropout layers, additionally stochastic gradient descent (SGD) has been used in it. Following these developments, many pioneer CNN architectures [6–13] were presented and each one of them brought in a new aspect to CNN architecture.

Along with these developments, the research studies related to communication systems investigate deep learning techniques. Especially, CNN based approaches become prevalent and a variety on many communication issues such as multiple input multiple output orthogonal frequency-division multiplexing (MIMO-OFDM) [14–16], cognitive radio [17–19], interference control [20–22], traffic classification [23], PAPR reduction [24], routing [25–27].

A lot of solutions based on CNN and its derivatives become a common method in the detection and processing of one and multidimensional signals. In [28], authors present a CNN based blind modulation classification algorithm to identify different types of frequency modulation. In [29], Khan et al. a propose deep neural network based (DNN) modulation identification scheme for digital coherent receivers. In [30], authors compare the traditional machine learning and CNN in terms of prediction time and performance for modulation classification. Also, in [31], they share improvements related to the extended new dataset. In [32], authors use CNN and Choi-Williams time-frequency transform together to recognize the cognitive radio waveform. In

[33], authors provide a new modulation scheme by combining generative adversarial networks (GAN) and a famous CNN architecture AlexNet. In [34] authors propose to improve signal detection performance which is faded by imperfect channel conditions with an adaptive ensemble deep neural network model. The results of the presented approach demonstrate a better reliability performance than the conventional scheme regarding OFDM symbol classification. In [35], Ye and et al. propose to estimate the channel and to detect symbols and they show the ability of deep neural networks to analyze the characteristic of wireless OFDM channels. Their experiment results indicate that deep learning models provide more robustness to the dynamic variety of communication channels.

1.2 Literature Review

Signal identification and detection are critical issues to overcome the security problems of communication systems. Researchers study on these issues directly or indirectly to avoid intentionally or unintentionally adverse effects and to improve the performance of the communication link. These problems are analyzed via various titles such as anomaly detection [36–38], interference control [39–41], mitigating attack signals [42–45] and so forth.

In this thesis, signal detection and identification problems are tackled over a communication system that is contaminated by jamming signals. Jamming attacks can be defined as sending variety signals intentionally for disrupting communication between the legitimate transmitter and receiver. Jammer attacks cause to block the transmission or to prevent the reception of the signal accurately depending on the attack type.

In literature, a lot of research studies investigate anti-jamming techniques. In [46], an adaptive synchronization algorithm proposed to increase the robustness of the LTE system against jamming attacks or partial band interference. Authors aim to find locations of primary synchronization signal (PSS) and the secondary synchronization signal (SSS) in LTE downlink frame. Simulations results of adaptive filtering based algorithm surpass previous algorithms. Shahriar and et.al [47] give prominence to vulnerabilities of OFDM in the existence of pilot tone based jamming attacks. They propose two pilot signal randomization schemes to mitigate the type of attacks and

their completely random scheme indicates a higher performance than the conventional scheme. A GLRT based detection method is proposed in [48] where is considered the attacks of multiple antenna jammer against a massive MIMO system. The compelling point of this methodology is to take advantage of unused pilots. On the other side, this approach contingent upon the assumption that jammer has knowledge regarding transmission protocol and the pilot set. In [49], to improve a protection mechanism from smart jamming attacks to 802.11 where authors propose a resilient channel hoping strategy. The performance of the method in terms of the throughput is noticeable in the presence of a jammer. Xiao and et al. [50] formulate a Stackelberg equilibrium of anti-jamming MIMO nonorthogonal multiple access (NOMA) transmission game to increase the NOMA communication quality. They proposed a reinforcement learning based power allocation methodology against smart jammer attacks and improve the signal-to-interference-plus-noise ratio (SINR) and sum rate. Another interesting study [51] treats smart jamming attacks with the issue of wireless energy harvesting. The proposed learning based algorithm converges in terms of efficiency compared to other algorithms. In [52], a countermeasure for jamming attacks is investigated for vehicular ad hoc networks and authors compare Q-learning based relay strategy with the proposed relay strategy and they obtain lower bit-error-rate (BER).

It can be seen from the above that a general summary of some studies in the literature is presented related to mitigate or prevent jammers. Also, another significant importance comes in sight. In order to provide secure transmissions against various jamming attacks, firstly presence of jammers must be detected and the type of attacker is determined. Then, anti-jamming techniques can be evaluated contingent on the type of jamming attack.

Jamming attacks can be defined and grouped as noise jamming, correlated jamming, interference jamming [53] or barrage jamming, partial band jamming, single/multiple tune jamming. This is changed depending on the statistical characteristic of a jamming signal or an attack model of the jammer to spectrum. In addition, the pattern of the jamming signal affects the jammer identification strategy. Aziz and et al. [54] model the reciprocal action as a Bayesian game between the LTE/LTE-A network and the smart jammer and propose to estimate the type of the jammer. Authors present a

threat based algorithm which uses Wilcoxon's rank-sum test and it shows an acceptable performance. In [55], an unsupervised learning based jammer detection approach is presented with a novel metric depends on variations of relative speed for the vehicular network. The presented methodology is capable to discriminate intentional and unintentional (interference) jamming attacks. Additionally, authors show receive signal strength indicator (RSSI), packet delivery ratio (PDR), SINR, etc. based schemes cannot differentiate these attacks or the unique pattern of a jamming signal. Jammer types are categorized as constant, deceptive, random and reactive jammers in [56]. Authors focus on the medium access control (MAC) layers attacks and present packet send ratio (PSR) and PDR based an identification model. Regarding the attacks to 802.11, in [57] several machine learning methods are performed via channel and performance metrics. A significant improvement on the detection accuracy is observed experimentally for indoor and mobile outdoor scenarios. In the study [58] of Zhang and et al., they present a different categorization scheme for cognitive jamming: fast versus slow jamming, and flat versus frequency selective jamming and estimate the spectral density of jammers.

To determine the identification of jammer is significantly important to design an anti-jamming strategy against adversary effects. Above all, an indispensable step is a detection of a presence/absence of an attacker to provide the maintenance of secure communication systems. In [59], a satisfactory function and a payoff matrix are defined related to capacity-rate relation as a performance measurement and authors determine the existence of jammer with the threshold-based approach. In [60], the presented detection algorithm takes the network performance measurements taken over time and decide about the presence of jammer using threshold. The main point makes this study different is to tackle the measurements by focusing on specific regions in time. Kurt and et al. [61] propose a robust online detection algorithm based on the online ML estimation against smart and more capable attackers. The results show the effectiveness of the presented approach with a lower delay than other approaches. In [62], the detection of a jammer is provided by using the threshold of RSSI measurements at the first step and SVM classifier. In [63], the author compares the minimum description length criterion and random matrix theory to detect pilot

contamination attack and the minimum description length based estimations show higher performance.

A lot of detection and identification methodologies are mentioned briefly to give a general overview, above. In addition, several countermeasure strategies are explained to emphasize the reasons why the detection and identification of an attacker are requisite as well as anti-jamming methods.

1.3 Contributions of Thesis

In this thesis study, deep learning based solutions are proposed related to some security problems of wireless communication systems.

Firstly, the problem of the identification of jammers that transmit signals to prevent the maintenance of communication is dealt with. Four scenarios which are two smart jammer attacks, a conventional attack, and the absence of the jammer attacks are considered for Long Term Evolution (LTE) network. Regarding attack scenarios, jamming signals are generated for different signal to noise ratios (SNRs), signal to jamming ratios (SJRs) and distances on MATLAB. Choi-Williams, Gabor wavelet, short time-frequency transformations are used to extract the jamming signal characteristic from raw samples. According to considered scenarios and cases, created datasets are used to train classifiers; CNN and SVM based. The proposed DCNN architecture is designed using convolutional, pooling and dense layers. Both classifiers are evaluated with several methods such as optimization and regularization techniques. Parameters of each model are tuned after a lot of experiments. Classifiers are compared in terms of identification performance and DCNN defined as state of the art shows better performance than the SVM classifier. In addition to this, the wavelet transform's confirmation is observed to represent the raw data.

Secondly, detection and identification of jammer attacks are investigated together and in this study, the datasets are gathered over a real-time test-bed. Six scenarios that contain barrage and reference signal jammer attack during the absence/presence of legitimate transmissions separately, the absence of any attacker and transmitter, the existence of transmission are considered. Complex Gaussian derivative wavelet and short time-frequency transformations are used as a pre-processing method. In the

classification stage, DCNN is chosen due to the performance at the previous work and it compared with DRNN which is also another state-of-the-art technique. The proposed DRNN architecture is designed using Long Short-Term Memory (LSTM) and dense layers. Two neural network models are trained using adaptive moment estimation optimizer and hyper-parameters of the models are determined after several trials. The obtained results show that DCNN classifier surpasses the DRNN classifier in terms of jamming signal detection and identification performance.

2. CONVOLUTIONAL NEURAL NETWORKS

Deep convolutional neural networks (DCNNs) are at the forefront especially in image classification methods with its particular architectural structure, unlike traditional machine learning methods which need features manually determined by an expert. Convolutional neural networks (CNNs) are a specialized kind of neural network for data processing and considered as a grid-like structure. CNNs use the convolution operator at least one layer unlike the other types of neural networks. Usage of the convolution operator provides three advantages that play significant roles for improving the machine learning system: sparse interactions, parameter sharing, and equivariant representations [64].

CNNs can be thought of as a series of layers. Generally, convolutional layers, downsampling layers are used to extract features and a flatten layer is preferred to create vector forms of feature maps before the classification part of the network architecture.

2.1 Feature extraction

Pooling, activation, and convolution are fundamental operations of CNNs to extract features. From the first layer to the last one, simple features are extracted initially, and gradually it continues to more complex features.

2.1.1 Convolution stage

The convolution layer is based on a discrete convolution process. Discrete convolution is given as the following;

$$y[t] = \sum_{a=-\infty}^{\infty} x[a]w[t-a], \tag{2.1}$$

where x is the input and w is the kernel or filter that sifts through the information in the input and filter the parts that are summation to it and exclude the rest. Input data of convolution layers are generally multidimensional array. A convolution operator

depends on tensor shape can be implemented in more dimension. The two-dimensional can be defined as in below

$$Y[k,l] = \sum_{a_1=-\infty}^{\infty} \sum_{a_2=-\infty}^{\infty} X[k-a_1, l-a_2]W[a_1,a_2], \tag{2.2}$$

where $X_{n_1 n_2}$ represents two dimensional input matrix and $W_{m_1 m_2}$ is a kernel matrix with $m_1 \leqslant n_1$ and $m_2 \leqslant n_2$. The main goal of the convolution operator usage is to reduce the input image to its essential features. A feature map, also known as an activation map, is produced by sliding the convolution filter over the input signal. The sliding scale is a hyper-parameter known as stride. The size of feature map or convolution layer output length for each dimensions can be realized using the following equation

$$o_d = \frac{n_d - m_d}{s_d} + 1, \tag{2.3}$$

where d is the number of dimensions, n_d and m_d represent the length of the input vector and the kernel length in d^{th} dimension, where s is the value of stride.

2.1.2 Activation functions

In neural networks, when output data is generated from input data, activation functions are proposed to introduce non-linearity. Some common activation functions in literature are described in follow.

2.1.2.1 Sigmoid function

Also called the logit or the logistic function. The output value of the function is between 0 and 1.

$$\phi(z) = \frac{1}{1+e^{-z}}. \tag{2.4}$$

2.1.2.2 Tangent Hyperbolic function

It is similar to sigmoid with the properties of being continuous and differentiable but its output value's range is between -1 and 1.

$$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}. \tag{2.5}$$

2.1.2.3 Rectified Linear Unit (ReLu)

It offers much faster learning than sigmoid and tangent functions because of the simpler mathematical operations. Although it is continuous, it is not differentiable.

$$\phi(z) = max(0, z). \tag{2.6}$$

2.1.2.4 Softmax function

It is the type of sigmoid function and the softmax output can be considered as a probability distribution over a finite set of outcomes [64]. Therefore it is used in the output layer of the classifier, especially for multiclass classification problems.

$$\phi(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \tag{2.7}$$

where z_i is input of the softmax, i is the output index and K is the number of classes.

Despite the presence of various activation functions such as sigmoid functions (2.4), hyperbolic tangent functions (2.5), rectified linear unit activation (2.6) is often preferred. When the tensors are all scaled to be positive, it is the most widely chosen as an activation function that preserves variance in the positive domain. In output layers, a sigmoid function can be preferred for binary classification. As a normalized function, the softmax function 2.7 is a very strong option to obtain meaningful class membership predictions in multiclass settings [65].

2.1.3 Pooling layer

Another important part of CNNs is the pooling operation. A pooling layer does not include learnable parameters like bias units or weights. Pooling operations are used to decrease the size of feature maps with some functions which calculate the average or the maximum value of each distinct region of size $a_1 \times a_2$ from the input. It helps the representation become slightly invariant to small translations of the input. A pooling layer solves disadvantages related to the probability of over-fitting and computational complexity [65].

2.1.4 Flatten layer

A flatten layer is used between feature extraction and classification sections to arrange tensor shape. The output tensor shape is mostly two or more dimensional tensor. Therefore the tensor shape is decreased to a one-dimensional vector with flatten layer to get suitable input size for dense layers.

2.2 Classification

Fully-connected layers are often preferred for the classification part of CNNs architectures.

2.2.1 Fully-connected layer

Fully-connected layers are also called dense layers and correspond to convolution layers with a kernel of size (1×1). In the fully-connected layer, all units are connected with the units at the previous layer. Outputs are multiplied with weight and are given as an inputs to the units of the next layers. This processes can be represented as follows

$$\mathbf{y} = W^{\top}\mathbf{x} + \mathbf{b}, \tag{2.8}$$

where $\mathbf{y}$ is the output vector of the fully connected layer, $\mathbf{x}$ is the input vector, W denotes the matrix includes the weights of the connections between the neurons, and $\mathbf{b}$ represents the bias term vector.

2.3 CNN Learning

CNNs occur a lot of layers and connections between these layers. As a result of this, they consist a lot of parameters which are required to be tuned. The main purpose of DCNNs is finding out the best values for parameters because they affect directly classification performance. The learning ability of DCNNs increases with tuning parameters. In this section, main issues of neural networks and some techniques are explained briefly.

2.3.1 Loss functions

A loss function quantifies the difference between the estimated output of the model (the prediction) and the correct output (the ground truth) to provide better convergence.

The type of loss function used in a CNN model depends on the end problem. The generic set problems for which neural networks are usually used (and the associated loss functions) can be categorized into the following categories [66].

- Binary classification (SVM hinge loss, Squared hinge loss).
- Identity verification (Contrastive loss).
- Multi-class classification (cross-entropy loss, expectation loss).
- Regression (SSIM, ‘1 error, Euclidean loss).

As they are not applied within the scope of this thesis, only the title cross-entropy loss function will be explained.

2.3.1.1 Cross-Entropy Loss Function

The cross-entropy loss function is largely used for binary classification problems and it is defined as follow:

$$\mathrm{L(p, y)} = -\left[y\mathrm{log}(p) + (1-y)\mathrm{log}(1-p)\right], \tag{2.9}$$

where $y \in \{0, 1\}$ represents a class label and p is the probability of the target. Regarding multiclass classification, the probability of each class is calculated according to the softmax function (2.7) and the cross-entropy loss (2.10) for an instance is generated as follows:

$$\mathrm{L}(\mathbf{p}, \mathbf{y}) = -\sum_{k=1}^{K} y_k \mathrm{log}(p_k), \tag{2.10}$$

where $\mathbf{y} = [0, \cdots, y_k, \cdots, 0]$ is the output vector, K is the number of classes and p_k is the estimated probability that the instance x belongs to class k. y_k is equal to 1 if the target class is k. Otherwise, it is equal to 0.

2.3.2 Optimization

The traditional tendency is using one of the closed-form functions to find correct solutions. Although it can be considered regarding small data sets, it becomes quite complex in the cases which contain a lot of parameters. Optimization algorithms are used when the closed form equation cannot be preferable due to the presence of

a singular matrix, large dataset. Gradient descent based or cognate methods such as adaptive moment estimation (ADAM), adaptive gradient algorithm (AdaGrad), root mean square propagation (RMSprop)..etc are convenient especially to train deep learning algorithms. In the scope of thesis, ADAM and gradient descent algorithms are explained as follows.

2.3.2.1 Gradient Descent

Gradient descent based optimization methods adjust the parameters iteratively to minimize the cost function which calculates a sum of loss functions using the training data set. When cross-entropy loss function is used as a loss function, the cost function is defined for binary classification problems as follows:

$$J(\theta) = -\frac{1}{N}\sum_{n=1}^{N}[y^{(n)}\log(p^{(n)}) + (1-y^{(n)})\log(1-p^{(n)})], \tag{2.11}$$

and the cost function is defined for multiclass classification problems as follows:

$$J(\Theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_k^{(n)}\log(p_k^{(n)}), \tag{2.12}$$

where N is the number of instances, θ is the parameter vector and Θ is the parameter matrix. Gradient descent measures the local gradient of the cost function with regards to the parameter vector, and it goes in the direction of descending gradient until the algorithm converges to minimum. At this point important hyperparameter must be determined carefully is learning rate which specifies how often updating of parameters is occurred.

Figure 2.1 shows the convergence of the cost function for two learning rates and both of them converge to the minimum of the cost function. The small learning rate is preferred to complete learning in a short time duration but this is not always applied. Regarding the case in Figure 2.2, the small learning rate is better to choose because the global minimum value is overlooked when the learning rate is large. The learning rate changes depending on the gradient of the cost function which is calculated at each iteration and at each unit. At the output layer the gradient of the cost function (2.12) can be defined for k^{th} class as follows:

$$\nabla_{\theta_k} J(\Theta) = \frac{1}{N}\sum_{n=1}^{N}\left(p_k^{(n)} - y_k^{(n)}\right)\mathbf{z_k}^{(n)}, \tag{2.13}$$

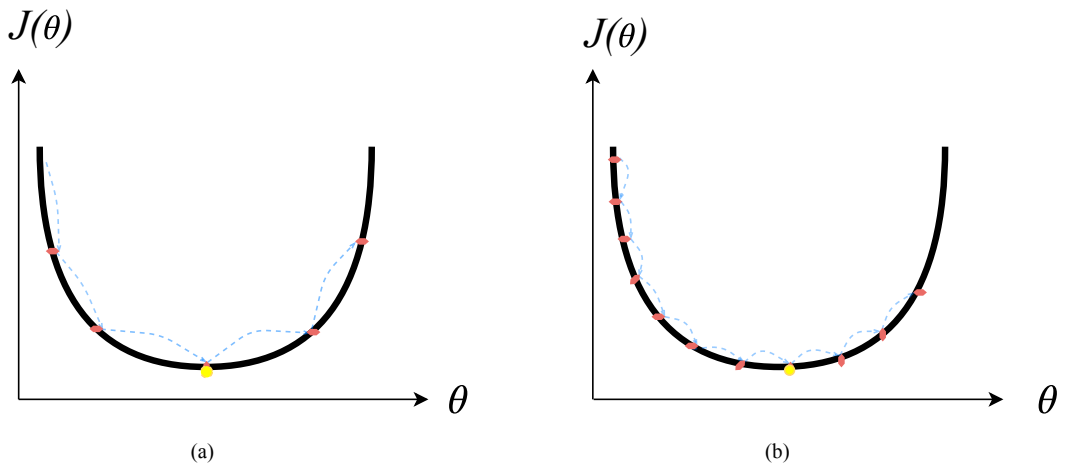

Figure 2.1 : The convergence representations of gradient descent algorithm via different learning rates when the cost function is convex. (a) High learning rate. (b) Small learning rate.

where $\mathbf{z_k}^{(n)}$ is the score regarding n^{th} instance. Learning rate is updated at each iteration according to the equations as in below:

$$\Theta_\iota = \Theta_{\iota-1} - \eta\Omega_\iota, \tag{2.14}$$

$$\Omega_\iota = \nabla_{\theta_k} J(\Theta_\iota), \tag{2.15}$$

where $\nabla(\cdot)$ indicates the partial derivation, η is the learning rate and Ω_ι is the gradient matrix of the parameter matrix at the iteration time ι. All parameters are updated

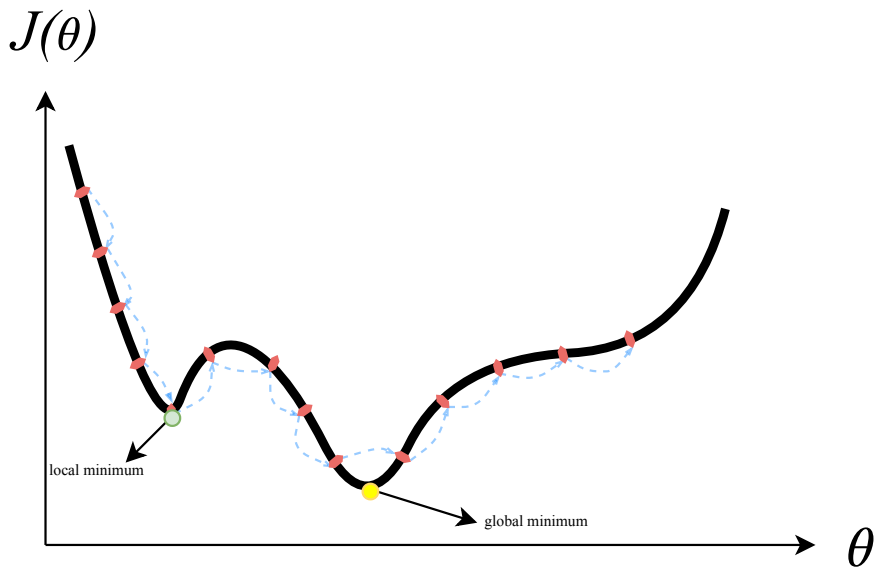

Figure 2.2 : The convergence representations of gradient descent algorithm via different learning rates when the cost function is convex.

according to the chain rule in the backpropagation algorithm in each iteration from the output unit to inputs.

- Batch Gradient Descent: It is another name of gradient descent. In batch gradient descent, algorithm calculate the derivation of the cost function using whole instances in the training dataset. It causes a long training duration especially when it is used to train large datasets.

- Stochastic Gradient Descent: When the gradient descent is improbable, it is applied. The parameters are updated according to the gradient of the cost function which is calculated over one sample. Therefore, it provides light computational complexity for one iteration and sometimes a faster convergence.

- Mini-Batch Gradient Descent: The gradient of the cost function is calculated over a small part of the training dataset and parameters are updated. The number of samples in the mini-batch called as batch size.

2.3.2.2 Adaptive Moment Estimation

Adam can be defined as an adaptive learning rate method because of the capability to compute individual learning rates for different parameters. It calculates the learning rate depends on the gradient of cost function but also estimates the first moment (mean) and the second moment (variance) of the gradient to update parameters. Estimations of mean and variance can be calculated using the following equations:

$$\widehat{E}\left[\psi\right]_{\iota} = \frac{E\left[\psi\right]_{\iota}}{1-\left(\gamma_1\right)^{\iota}}, \tag{2.16}$$

$$\widehat{E}\left[\psi^2\right]_{\iota} = \frac{E\left[\psi^2\right]_{\iota}}{1-\left(\gamma_2\right)^{\iota}}, \tag{2.17}$$

where ψ indicates the gradient of the cost function, γ_1 and γ_2 are values of the decay. After that, the parameters are updated according to the following equation:

$$\theta_{\iota}^{\tau} = \theta_{\iota-1}^{\tau} - \frac{\eta}{\sqrt{\widehat{E}\left[\psi^2\right]_{\iota}} + \epsilon}, \tag{2.18}$$

where τ denotes adaptive learning rate of each individual parameter and ϵ is a small term which is used to prevent division to zero.

2.3.3 Backpropogation

Backpropagation has several definitions in the literature. We use backpropagation in the meaning that the iterative procedure decreases the cost in a sequence by adjusting the weights. Backpropagation does not update the weights of the model and adjustment of weights is handled by the optimizer depending on the gradient of the cost function.

Backpropagation algorithm takes the partial derivative of the cost function with respect to the weights in accordance with the chain rule (2.19) and propagates back to the network from the outputs to the inputs. Regarding the multiclass classification problem, we assume that softmax function and sigmoid function are used at the output layer, hidden layer respectively. Gradients are computed related to the neural network architecture in Figure 2.3 for one instance as follows. To update v_{it}, a weight at the second layer, the chain rule is:

$$\frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial v_{it}} = \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_i} \frac{\partial p_i}{\partial z_i} \frac{\partial z_i}{\partial v_{it}}, \tag{2.19}$$

$$\frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_i} = -\sum_{k=1}^{K} \frac{y_k}{p_k}, \tag{2.20}$$

$$\begin{aligned} \frac{\partial p_i}{\partial z_i} &= \begin{cases} \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} - \left(\frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}\right)^2, & i = k \\ -\frac{e^{z_i} e^{z_k}}{\left(\sum_{k=1}^{K} e^{z_k}\right)^2}, & i \neq k \end{cases} \\ &= \begin{cases} p_i - {p_k}^2, & i = k \\ -p_i \, p_k, & i \neq k \end{cases} \end{aligned} \tag{2.21}$$

$$\begin{aligned} \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial z_i} &= \sum_{k=1}^{K} \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_i} \frac{\partial p_i}{\partial z_i}, \\ &= \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_i} \frac{\partial p_i}{\partial z_i} - \sum_{k \neq i}^{K} \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_k} \frac{\partial p_k}{\partial z_i}, \\ &= -y_i(1 - p_i) + \sum_{k \neq i}^{K} y_k p_i, \\ &= p_i - y_i. \end{aligned} \tag{2.22}$$

$$z_i = \sum_{t=1}^{T} s_t v_{it} + b_t, \tag{2.23}$$

where b_t is the bias term, s_t is the output of t^{th} neuron at hidden layer.

$$\frac{\partial z_i}{\partial v_{it}} = \sum_{t=1}^{T} s_t, \tag{2.24}$$

$$\frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial v_{it}} = (p_i - y_i) \sum_{t=1}^{T} s_t, \tag{2.25}$$

To update w_{td}, a weight at first layer chain rule is:

$$\frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial w_{td}} = \frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial p_i} \frac{\partial p_i}{\partial z_i} \frac{\partial z_i}{\partial s_t} \frac{\partial s_t}{\partial u_t} \frac{\partial u_t}{\partial w_{td}}. \tag{2.26}$$

$$\begin{aligned} \frac{\partial s_t}{\partial u_t} &= \left(\frac{1}{1+e^{-u_t}}\right)\left(1 - \frac{1}{1+e^{-u_t}}\right), \\ &= s_t\left(1 - s_t\right). \end{aligned} \tag{2.27}$$

$$u_t = \sum_{n=1}^{N} x_d w_{td}, \tag{2.28}$$

$$\frac{\partial u_t}{\partial w_{td}} = \sum_{n=1}^{D} x_d, \tag{2.29}$$

$$\frac{\partial \mathrm{L}(\mathrm{p_i}, \mathrm{y_i})}{\partial w_{td}} = \sum_{d=1}^{D} \left(p_i - y_i\right) v_{it}\ s_t\left(1 - s_t\right) x_d. \tag{2.30}$$

2.3.4 Regularization

There are many parameters in the training phase of DCNNs, and sometimes this causes over-fit means that the model is very successful on the training data, but fails when compared to new data. Regularization techniques are used as a solution to such problems.

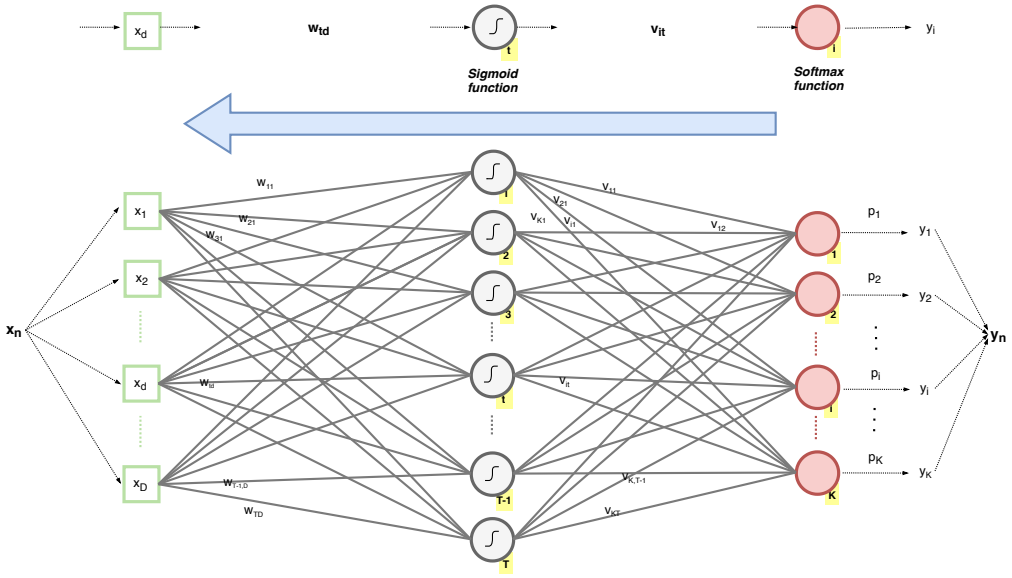

Figure 2.3 : Backpropagation algorithm on two layer neural network.

2.3.4.1 Batch Normalization

Batch normalization provides to learn a more complex or flexible model, by normalizing the mean and variance of the output activations. The distribution of activations at each layer shows a variation when the parameters are updated during training. It improves learning by reducing this internal covariance shift. In this way, a model is more resistant to problems such as vanishing, exploiting problems.

As a first step, given d-dimensional feature vector $\mathbf{f} = (f^{(1)} f^{(2)} \cdots f^{(d)})$, all features are normalized as in below;

$$\widehat{\mathbf{f}} = \frac{f^{(d)} - \mathrm{E}[f^{(d)}]}{\sqrt{\mathrm{Var}[f^{(d)}]}} \,. \tag{2.31}$$

After the feature normalization, batch normalization can be defined for each one of the batches as in below:

$$\mu_B = \frac{1}{R} \sum_{r=1}^{R} \mathbf{f_r}, \tag{2.32}$$

$$\sigma_B^2 = \frac{1}{R} \sum_{r=1}^{R} (\mathbf{f_r} - \mu_B)^2 \,, \tag{2.33}$$

where R represents the total number of features at one batch, μ_B and σ_B^2 are mean and variance of the batch respectively.

$$\widehat{\mathbf{f}}_r = \frac{\mathbf{f}_r - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \,. \tag{2.34}$$

All normalized features are scaled and shifted to hinder the network ability to fully utilize nonlinear transformations. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also in some cases eliminating the need for Dropout [67].

2.3.4.2 Dropout

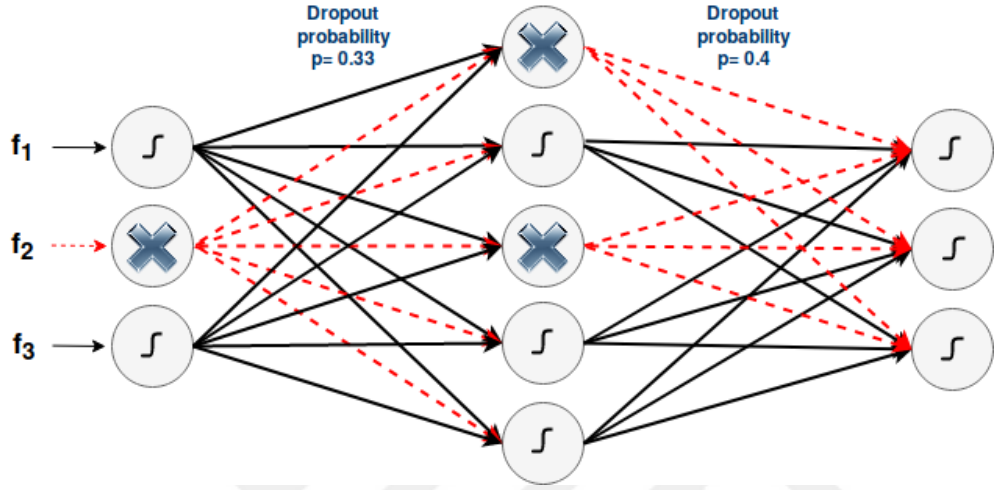

Figure 2.4 : Dropout

Dropout can be seen as a stochastic regularization technique. It prevents overfitting and provides a way of approximately combining exponentially many different neural network architectures efficiently [68]. Dropout prevents to move some of the outputs of the previous layer to the next layer. This can be considered as a masking applied to cross-layer transitions. Dropout is applied to a unit in a layer which must learn the pattern with randomly selected outputs of previous units. In this way, the hidden units are enforced to extract valuable features. Moreover, it reduces the risk of training data memorization. Hyperparameters expressing the probability of the masking process are called "dropout rate". The process can be seen in Figure 2.4.

3. IDENTIFICATION OF SMART JAMMERS: LEARNING BASED APPROACHES

Orthogonal frequency division multiple access (OFDMA) technique constitutes the physical layer multiplexing method of choice for the Long Term Evolution (LTE) networks and its later Releases due to its robustness against noise and fading impacts [69]. Based on the advantages provided by OFDMA, LTE has brought up numerous favorable properties including but not limited to higher data rates, better coverage, higher energy efficiency and lower latency than previous cellular networking technologies [70]. Despite the advantages that come with the OFDMA technique, LTE networks suffer from attacks generated by active radio nodes [71]. These attacks generally include cases where an attacker (a jamming node) transmits a signal aiming to falsify the receiver or disrupt the communication. Such communication disruptions are frequently referred to as layer-1 denial-of-service (DoS) attacks.

Jamming attacks have been one of the conspicuous issues in wireless communication networks. As communication systems become an integral part of our daily lives, transmission of the critical information is expected to be more reliable, increasing the effective impact of the jamming attacks. These attacks can be initiated in various ways through different types of jamming nodes. In order to avoid or to mitigate the effects of the jamming, it is critical to detect the existence of a jamming node and to detect its attack methodology. These processes are respectively named as jammer detection and jammer identification.

As detailed in the following section, jammer identification is usually based on the network measurements or signal characteristics. On determining the attack type, identification algorithms may use different probabilistic approaches. Machine learning techniques also gain importance on identification and detection algorithms. Especially, deep convolutional neural network (DCNNs) have been studied in different communication problems that require quick response in real-time applications. The

main application areas can be listed as localization [72], modulation classification [28], channel decoding [73] and waveform recognition [32].

A common step in all of these detection and classification methods is a time-frequency transformation (TFT) based pre-processing which is capable of capturing the jamming effects on the received signal. Most commonly used TFT method is spectrogram and has been utilized for different signal analysis purposes [74]. Spectrogram is based on short time Fourier transform (STFT) and represents the signal on the time-frequency plane. As it will be detailed below, spectrogram uses fixed window size which results in fixed resolution of the signal on time-frequency plane.

Concurrent with the evolution of communication networks, newly introduced jammer types may attack in very short time intervals or narrow and changing frequency bands. In that way, jamming nodes are able to hide in the time-frequency plane and become invisible to conventional TFT methods such as STFT. These kind of attackers are named as smart jammers, and their effects are already shown in numerical and measurement studies [75, 76]. Reduced hardware costs with easily accessible software enable smart jammers to observe the transmissions and detect the vulnerable parts of the transmitted packets, possibly targeting the reference signals. With these observations, smart jammers gain an advantage over the legitimate nodes as shown in [54]. Defense mechanisms against jamming attacks are proposed from different perspectives such as repeated game algorithms [54], or frequency hopping algorithms [77]. These network strategy algorithms require the knowledge of the jammer type in order to combat against the jammer attack. Therefore, the jammer identification is a requirement prior to the jamming prevention/anti-jamming systems.

Motivated by this problem, we propose a wavelet tranformation based pre-processing for smart jammer identification in order to improve detection and classification performance. Wavelet transforms are commonly used signal processing techniques thanks to the tunable time and frequency resolution properties [78]. Similar to Fourier transform, wavelet transforms represent a signal through a linear combination on particular basis functions. In Fourier transform, the basis functions are sinusoidals which are not localized on time-frequency plane due to continuous oscillation. On the contrary, wavelets are finite in time and localized on time-frequency plane,

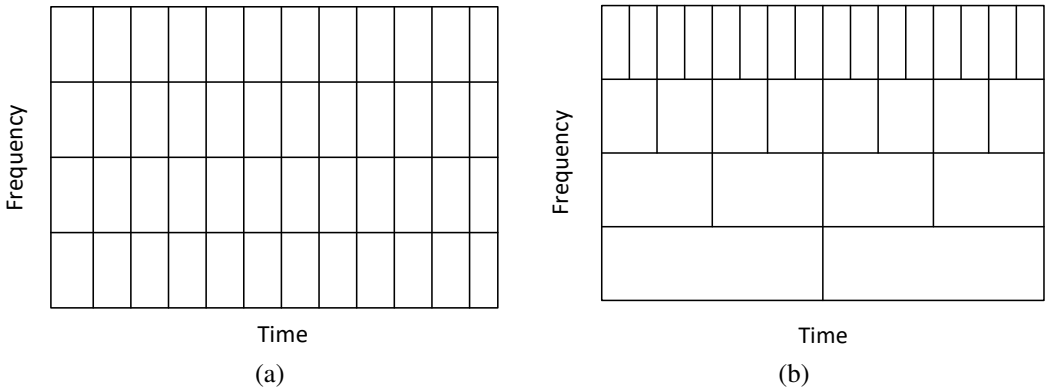

Figure 3.1 : An exemplary graphical representation of (a) short time Fourier transform (STFT) and (b) wavelet transform.

enabling successful detection and identification of the smart jammers hiding in the time-frequency plane.

STFT provides an observation on temporal changes of the signal by applying a rectangular window to the signal before Fourier transform. Since same rectangular window is applied to the signal in STFT, time-frequency resolution of the signal would be same at all positions. Unlike STFT, varying window size in the wavelet transforms provides multi-resolution on the different positions of the signal [79], as highlighted in Figure 3.1. In this way, we can provide an adaptive resolution property against changing signal characteristics resulting from smart jammers.

We consider the LTE-downlink channel jamming scenario, where a stationary monitoring node (MN) captures downlink channel observations by continuously processing the received baseband signal. The MN is synchronized with the cell and a stationary jammer attacks the transmission from the base station (eNB), as shown in Figure 3.3. Note that, MN does not have any prior knowledge about the jammer attack. In the presence of a smart jammer, two major vulnerabilities become apparent in LTE-downlink transmission: synchronization signals and reference signals [80]. These signals are transmitted along with the message signal over the network. Unlike message signals, they contain critical information and even a partial disruption of this information may cause loss of the complete LTE packet. Our main contributions can be listed as the following;

- We introduce a novel system model for the identification of the smart jammers. Proposed identification system can be divided into two steps: (i) a pre-processing

step to highlight the disrupted parts of the signal, (ii) a classification step that automatically identifies jamming signals.

- We provide a wavelet based pre-processing step that conveys multi-resolution representation of the signals. The validity of the proposed pre-processing step is confirmed via simulations. We compare Gabor wavelet transform with TFT's that were previously used for classification pre-processing: spectrogram and Choi-Williams.

- For the classification step, we propose a DCNN architecture to automatically extract the features of the transformed signals and to classify them.

- As an alternative for the classification step, we utilized SVM also for the first time considering the smart jammer identification. The classification accuracies of two classification schemes are compared with each other.

- The proposed identification scheme is repeated for different location cases considering that the identification accuracy could change with respect to distances between the jammer and the MN or between the eNB and the MN.

The rest of this paper is organized as follows. In Section II, we discuss jammer detection, classification and mitigation literature. In Section III-A, we briefly review some relevant features of LTE downlink signal model. In Section III-B, we discuss possible jammer attack types. In Section IV, we present the proposed system model for jammer identification. Numerical analysis and simulations are presented in Section V, and the open issues are given in Section VI. Finally, the paper is concluded in VII.

3.1 System and Attacker Model

This section is devoted to the review of relevant characteristics of LTE downlink channel model and jamming attack models. As previously mentioned, smart jammers would require some insights about the physical layer to focus on the vulnerabilities of the system. Therefore, before describing jamming attack models, the physical layer properties of an LTE system should be addressed.

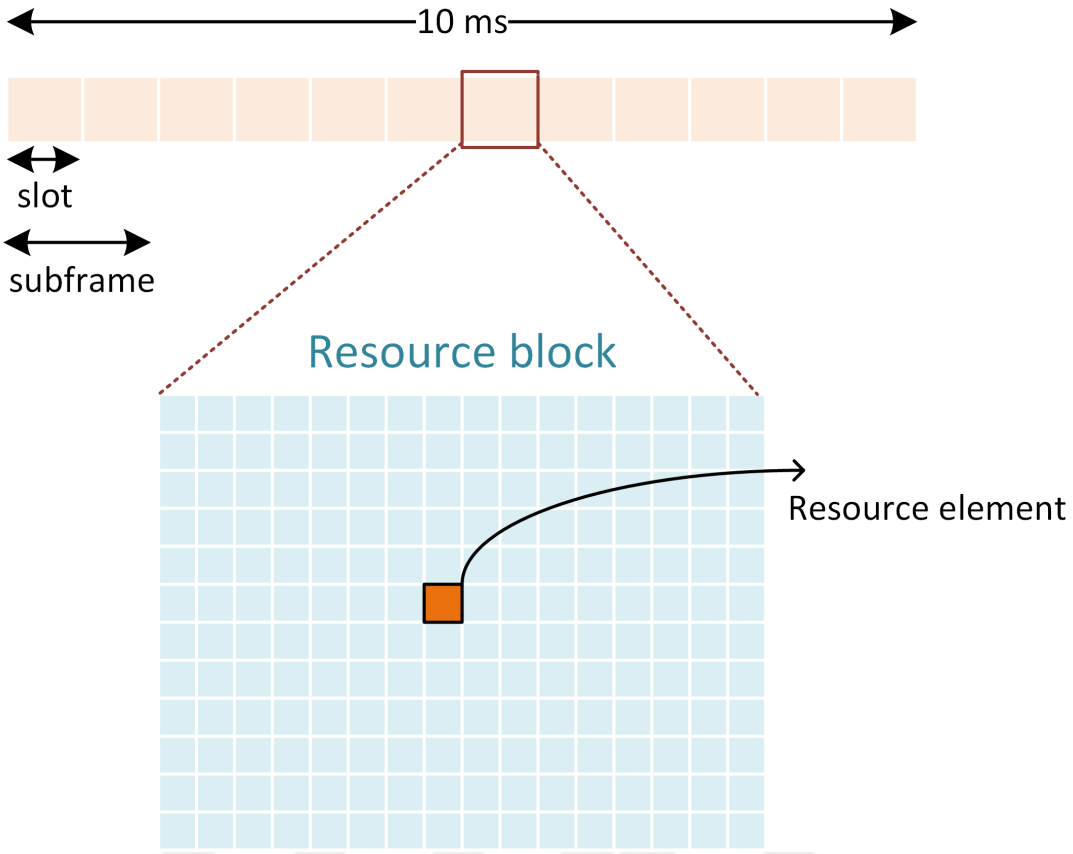

Figure 3.2 : LTE frame, subframes and time slots.

3.1.1 LTE - PHY model

LTE downlink channel uses OFDMA as the channel access scheme. The transmission can be in frequency division duplex (FDD) or time division duplex (TDD) mode [81]. Figure 3.2 shows an illustration of an LTE frame considering FDD transmission mode. Data is delivered in frames of 10 milliseconds. An LTE frame is composed of ten subframes of 1 ms, and each subframe contains 2 time slots of 0.5 ms. Each time slot contains L symbols, and L depends on the cyclic-prefix (CP) mode. K denotes the total number of subcarriers. Depending on the transmission bandwidth, K changes between 128 to 2048. The smallest defined unit, $a_{k,l}$, denotes a resource element which consists of k^{th} subcarrier during the l^{th} symbol, where $k = 0, 1, \cdots, K-1$ and $l = 0, 1, \cdots, L-1$. According to the selected bandwidth and protocol, control and message signals are mapped into empty resource blocks.

From an attacker perspective, obtaining the exact subcarrier and symbol index of the control signals can be beneficial because attacker can damage the transmission by

attacking only the resource elements where control signals are mapped. The remaining part of this section will focus on the generation and mapping of the control signals in LTE downlink channel.

3.1.1.1 Synchronization signals

In order to determine and to synchronize to a cell, a user equipment (UE) needs to acquire the frame timing information, estimate the carrier frequency offset (CFO) and also identify the cell. For this purposes, eNB generates and maps two signals into resource elements: primary synchronization signal (PSS) and secondary synchronization signal (SSS).

eNB maps the PSS in the last symbol of first time slot of the first subframe (subframe 0). The PSS is also mapped in subframe 5 which means UE can be synchronized on 5 ms basis. SSS symbols are also mapped in the same subframe of PSS with the same subcarrier indexes but a symbol before the PSS.

To generate the PSS and the SSS, the transmitter uses a complex valued Zadoff Chu (ZC) sequence [81]. Let $d_u(m)$ denote the ZC sequence of the PSS and the SSS,

$$d_u(m) = \begin{cases} e^{\frac{-j\pi um(m+1)}{63}}, & 0 \leq m \leq 30 \\ e^{\frac{-j\pi u(m+1)(m+2)}{63}}, & 31 \leq m \leq 61 \end{cases}, \tag{3.1}$$

where u is the root value for the ZC series and changes with the index of the symbol. In this case $d_u(m)$ is mapped according to the relevant resource elements;

$$a_{k_s,l_s} = d_u(m),$$

where $m = 0, 1, \ldots, 61$ and $u \in \{0, 10\}$. k_s denotes the subcarrier indices of mapped resource elements with $k_s = m - 31 + K/2$. l_s denotes the symbol indexes of the mapped resource elements and set to $L - 2$ for the PSS and $L - 1$ for the SSS.

At the UE, first PSS is extracted for time frame and frequency synchronization. In the next step UE extracts the SSS. Through the SSS, the UE extracts about the CP mode and the duplexing mode used by the cell. From their combination, the mapped locations of the reference signals can be found in order to realize the channel estimation and equalization steps. In case of any disruption on the PSS and SSS, not only the synchronization at the UE is affected, but also the packet will get corrupted because of the erroneous channel estimation [46].

3.1.1.2 Reference signals

Reference signals are periodically transmitted in LTE networks to perform channel estimation and frequency domain equalization at the receiver side after CP removal and demodulation steps. The cell identification determines time and frequency domain locations of the quadrature phase shift keying (QPSK) modulated reference signals. In this work, subcarrier and symbol indices of the reference signals are assumed as fixed, respectively as $k_{rf} = 0, 7, 14, \cdots, N$ and $l_{rf} = 0, 7, 14, \cdots, N$. Each $a_{k_{rf},l_{rf}}$ is filled with a random complex number, $a_{k_{rf},l_{rf}} \in \{\frac{1+j}{\sqrt{2}}, \frac{1-j}{\sqrt{2}}, \frac{-1+j}{\sqrt{2}}, \frac{-1-j}{\sqrt{2}}\}$. UE is assumed to know the index values k_{rf}, l_{rf} as well as the values of the reference signals. The receiver produces an initial estimate for the channel coefficients of the received signal by utilizing the reference signals. Afterwards, these initial estimates are interpolated by using the additional information coming from the other resource elements, and the channel coefficient estimates are finalized. Following the channel estimation, UE recovers the received packet by using the estimated channel coefficients for equalization. If the estimated coefficients are erroneous, then coefficients belonging to other resource elements will also be inaccurate after interpolation. Additional control signals may also get mapped into resource blocks [81]. Yet, in our model, resource elements are filled by the union of the resource elements filled with synchronization signals, reference signals and message signals. The mapped subcarrier and symbol indexes for message signal can be denoted with k_m and l_m, respectively.

After mapping all signals into resource blocks, discrete time baseband representation of the transmitted downlink LTE signal $s(l)$ can be obtained as

$$s(l) = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} a_{k,l} e^{j2\pi kl/K} \omega\left(l/K + K_{CP} - 1\right), \tag{3.2}$$

where K_{CP} denotes the cyclic-prefix length and $\omega(l')$ is a discrete rectangular window that is defined by

$$\omega(l') = \begin{cases} 1 & ; 0 \leq l' \leq 1 \\ 0 & ; \text{otherwise} \end{cases}. \tag{3.3}$$

3.1.2 Jamming attacks

Jamming attacks can be classified according to different characteristics such as attacked bandwidth and jamming signal transmission. We will focus on jammers

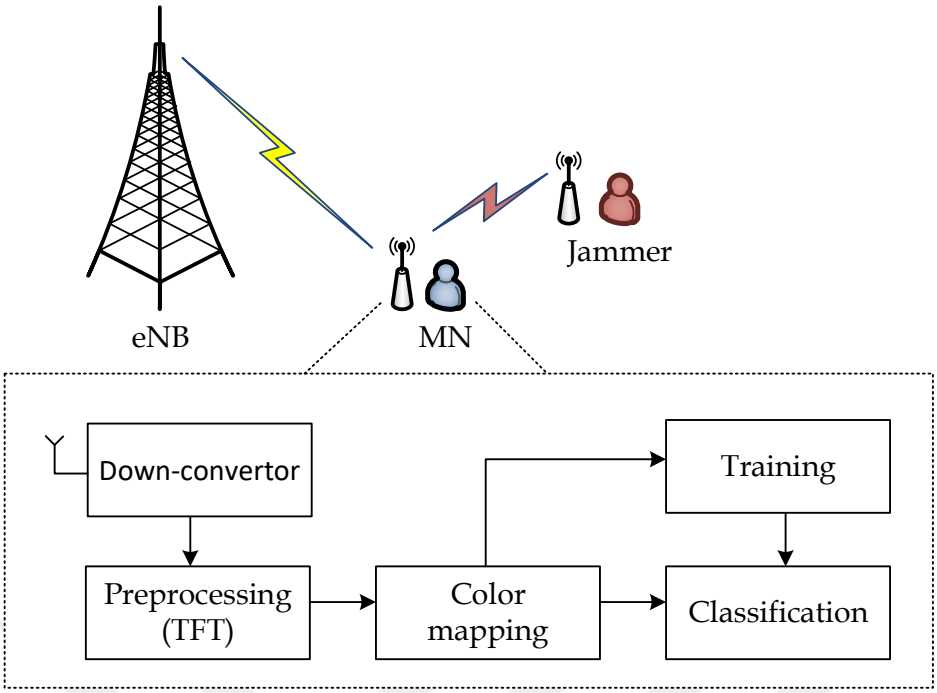

Figure 3.3 : A base station (eNB), monitoring node (MN) and a jammer in a LTE cell. The received signal from the MN would be the superposition of LTE signal and jammer signal.

that attack the LTE characteristics explained above. Therefore four different types of jamming cases are considered: (i) there is no attack to the system, (ii) the complete frequency band is attacked, (iii) only the synchronization signals are attacked and (iv) only reference signals are attacked. Following titles are the specific names for these different jammer types respectively.

3.1.2.1 Barrage jamming

Barrage jamming (BJ) is the most frequent form of the jamming attack. Its detection [82], [83] and mitigation [84] have been exhaustively discussed in the literature. In BJ, the attacker continuously transmits band limited noise over the entire spectrum of the receiver. Note that with enough observations over the network, attacker may transmit noise only during the transmission of a specific signal [85]. This approach requires more complexity or additional information with respect to the network, but it reduces the energy consumption of the jammer [59]. With additional knowledge on the network, BJ attacks can further be diversified, but in this work BJ is assumed to transmit noise over the entire bandwidth of the receiver. Discrete time representation

of the transmitted BJ signal is

$$j_b(l) \sim \mathcal{CN}(0, \sigma_j^2),$$

where $\mathcal{CN}(\mu, \sigma^2)$ denotes the complex normal distribution with mean μ and variance σ^2. Even though BJ does not require any prior knowledge of the network, in order to initiate the following attacks, an attacker will require to extract some information about the LTE network.

3.1.2.2 Synchronization signal jamming

Unlike BJ, in synchronization signal jamming (SSJ), the attacker should be aware of the locations of the resource blocks as explained above. Although there are several ways of jamming synchronization signals, we assume that the attacker generates an LTE frame similar to the transmitter, but maps only PSS and the SSS signals to related resource elements. The rest of the packet is filled with zeros. The process can be shown as

$$a_{k_s,l_s}^{j_s} \sim \mathcal{CN}(0, \sigma_s^2),$$

where k_s and l_s are also given above. Remaining resource blocks of the jammer are filled with zeros: $a_{k_{rf},l_{rf}}^{j_s} = 0$ and $a_{k_m,l_m}^{j_s} = 0$. In this case, discrete time representation of the transmitted SSJ signal can be denoted as

$$j_s(l) = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} a_{k,l}^{j_s} e^{j2\pi kl/K} \omega\left(l/K + K_{CP} - 1\right). \tag{3.4}$$

Note that, the SSJ signal transmission should be synchronized with the eNodeB, and the indices, k_s and l_s, should also be known by the attacker.

3.1.2.3 Reference signal jamming

In reference signal jamming (RSJ), the attacker is assumed to know the locations of the reference signals. For RSJ signal generation, we have followed an approach similar to SSJ signal generation. We assumed that the attacker generates the random signal only on the reference signals locations and maps zeros on the data locations similar to the SC. The mapping can be shown as

$$a_{k_{rf},l_{rf}}^{j_{rf}} \sim \mathcal{CN}(0, \sigma_r^2),$$

where k_{rf} and l_{rf} are also given above. The remaining resource elements are filled with zeros; $a_{k_s,l_s}^{j_{rf}} = 0$ and $a_{k_m,l_m}^{j_{rf}} = 0$. In this case, discrete time representation of the

RSJ signal is

$$j_{rf}(l) = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} a_{k,l}^{j_{rf}} e^{j2\pi kl/K} \omega\left(l/K + K_{CP} - 1\right). \tag{3.5}$$

3.1.3 Received signal model

In an ideal case, received signal is affected by the channel attenuation and the noise. Considering a flat fading discrete time uncorrelated channel in the existence of a jammer, we can obtain the received baseband signal as

$$r(l) = \frac{s(l) \cdot h(l)}{\sqrt{d_1^{PL}}} + \frac{j_i(l) \cdot g(l)}{\sqrt{d_2^{PL}}} + \eta(l), \tag{3.6}$$

where $j_i(l)$ is the jamming signal of the jammer type i, while $i \in \{b, s, rf\}$ respectively for BJ, SSJ and RSJ. $r(l)$ is the received signal, $h(l)$ is the channel coefficient for the transmitted signal, $g(l)$ is the channel coefficient for the jammer signal, d_1 is the distance between the eNB and the MN, d_2 is the distance between the jammer and the MN, PL is the path loss exponent and $\eta(l)$ is the noise at the receiver. The channel coefficients are assumed to be complex valued random variables with $h(l), g(l) \sim \mathcal{CN}(0, \sigma^2)$ distribution, and the noise at the receiver is assumed as $\eta(l) \sim \mathcal{CN}(0, \sigma_n^2)$. Resulting signal-to-noise (SNR) ratio expression can be obtained as the following

$$\text{SNR} = \frac{P_s \sigma^2}{\sigma_n^2 d_1^{PL}}, \tag{3.7}$$

where P_s denotes the power of the transmitted signal. Another important parameter for jammer analysis is the signal-to-jamming ratio (SJR) and given as the following

$$\text{SJR} = \frac{P_s}{P_{j_i}} \left(\frac{d_2}{d_1}\right)^{PL}, \tag{3.8}$$

where P_{j_i} is the jamming power. In the following section, jammer signal identification model will be presented. We aim to identify the jammer type, i, by applying different transformation and classification methods to the $r(l)$.

3.2 Preprocessing with Time-Frequency Transformations

The general system structure and alternative operations for the system model are presented in this section. Identification system model is shown in Figure 3.3. The pre-processing part of the identification system is carried out via a TFT which

represents a 1D time signal into 2D time-frequency plane. After transformation, 2D output signal can be named as time-frequency representation (TFR). Before classification, the TFR of the received signal is saved as an image. The second part of the identification system includes the automatic classification for the generated images.

Time-frequency transformations (TFT) allow analyzing the temporal changes of the signal, and they give a compact representation on the time-frequency domain. In his famous overview paper [86], Cohen explains this advantage with an example on sunlight analysis. If the collected data is Fourier transformed, the power density spectrum does not show that the spectral composition of the signal is very different in sunset or sunrise in comparison with other time periods. Therefore, one can suggest that we can capture measurements in predefined time intervals and Fourier transform these samples. By extending or narrowing the time period we can find temporary changes in our data. This example forms the fundamental idea behind the STFT or its magnitude square representation spectrogram, and it also indicates the trade-off between time and frequency resolution. Other Fourier based TFT's can be considered improvements on the spectrogram to provide effective solutions to tackle this trade-off.

3.2.1 Short time Fourier transformation

In spectrogram, first a windowing function $\omega(l)$ is applied to the received signal to divide it into short time periods. Then fast Fourier transform (FFT) is applied separately to each time period. We can show the STFT of the $r(l)$ as

$$R_s(c,\kappa) = \frac{1}{\sqrt{2\pi}} \sum_{l=(-L+1)/2}^{(L-1)/2} r(l)\omega(c-l)e^{-j2\pi cl/L}, \tag{3.9}$$

where $c = t \times f_s$ indicates the sample number, and κ is the frequency variable. For our case, frequency values of the related subcarriers as $\kappa = k\Delta_f$ where Δ_f shows the frequency difference of the consecutive subcarriers. $\omega(l')$ is a discrete time rectangular function with length $\tilde{l}$ having a unit magnitude for $(-\tilde{l}+1)/2 \leq l' \leq (\tilde{l}-1)/2$. Note that $\omega(l')$ can also be generated as a Hamming or Hanning function, but we only consider the rectangular form because of its simplicity. Then the spectrogram of the signal is

$$P_s(c,\kappa) = |R_s(c,\kappa)|^2. \tag{3.10}$$

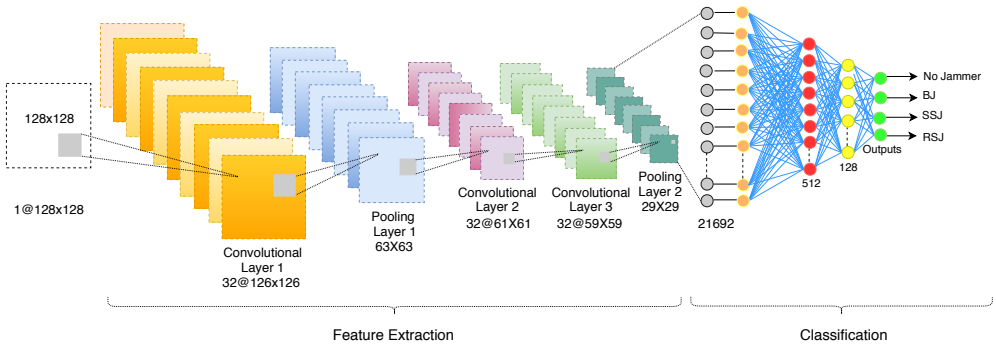

Figure 3.4 : A detailed DCNN block diagram. $F@\Upsilon_1 \times \Upsilon_2$ where F represents the number of feature maps in the corresponding layer, whereas $\Upsilon_1 \times \Upsilon_2$ denotes the size of an individual feature map.

Time resolution of the spectrogram is $\delta_{t_S} = (\tilde{l} - 1)/(2f_s)$ [87]. Using this approach, we can obtain the frequency content of the signal for short time periods.

3.2.2 Choi-Williams transformation

First proposed in [88], Choi-Williams Transform (CWT) actually aims to overcome difficulties on another TFT named as Wigner-Ville transformation (WVT). In WVT, multitone signals generate high power cross terms that should be zero. CWT remarkably reduces the cross-terms without worsening the spectral representation. The CWT of the received signal is

$$R_{cw}(c,\kappa) = 2\sum_{l=0}^{2L-1} S'(c,l)e^{-j2\pi\kappa l/L}, \tag{3.11}$$

where $S'(c,l)$ is

$$S'(c,l) = \begin{cases} S(c,l)\,, & 0 \le l \le L-1 \\ 0\,, & l = L \\ S(c,l-2L)\,, & L+1 \le l \le 2L-1 \end{cases} \tag{3.12}$$

and

$$S(c,l) = \omega(\nu)\sum_{\nu=-L/2}^{L/2} \frac{1}{\sqrt{4\pi l^2/\sigma}} e^{-\frac{(\nu-1)^2}{4\pi^2/\sigma}} r(l+\nu)r^*(l-\nu), \tag{3.13}$$

Similar to spectrogram, we can define the absolute square of the transform:

$$P_{cw}(c,\kappa) = |R_{cw}(c,\kappa)|^2. \tag{3.14}$$

The effective time resolution is $\delta_{t_{cw}} = (\tilde{l} - 1)/(2f_s)$ [87]. Finally the resulting TFR $P_{cw}(c,\kappa)$ is color-mapped and saved for the further classification process.

3.2.3 Gabor wavelet transformation

In the presence of a smart jammer, the received signal $r(l)$ would show varying characteristics over time. While the attacked parts exhibit abrupt changes in the signal, non-attacked parts would have smooth changing components. These abrupt changes become apparent in higher frequency components. Increasing the scaling factor of the wavelet provides a better representation of the abrupt changes on the signal. We can construct a TFR from the received signal by sequencing the arrays obtained from different scale factors of wavelet transformations.

Among various wavelet bases, Gabor functions provide the optimal resolution in both time and frequency domains by using a Gaussian shaped windowing function [86]. Gabor wavelet transformation function can be expressed as

$$\Psi(c, f) = \frac{f}{\gamma\sqrt{\pi}} e^{\frac{f^2}{\gamma^2}c^2} e^{j2\pi cf}, \tag{3.15}$$

where f/γ is the scaling factor and γ is the quality factor and $f = 1, 2, \cdots, F$. The transformation is carried out via convolving the received signal $r(l)$ with defined Gabor wavelet transformation function:

$$R_{gm}(c, f) = r(l) * \Psi(c, f), \tag{3.16}$$

where $*$ denotes the convolution operation and f shows the different frequency values on which wavelet transform is applied. The TFR, also named as scalogram for Gabor-wavelet transform is

$$P_{gm}(c, f) = |R_{gm}(c, f)|^2, \tag{3.17}$$

where the time resolution of the scalogram is defined as $\delta_{t_{gm}}(f) = \gamma/f$ [87]. P_s, P_{cw} and P_{gm} are TFR's of the received signal and they are constructed as 2D matrices. They are stored in the memory for training the classification methods. The values in these magnitude level representations are color-mapped by their amplitude values. The color-mapping operation can be seen as a quantization of the magnitude levels into different color levels.

3.3 Identification Methods and Numerical Analysis

3.3.1 Data acquisition

In this work, we consider SVM and DCNN for classification purposes due to their high performances as reported in the literature [89], [90].

LTE signal generation and transmission scheme is modeled with the MATLAB LTE Toolbox. It offers a realistic modeling and testing framework for LTE networks. The parameters regarding to complete system can be found in Table **??**. The channels between radio nodes are assumed as flat fading with unit gain Rayleigh distribution.

Four different jamming scenarios are considered, and the classification labels are: the absence of a jammer (no jamming), BJ, SSJ, and RSJ. The jamming is assumed to be active throughout a packet transmission duration. The transmission is repeated 5000 times for each scenario. In each scenario, the received signal r is generated in accordance with (6).

Same process is repeated considering four different combinations of the jammer and the MN locations. The location cases and their equivalent SNR/SJR values with error vector magnitude (EVM) values are given in Table 3.2. As indicated in [91], EVM is an error metric that strongly depends on the signal-to-noise/interference ratio (SINR). C_1 corresponds to the case when the jamming signal does not significantly reduce the transmission. On other cases, EVM values are higher than 70%, indicating that the transmission is blocked by the jamming signal. The transmitted signal is assumed to have unit power. SNR/SJR values are obtained by changing the variance of the noise signal and jamming power. For all location cases, radio nodes are assumed to be stationary.

At the second step of the analysis, three different TFT's are applied on the received signals in order to compare their performances with classifiers. Figure 3.6 shows the images obtained from the P_s, P_{cw} and P_g on logarithmic scale. These images are saved in portable network graphics (png) format and fed into two classifiers. All images for a given transformation are cropped to the same size in order to eliminate misleading features.

Table 3.1 : Parameters for numerical analysis.

Step	Parameter	Value
Signal Transmission	# of subcarriers	140
	# of subframes	10
	Duplex Mode	FDD
	Cyclic Prefix	Normal
	Modulation Type	QPSK
	d_1	1, 1.5
	d_2	1, 1.5
	SNR (dB)	0, 5
	SJNR (dB)	-5, 0, -5
	Channel Fading	Rayleigh
	Channel Gain	1
	Sampling Frequency	1.92 MHz
	PL	4
	n_1	128
	n_2	128
Time-Frequency Transforms	σ_{cw}	100
	γ	$5/2\sqrt{2\log 2}$
	F	256
	δ_{t_s} , $\delta_{t_{cw}}$	0.1024
	$\delta_{t_{gm}}$	$3.221/f$
	L	512
	Colormap	Jet
PCA-SVM Classification	PCA component number	1000
	C	1
	Kernel function	linear
	Gamma value	0.001
	Batch size	64
DCNN Classification	Epoch number	40
	Convolution layer stride	(1,1)
	Optimizer	SGD
	Learning rate	0.1
	Loss Function	Categorical cross entropy
	Total parameters	13,867,988
	Trainable parameters	13,866,508
	Non-trainable params	1,480

As it can be seen from Figure 3.6, the existence of BJ is observable for every pre-processing method. Considering spectrogram, differences among no-jamming, SSJ and RSJ are not apparent due to the single resolution property as discussed in Section III. Since most of the energy is located near the low frequencies, the abrupt changes occuring in high frequencies are not observable from the generated

Table 3.2 : Description of location cases and their distances, SNR, SJR and SJNR equivalents

Cases	d_1	d_2	SNR	SJR	EVM (%)			
					No	BJ	RSJ	SSJ
C_1	0.5	1.5	10	10	16.58	18.25	17.69	17.14
C_2	1	1	5	0	23.14	95.82	85.18	77.64
C_3	1	1.5	5	5	23.14	86.15	69.22	71.92
C_4	1.5	1	0	-5	30.58	99.72	92.19	79.14
C_5	1.5	1.5	0	0	30.58	89.18	83.26	75.84
C_6	1.5	0.5	0	-10	30.58	99.16	96.24	98.47

images. For CWT, the difference between no-jamming and BJ are not as apparent as spectrogram and Gabor wavelet transform, because of the obscure background. As discussed above, the ambiguity arises from the cross-term effects is still observable, despite the cross-term minimizing effects of CWT. Yet, classification results in the following of this paper show that the features in the images are still separable by the classifiers.

In Gabor wavelet transform the images of four different jamming scenarios are separable from each other. For the no-jamming scenario, energy is mostly localized in the low-frequency values. As it can be seen from the turquoise colored region, received signal contains abrupt changes due to the channel fading and the noise. These changes are closely located energy content on the time-frequency domain. Hence, these changes are represented with the same color. In BJ scenario, jammer attacks are spread over the entire time-frequency plane.

As the power of the jammer increases, the abrupt changes on the signal become dominant over the time-frequency plane. In SSJ and RSJ, the attacked locations on time-frequency plane become apparent due to the multi-resolution property of wavelet transform. The emitted energy from SSJ and RSJ becomes apparent in Gabor wavelet transform since the wavelets are localized on time-frequency plane.

3.3.2 Classification

In classification part, we compare two machine learning algorithms; DCNN and SVM.

3.3.2.1 Support vector machine

Support vector machine (SVM) is a supervised learning method and was firstly developed for binary classification. The main idea behind SVM is finding the optimal decision boundary by discriminating the feature vectors. The decision boundary is also named as a hyperplane and for optimality it has to maximize the separation between two data classes. The method starts with selecting a hyperplane in the feature space and is defined as follows;

$$\mathbf{w}^T\mathbf{x} + b = 0, \tag{3.18}$$

where **x** is the feature vector or in this study output of PCA, **w** is the support vector, $(\cdot)^T$ denotes transpose operation and b is the bias term. SVM separates the data classes by maximizing the margin defined as minimum distance of any data points to the decision boundary. If the data are not linearly separable, optimization problem is defined as below to find the maximal margin hyperplane;

$$\begin{aligned} \min_{\xi,\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + \lambda \textstyle\sum_{p=1}^{m} \xi_p \\ \text{subject to} \quad & y^{(p)}(\mathbf{w}^T\mathbf{x}^{(p)} + b) \geq 1 - \xi_p \\ & \text{and} \;\; \xi_p \geq 0 \;\; p = 1, ..., m. \end{aligned} \tag{3.19}$$

where m is the number of training samples, $\mathbf{x}^{(p)}$, $\mathbf{y}^{(p)}$, ξ_p are related to p^{th} training sample and $y^{(p)}$ is a class label, which has one of only two values, either -1 or 1. ξ_p is a slack variable. The first term minimizes the distance to the closest data point, and the second term reduces the number of misclassified points. Optimization problem is constructed as the Lagrangian as given below

$$\begin{aligned} &\mathcal{L}(\mathbf{w}, b, \mathbf{a}) = \tfrac{1}{2}\|\mathbf{w}\|^2 + \lambda \textstyle\sum_{p=1}^{m} \xi_p \\ &-\textstyle\sum_{p=1}^{m} a_p[y^{(p)}(\mathbf{w}^T\mathbf{x}^{(p)} + b) - 1 + \xi_p] - \textstyle\sum_{p=1}^{m} \mu_p\xi_p, \end{aligned} \tag{3.20}$$

where $0 \leq a_p \leq \lambda$, λ is a hyperparameter called as penalty of the error term or regularization term, and $\mu_p \geq 0$ is the Lagrange term. When the above equation is solved, it turns into following form;

$$f(x) = \textstyle\sum_{p=1}^{m} a_p y^{(p)} \mathcal{K}(\mathbf{x}^{(p)}, \mathbf{x}) + b, \tag{3.21}$$

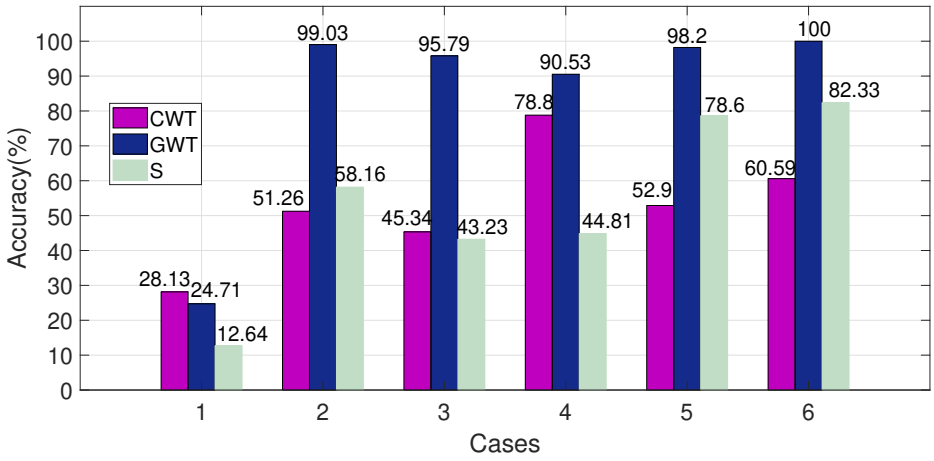

Figure 3.5 : Different time-frequency analysis comparison for SVM classification.

where $\mathcal{K}$ is a kernel function, and $f(x)$ is called as decision function or score function which is used to compute score for each input vector. According to the output of (3.21), SVM predicts the class of each input.

DCNN architecture is implemented in KERAS Python library interface that works with Tensorflow back end, while SVM is implemented in Phyton. We train each model with 1000 images and the size of images are 128×128. On parameter selection, we utilize the accuracy of classification as the main metric. As shown in Figure 3.4, DCNN architecture consists of three convolution layers, two pooling layers and three full-connected layers. We choose 2×2 for pooling size and 3×3 filter size for convolutional kernels. After each layer we apply batch normalization and ReLu function, except the output layer in which softmax function has been used. Also we employ dropout layer of 40% rate to prevent over-fitting. On the training of the model we use cross entropy $J(\Theta)$ as a cost function [92]:

$$J(\Theta) = \frac{-1}{m} \sum_{p=1}^{m} \sum_{n=1}^{M} y_n^{(p)} \log(q_n^{(p)}). \tag{3.22}$$

where M is the number of classes and m is number of instances. $q_n^{(p)}$ is the estimated probability that the instance $x^{(p)}$ belongs to class k. $y_n^{(p)}$ is equal to 1 if the target class for the p^{th} instance is k. Otherwise, it is equal to 0.

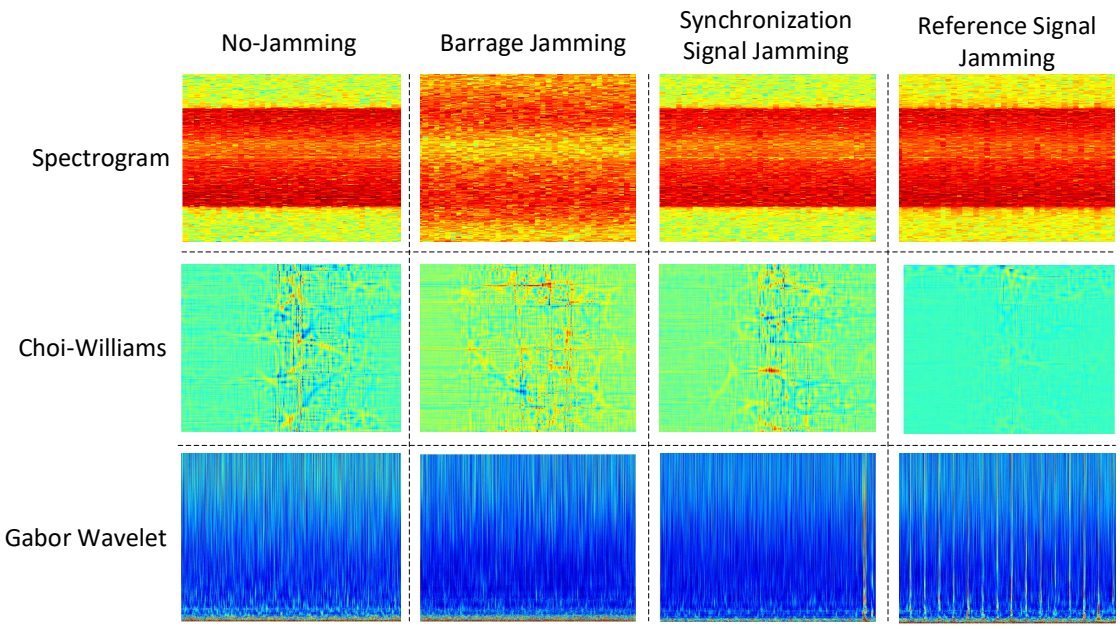

Figure 3.6 : Spectrogram, CWT and Gabor wavelet transform applied images. (from left to right no-jamming, BJ, SSJ and RSJ; from up to down spectrogram, CWT, Gabor wavelet transform)

In the next step, we compute the gradient vector for every class and use stochastic gradient descent optimizer to find the parameter matrix Θ which minimizes the cost function.

Besides DCNN, we also applied SVM to compare their classification accuracies. Unlike DCNN, SVM requires an explicit feature extraction step to realize classification. Principal component analysis (PCA) is used as an effective and robust method for feature extraction [93]. The central idea of PCA is to reduce the dimensionality of the data set in which there are a large number of interrelated variables, while retaining as much as possible of the variation present in the data set [94]. After feature extraction through PCA, the dimension of the input vector $\mathbf{x}$ fed to the classifier is equal to the number of extracted principle components. SVM is applied to the multi-class classification problem by using the "one against the rest" approach [53]. The fundamental idea underlying this approach is to realize the multi-class classification by using multiple binary SVMs collectively. After completion of the classification model training, 18400 test images are used to determine the performance of the algorithms for identifying the jammer types considering a single position case. For every position case,

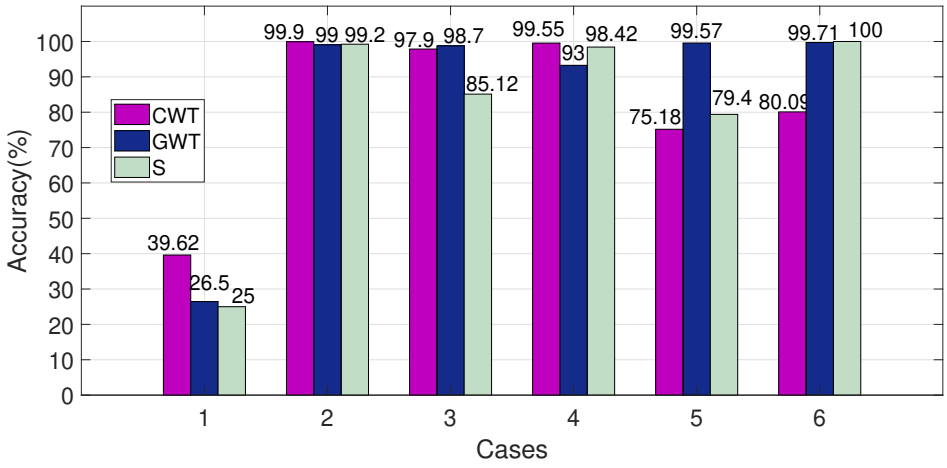

Figure 3.7 : Different time-frequency analysis comparison for DCNN classification.

same training and test process is repeated. Classification accuracy results obtained by SVM and DCNN for different simulation setups can be seen from Figure 3.5 and Figure 3.7.

Figure 3.5 shows the classification accuracies when SVM is selected as the classifier. The groups at horizontal axis show the case numbers, where their SNR/SJR equivalents can be found from Table 3.2. Considering C_1, the identification accuracies are lower than other cases. In this case, jamming signal is not detected by the identification system. However note that, the jamming signal cannot block the transmission anyways, which makes identification obsolete at the first place as the identification accuracies are lower than other cases. Considering other cases, the figure shows that the Gabor wavelet transform outperforms other classification methods with a minimum of 90% accuracy. As the signal and the jamming powers become closer to each other, in other words as the SJR approaches to zero, the classification accuracy of the Gabor wavelet transform increases. In high SNR values, classification accuracy is higher than the low SNR values considering Gabor wavelet transform. Considering C_6, the power of the jamming signal is much more powerful than the message signal. In this case, jammer signal becomes dominant over the time-frequency plane, where identification accuracy improves for all pre-processing methods.

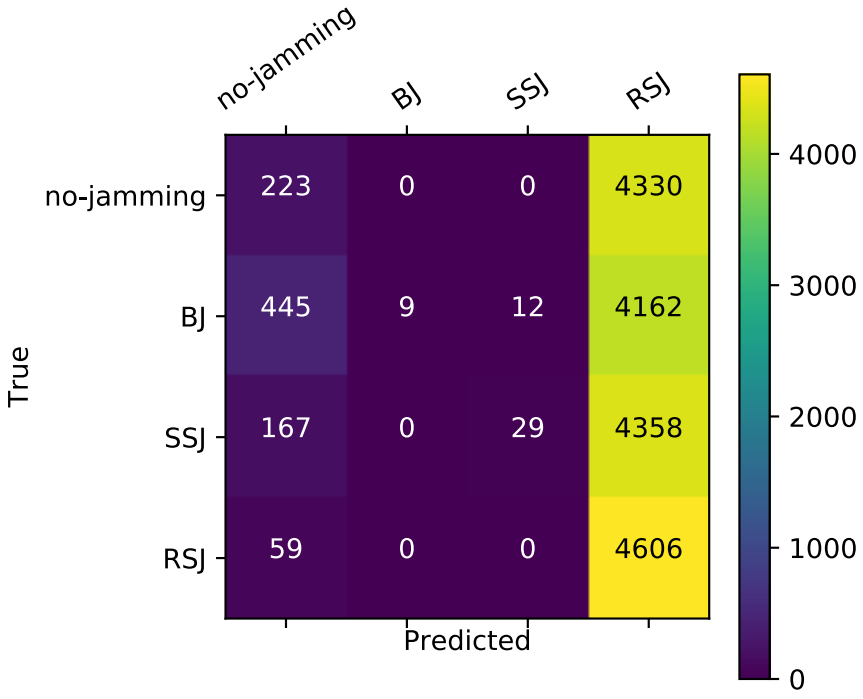

Figure 3.8 : DCNN classifier's confusion matrix for C_1

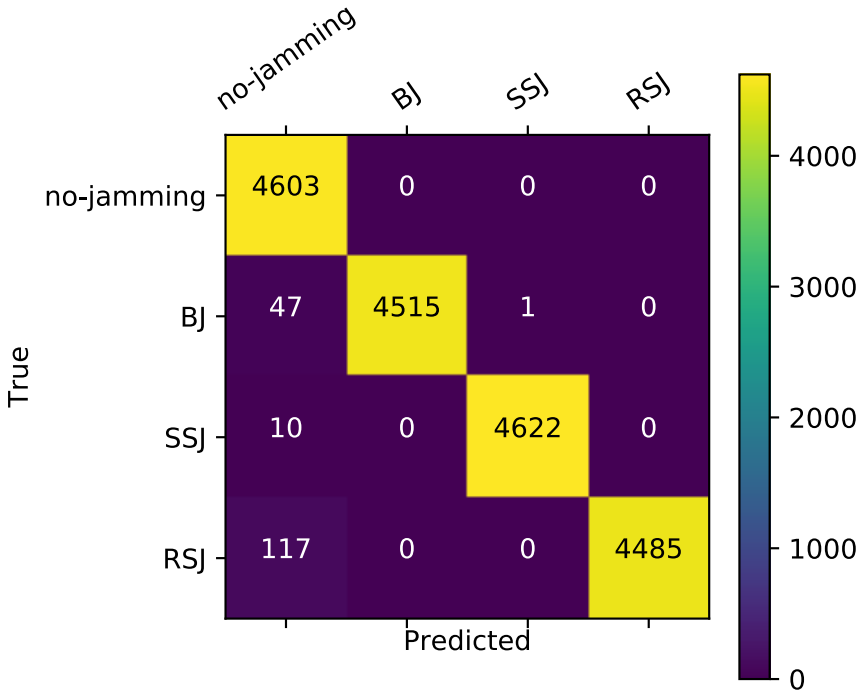

Figure 3.9 : DCNN classifier's confusion matrix for C_2

Interestingly, other transform methods do not follow the same accuracy behavior as the Gabor wavelet transform. Both Choi-Williams transform and spectrogram work better

in the low SNR region. Especially Choi-Williams works best when SJR is low. This is plausible because when the power of the jamming signal is higher than the message signal, the jamming patterns become more apparent.

Figure 3.7 shows the classification accuracies when DCNN is selected as the classifier. Similar to the results of SVM classifier, the jammer activity cannot detected by the DCNN classifier in C_1. In other cases, all transformation methods perform better than the SVM classifier in this case. Gabor wavelet transform gives the best accuracy results overall with a minimum 93% accuracy. Similar to the SVM classification, Gabor wavelet works better as the SJR approaches zero. Although the remaining transformation techniques show high performance for C_2 and C_4, their performance are steadily reduced in C_3 and C_5.

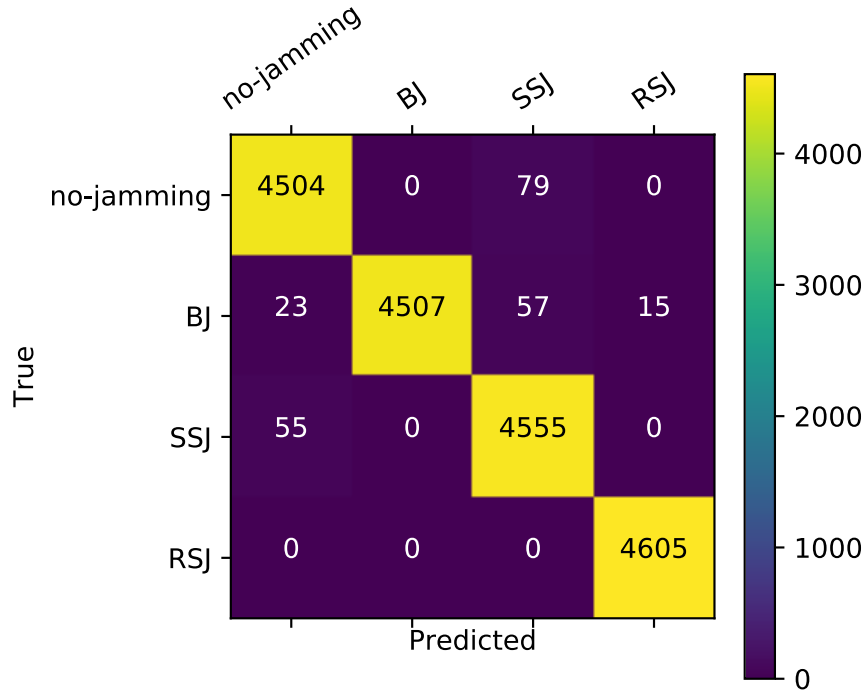

Figure 3.10 : DCNN classifier's confusion matrix for C_3

If we evaluate the overall success of the classification methods, the first noticeable feature is the superior jammer classification performance of the DCNN for all different transforms and setups. SVM can essentially be considered as a shallow neural network [95]. Although SVM has respectable performance in certain cases, DCNN with its

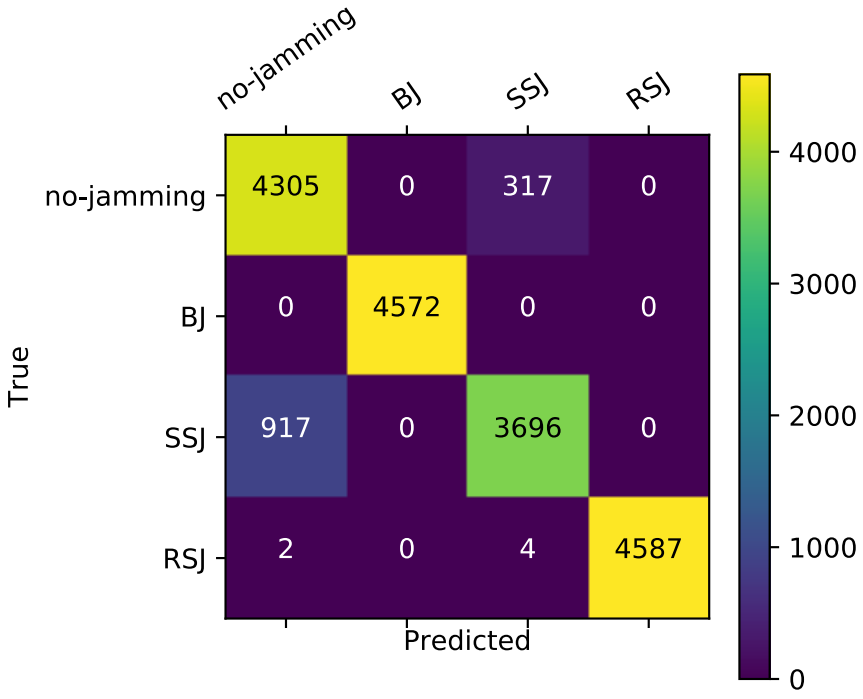

Figure 3.11 : DCNN classifier's confusion matrix for C_4

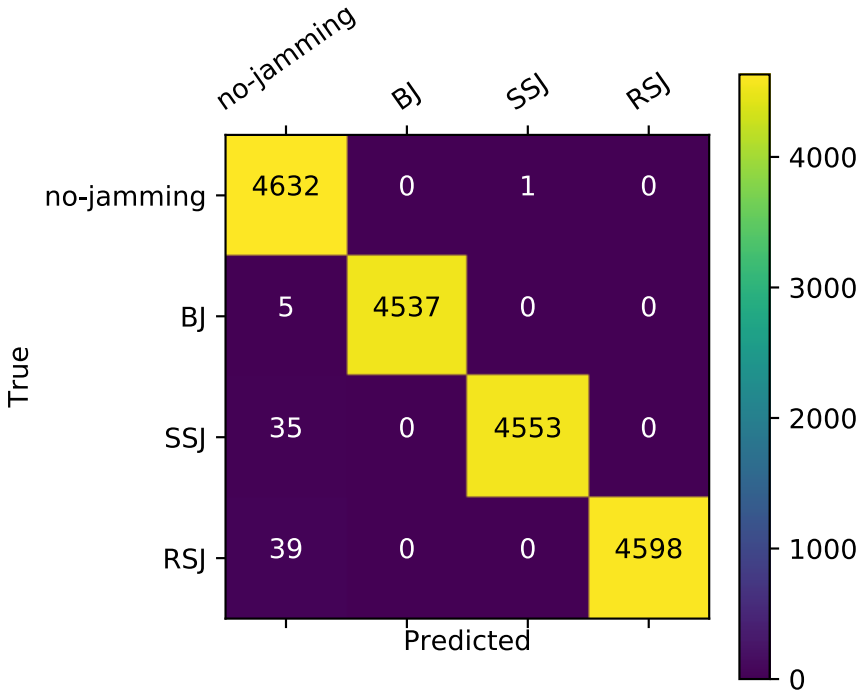

Figure 3.12 : DCNN classifier's confusion matrix for C_5

hidden layers provides consistent and robust performance for all differing setup choices and TFT methods.

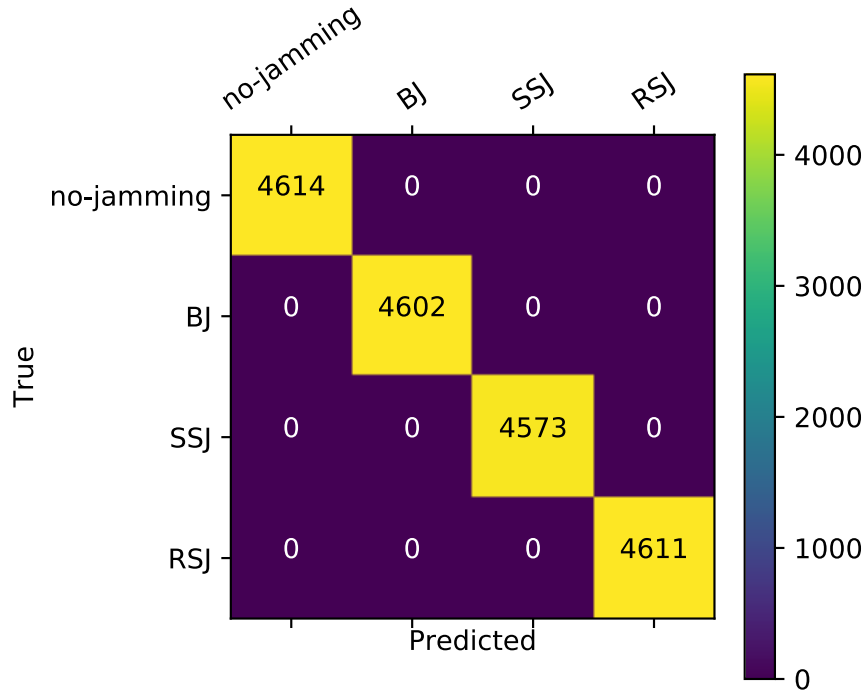

Figure 3.13 : DCNN classifier's confusion matrix for C_6

Considering the results in Figure 3.5 and 3.7 together, best working identification method would be the combination of Gabor wavelet transform and DCNN classifier. As expected from the previous sections, Gabor wavelet transform gives the best representation among other transformation methods because the wavelet characteristics alter the time-frequency plane resolution and highlight abruptions resulting from jamming.

Figures 3.8, 3.9, 3.10, 3.11, 3.12, 3.13 show the confusion matrices of Gabor wavelet transform when used with the DCNN classifier. These confusion matrices describe the performance of the classification model by demonstrating the comparison of model prediction and true class for the test dataset. It is a quality measure for the class by class identification performance of the proposed method. In C_1, DCNN can not accurately differentiate between the classes, because the very low powered jamming signal can not get detected on the time-frequency plane. In C_2, the most commonly confused classes are RSJ and no-jammer. One important observation from all of the cases is that there is almost no confusion between the different jammer types.

3.4 Conclusion

In this study, a novel jammer identification system is presented and the identification performance of the system over various jamming attack cases are determined. The proposed system can differentiate three main types of jamming attacks: barrage jamming, synchronization signal jamming and reference signal jamming cases along with the absence of jammer case. Even though barrage jammer can be easily detected as an unexpected random noise combined with the original signal, other types of attacks are harder to detect and to tolerate since they effectively hide in the targeting time-frequency plane and possibly disrupt communication with a low power. The proposed system model is composed of a wavelet-based pre-processing step and a deep learning based classification stage. We consider an LTE downlink communication scenario, where the effectiveness of the wavelet transform based approach is clearly observed, even in the presence of smart jamming attacks. Considering different pre-processing methods, the superior performance of DCNN is observed in comparison to SVM.

4. JAMMER DETECTION BASED ON ARTIFICIAL NEURAL NETWORKS: A MEASUREMENT STUDY

In parallel with the rapid development of information technologies, the comprehensiveness of communication systems expands and usage rates increase. All of these advances force wireless communication systems to target higher data rates and require more reliable communication systems. Orthogonal frequency division multiplexing (OFDM) has a design of the orthogonal subcarriers in the frequency domain and provides a high spectral efficiency. In the fourth and the fifth generation communication standards, OFDM is selected due to many advantages such as robustness against interference and noise, and due to its ability to eliminate multipath effects. However, it is still vulnerable to jamming attacks. At this point, detection and identification of the jamming signals become critical to improve countermeasures and prevent the jamming attacks.

Many studies investigate the solutions to avoid or overcome the jamming threats on OFDM-based communication systems. In [96], an adaptive filtering based algorithm is proposed to suppress the impact of the jamming signal and interference. The authors in [97] propose two pilot tone randomization scheme to mitigate the pilot jamming attack. This scheme shows better performance in pilot jamming attacks but conventional deterministic scheme outperforms both random schemes in the absence of jamming. At this juncture, the detection of a jammer presence and the type of attack have a significant importance to facilitate preventive measures. In [56], the existence determination of the jammer is defined as a binary hypothesis test that depends on random matrix theory and a rejection method are presented. An approach in [98], based on the extraction of the statistics from the jamming-free symbols of the direct sequence spread spectrum synchronizer to detect jammer. In [99] authors propose an approach hinged on unsupervised learning techniques relating to jammer detection and identification issues, [100] presents another solution based on machine learning techniques for the attacks to IEEE 802.11 networks and compares methods in terms of their performance. In [89],

an adversarial machine learning approach is introduced to launch jamming attacks on wireless communications and a defense strategy is presented. In [57], a convolutional neural network (CNN) and support vector machines are utilized for the identification of jammers. In general, the presented approaches are concentrates on prevalent metrics such as received signal strength indicator, packet delivery ratio or under the assumption of jammer attack is exist and they do not include test bed verification.

In this study, wide-band and narrow band jammer attacks are considered in both the presence and the absence of the legitimate transmitter. Firstly, we aim to detect the existence of jammer and to identify the type of attacks by implementing state-of-the-art techniques of artificial neural networks: DCNN and deep recurrent neural network (DRNN). Secondly, we show the detection and identification merit of deep learning techniques on real-time data and attack scenarios that are implemented on software defined radios nodes using an OFDM based communication system. Also, we present the results via datasets for different positions and gains of transmitter and jammer nodes. Thirdly, pre-processing methods based on time-frequency transformation (TFT) used with the purpose of improving the representation capability in case of wideband and narrowband jammers. The obtained results show that the presented DCNN method detects and identifies the jamming attacks with a higher forecasting performance than DRNN regarding all considered scenarios.

4.1 System and Attacker Model

General steps of OFDM based communication system can be summarized as: the information bits are converted into the modulation symbols and the reference symbols also known as pilot symbols are inserted in certain intervals to estimate channel state information and these symbols are passed to the time domain by the inverse fast Fourier transform (IFFT) operation. Following this process, the cyclic prefix (CP) is inserted to eliminate inter-symbol interference effects at the destination. Generated symbol sequences are sent and the received signal can be modeled in the frequency domain as follow:

$$\mathbf{Y} = \mathbf{H_1}\mathbf{X} + \mathbf{W}, \tag{4.1}$$

where $\mathbf{X} = [X(0),\ X(1),\ \cdots,\ X(N-1)]^T$ and $\mathbf{Y} = [Y(0),\ Y(1), \cdots,\ Y(N-1)]^T$ denote the transmitted and received symbol sequences. $\mathbf{H_1} = \mathrm{diag}\{H_1(0),\ H_1(1),\ \cdots, H_1(N-1)\}$ is channel matrix and N is the number of subcarriers. $\mathbf{W}$ is an independent and identically distributed (i.i.d.) additive white Gaussian noise (AWGN) with distribution $\mathcal{CN}(0,\ \sigma_n^2)$ where σ_n^2 is the variance of noise. When the jammer is present, the received signal is defined as in below:

$$\mathbf{Y} = \mathbf{H_1}\mathbf{X} + \mathbf{W} + \mathbf{H_2}\mathbf{J}, \tag{4.2}$$

where $\mathbf{J}$ represent the jamming signal and H_2 is the corresponding channel matrix between the jammer and the receiver.

4.1.1 Barrage Jamming

A barrage jammer (BJ) attacks to the whole transmission channel using strong noisy signals and disrupt target information [97]. When BJ signals are generated as a white Gaussian noise with distribution $\mathcal{CN}(0,\ \sigma^2)$, the jamming symbol sequences can be defined at frequency spectrum as follows:

$$\mathbf{J_N} = [J_N(0),\ J_N(1),\ \cdots,\ J_N(N-1)]^T \tag{4.3}$$

The cases where active BJ is present, the jamming signal in the equation (4.2) is $\mathbf{J} = \mathbf{J_N}$.

4.1.2 Reference Signal Jamming

A reference signal jammer (RSJ) disrupts the target information by attacking certain parts of a transmitted signal [97]. RSJ can be considered as a narrowband attack, focusing reference signals, affecting the quality of channel estimates at the transmitter. In contrast to the BJ, since RSJ attacks the pilot signals in an OFDM symbol, it utilizes its power in a more effective manner. Also, this scheme is referred to as the smart jammer because of its effective power consumption. The comb-type pilot frequency and the sequence of pilot locations are denoted with ρ and φ, respectively:

$$\varphi(n) = \begin{cases} \sqrt{\rho}\,, & \mathrm{mod}(n, \rho) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

$$\mathbf{\Theta} = \mathrm{diag}\{\varphi(0),\ \varphi(1),\ \cdots, \varphi(N-1)\}. \tag{4.5}$$

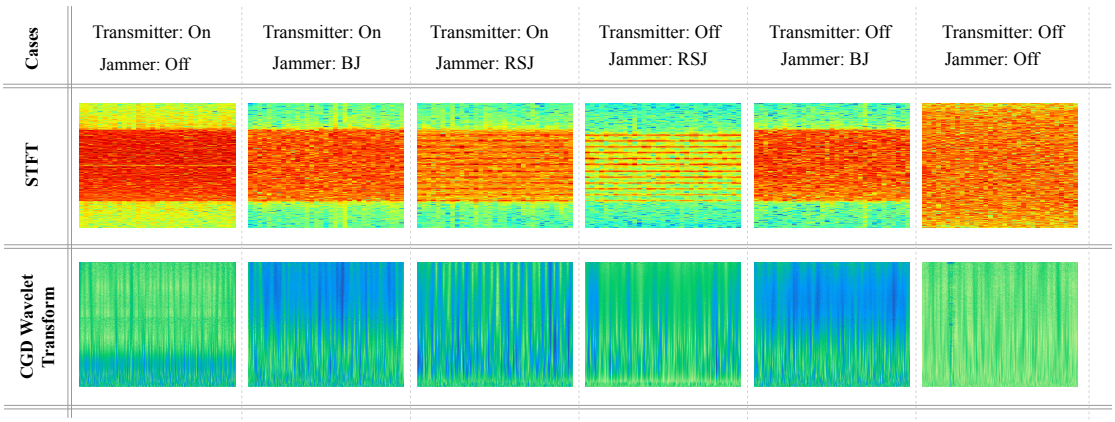

Figure 4.1 : CGD wavelet transform and STFT based received signal representations regarding SC-1.

When RSJ is exist, the jamming signal is $\mathbf{J} = \mathbf{\Theta}\ \mathbf{J_N}$.

4.2 Pre-processing Methods

An appreciable approach is using time-frequency transform methods to provide a discriminative illustration of the raw signals. In this study, two different TFTs are applied to the received signal. Our main motivation to use pre-processing techniques is to improve identification and detection performances.

4.2.1 Short-Time Fourier Transform

The short time frequency transform (STFT) is the discrete Fourier transform of the signal along fixed window lengths. It allows to focus on changes in frequency or time domain depending on the window width. The STFT of $x(n)$ is defined as follows:

$$\Psi(n,k) = \sum_{m=0}^{L-1} x(n+m)w(m)e^{-j\frac{2\pi}{D}km}, \tag{4.6}$$

where $x(n)$ is a discrete signal, D is the number of samples, L is the length of the window, $w(m)$ is a window function which slides on the discrete signals and power spectrum, $\aleph(n,k)$ can be defined as follows:

$$\aleph(n,k) = |\Psi(n,k)|^2 \ . \tag{4.7}$$

4.2.2 Complex Gaussian Derivative Wavelet Transform

Complex Gaussian derivative (CGD) wavelet transform is the one of continuous wavelet transforms which build a time-frequency representation of a signal and provide very good time-frequency localization. The continuous wavelet transform of $x(n)$ is defined as follows [101]:

$$\Psi_n(s) = \sum_{k=0}^{N-1} x(k) \sqrt{\frac{\wp}{s}} \psi_0^* \left[\frac{(k-n)\wp}{s} \right], \tag{4.8}$$

where the $(\cdot)^*$ indicates the complex conjugate, ψ_0^* represents normalized wavelet function, s is the scale parameter, $\wp$ is the time interval and n is the time index. The m^{th} order derivative of complex Gaussian wavelet function is:

$$\psi_0(\eta) = \frac{(-1)^{m+1}}{\sqrt{\Gamma\left(m+\frac{1}{2}\right)}} \frac{d^m}{d\eta^m} \left(e^{-\eta^2} \right), \tag{4.9}$$

where η is a non-dimensional time parameter. Then, power spectrum, $\mho(s)$ of the wavelet can be defined as follows:

$$\mho(s) = |\Psi_n(s)|^2 \ . \tag{4.10}$$

4.3 Data Acquisition

The network consists of a receiver, a transmitter, and a jammer node. CP removal complex signals are gathered without being inserted into any process like equalization, demodulation. In the test-bed design, Universal Software Radio Peripheral-2921s (USRPs) are used as software defined radio (SDR) nodes for the transmitter, the receiver, and the jammer. Two different scenarios are evaluated by changing the position of jammer and transmitter nodes. The distance between receiver and transmitter, d_{rt} is equal to $1.5m$ and $1m$ for scenario 1 (SC-1) and scenario 2 (SC-2), respectively. The distance between receiver and jammer, d_{rj} is set to $1m$ and $1.5m$ for SC-1 and SC-2, respectively. Regarding each scenario, a dataset is created over the designed test-bed. Test-bed parameters can be seen from Table **??** and Figure 4.5. A dataset has 24000 samples and a sample consists of 16896 OFDM symbol.

Raw signals are pre-processed with TFTs. To analyze the experimental data in both time domain and frequency domain, CGD and STFT are preferred. In this way, the

neighborhood relationships between time and frequency components are represented using different time-frequency resolutions. Figure 4.1 shows the illustrations of instances regarding SC-1. Exemplary frequency spectra of the signals of BJ and RSJ with a center frequency of 2.45 GHz are shown in Figures 4.2 and 4.3.

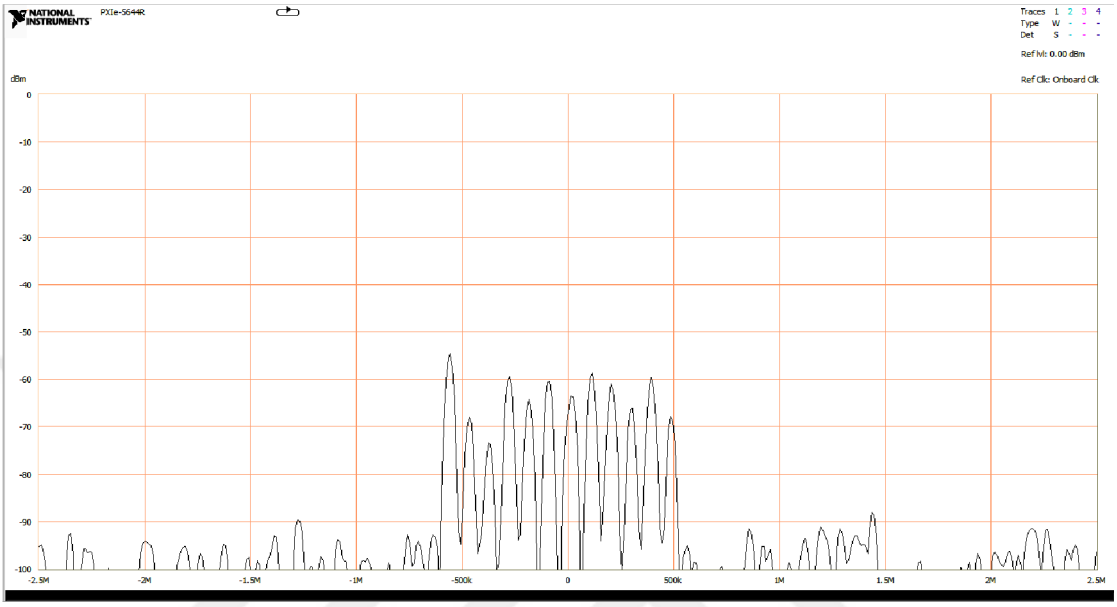

Figure 4.2 : Real-time frequency spectra of RSJ signals

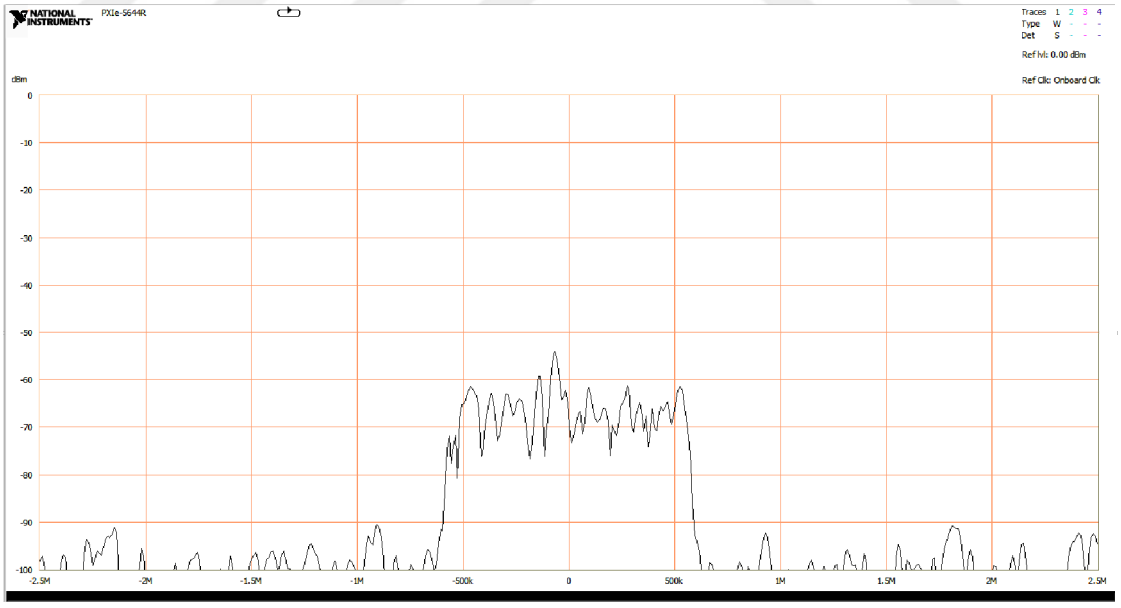

Figure 4.3 : Real-time frequency spectra of barrage jammer signals

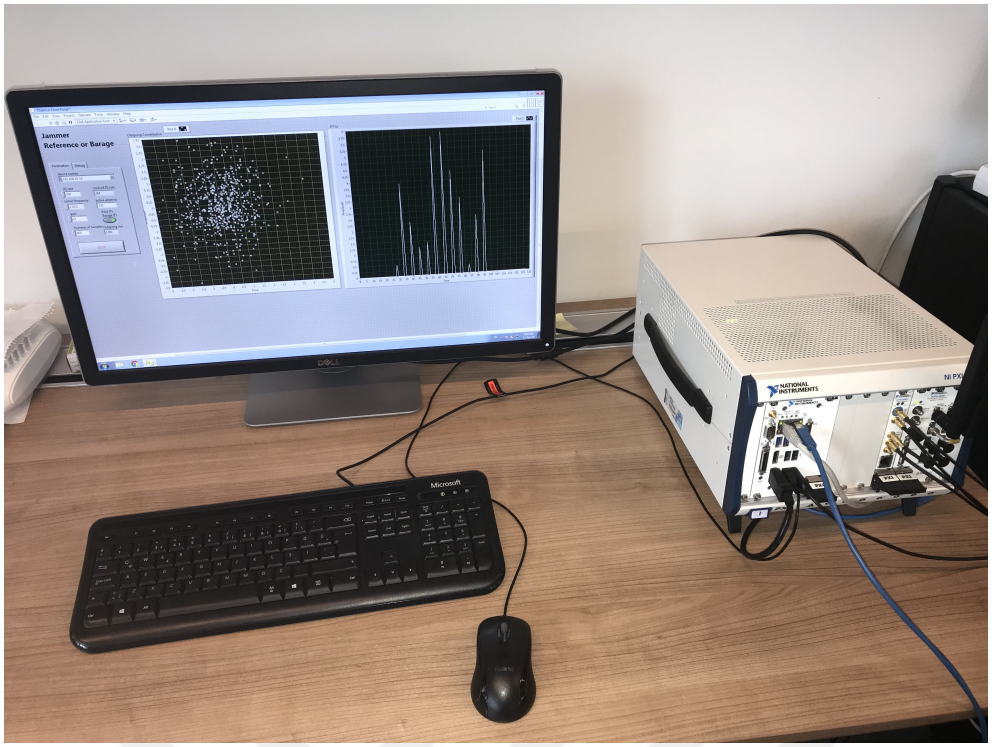

Figure 4.4 : The testbed for data collecting. Host computer.

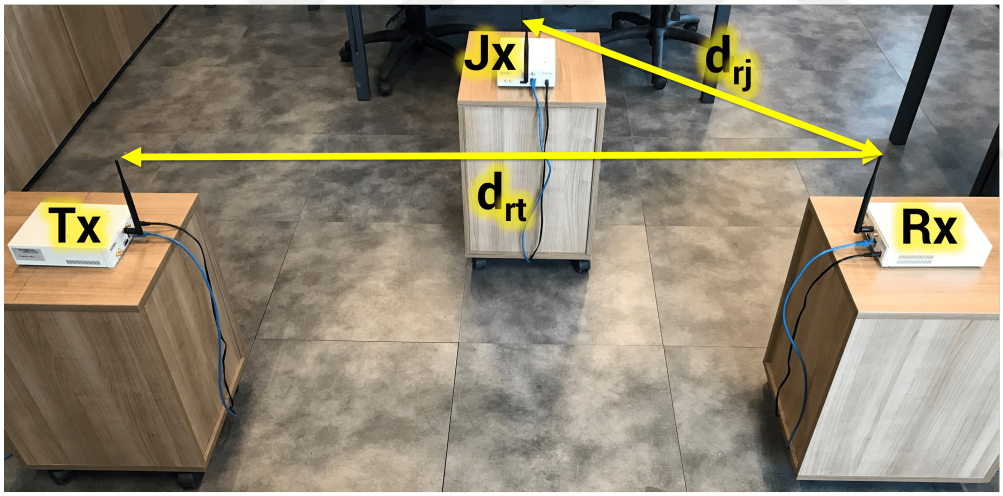

Figure 4.5 : The USRP testbed for data collection.

4.4 The Proposed Classification Architectures

In this study, two neural network architectures are implemented to determine the presence of jammer and identification of the attack type.

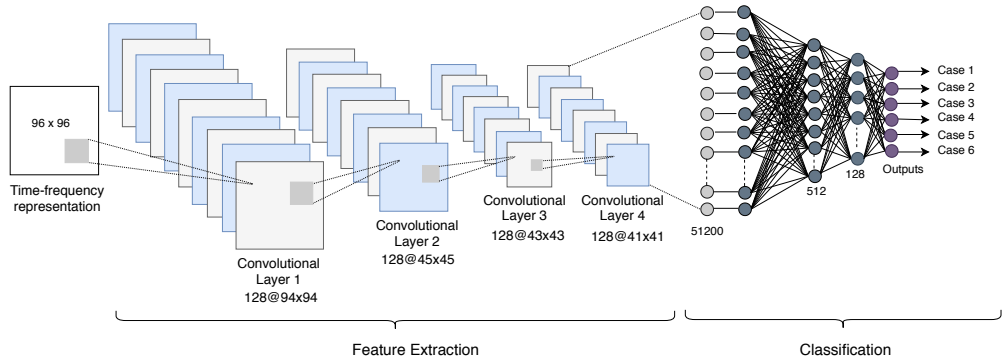

Figure 4.6 : The proposed CNN architecture.

- *Deep Convolutional Neural Network:* The proposed DCNN architecture, Figure 4.6 is composed of two main parts which are feature extraction and classification. The feature extraction part is composed of 4 convolution, 2 pooling and a flatten layers. Throughout the sequence of layers, convolutional layers extract the essence of each data sample from basic features to complex ones with convolutional kernels and produce feature maps. Pooling layers eliminate less critical features on these feature maps. In the proposed architecture, the convolutional kernel size is 3×3, the number of feature maps is 128 for each convolutional layer, and the pooling size is 2×2. In the classification part, 3 fully-connected layers are used and the number of units at each layer are 512, 128, 6, respectively. Except for the output layer, rhe rectified linear unit (ReLu) is used in all layers.

- *Deep Recurrent Neural Networks:* The proposed DRNN based architecture is designed using Long short-term memory (LSTM) layer which has the ability to store an important input as long as it is necessary [102]. The proposed DRNN architecture is designed using LSTM cell which has the ability to store an important input as long as it is needed. In this way, it provides the ability to capture long-term patterns. Our DRNN architecture consists of two LSTM layers and 4 fully-connected layers. Each LSTM layer has 576 units and the number of units at each layer are 1024, 512, 128, 6, respectively. Tangent hyperbolic function, ReLu and Softmax function are used as activation functions at LSTM layers, fully-connected layers, respectively.

As a regularization technique, which provides a prevention against overfitting, dropout layers are implemented to fully-connected layers in both architectures. The batch

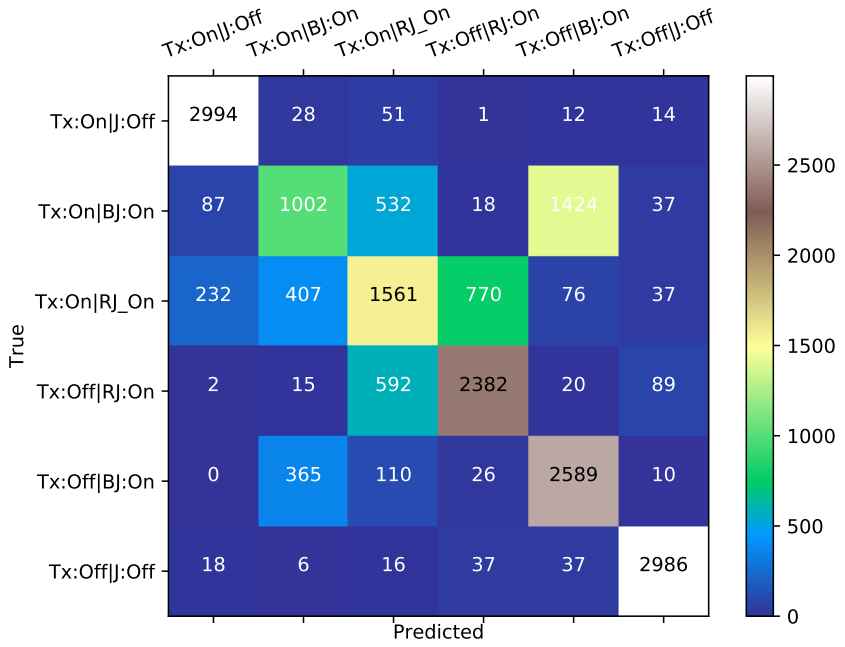

Figure 4.7 : Confusion matrix for SC-1, CGD, DRNN.

normalization layer is included to enable a faster training process. Tensors are generated with a uniform distribution to initialize kernels for all layers. At the output layer of both architectures, Softmax activation function is used due to the confirmation to multi-class classification problems. The cost of the model is calculated using categorical-cross entropy loss function and adaptive moment estimation (ADAM) function is preferred to optimize the architecture.

4.5 Results

In this paper, we target to make a comparison in terms of performance of two state-of-the-art neural network architectures and pre-processing methods regarding jammer detection. We generate 2 datasets for each scenario using SDR nodes. They are processed with CGD wavelet transform and STFT.

5400 samples are used for the training of the proposed DCNN and DRNN and 18600 samples are used to test both schemes. DCNN and DRNN architectures are implemented in KERAS Python library interface that works with Tensorflow backend.

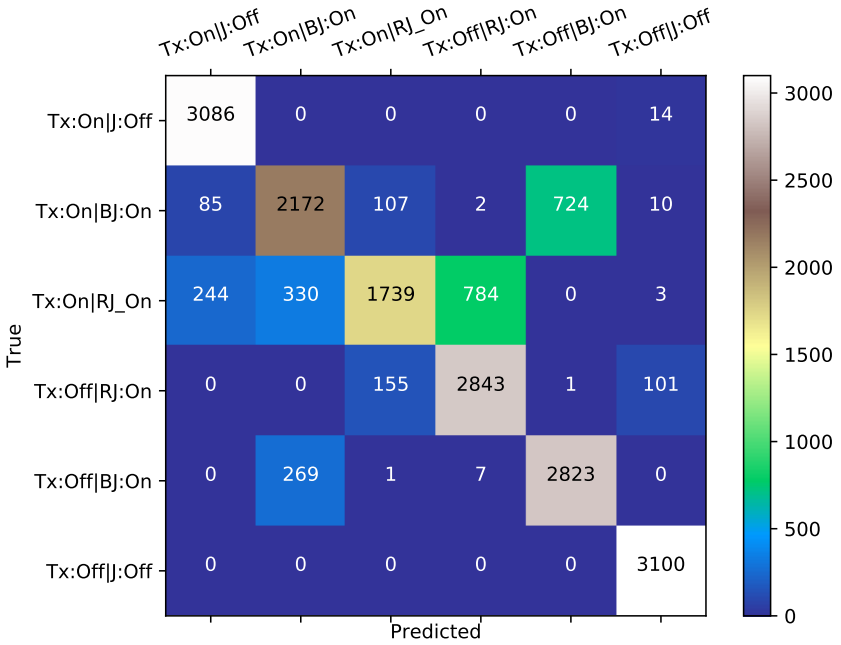

Figure 4.8 : Confusion matrix for SC-1, CGD, DCNN.

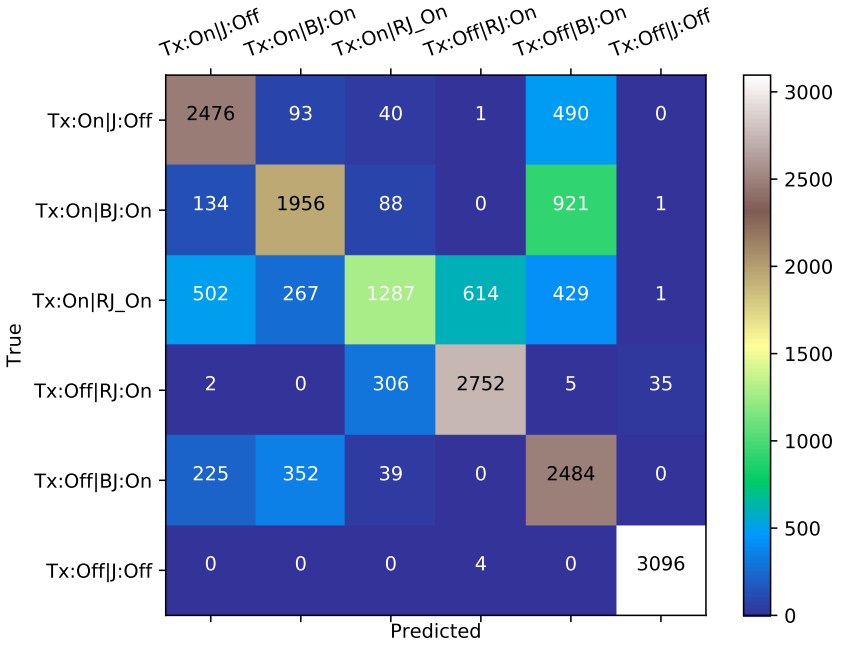

Figure 4.9 : Confusion matrix for SC-1, STFT, DRNN.

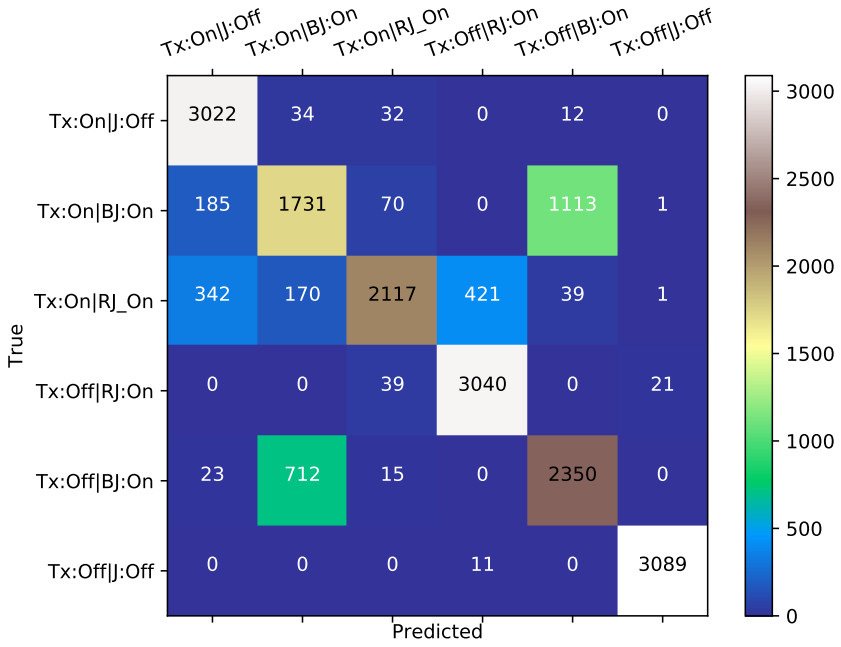

Figure 4.10 : Confusion matrix for SC-1, STFT, DCNN.

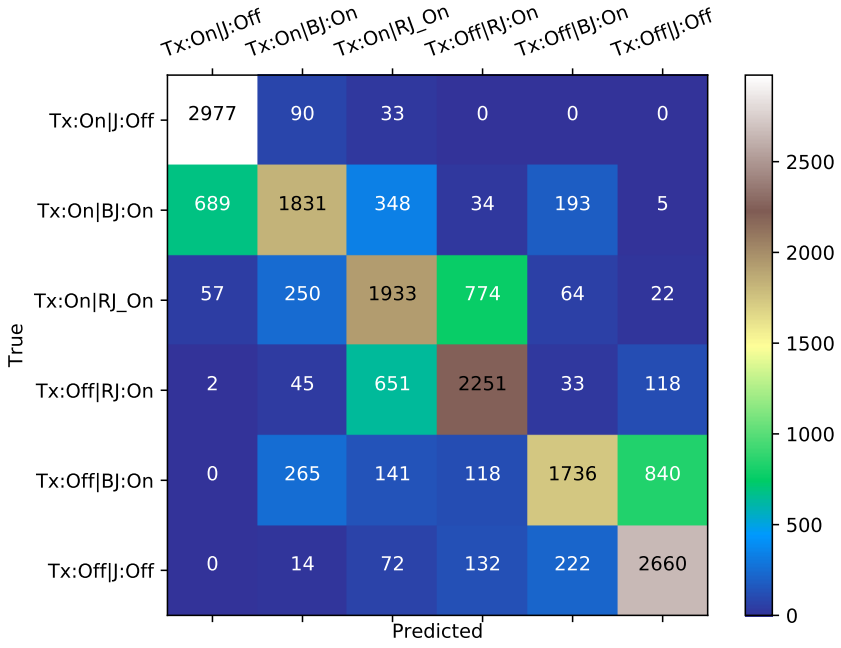

Figure 4.11 : Confusion matrix for SC-2, CGD, DRNN.

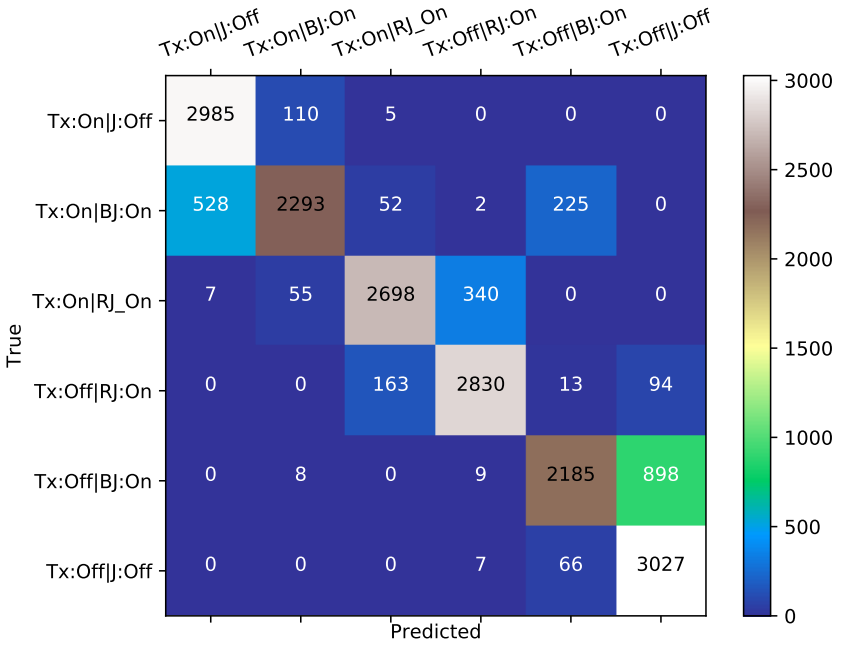

Figure 4.12 : Confusion matrix for SC-2, CGD, DCNN.

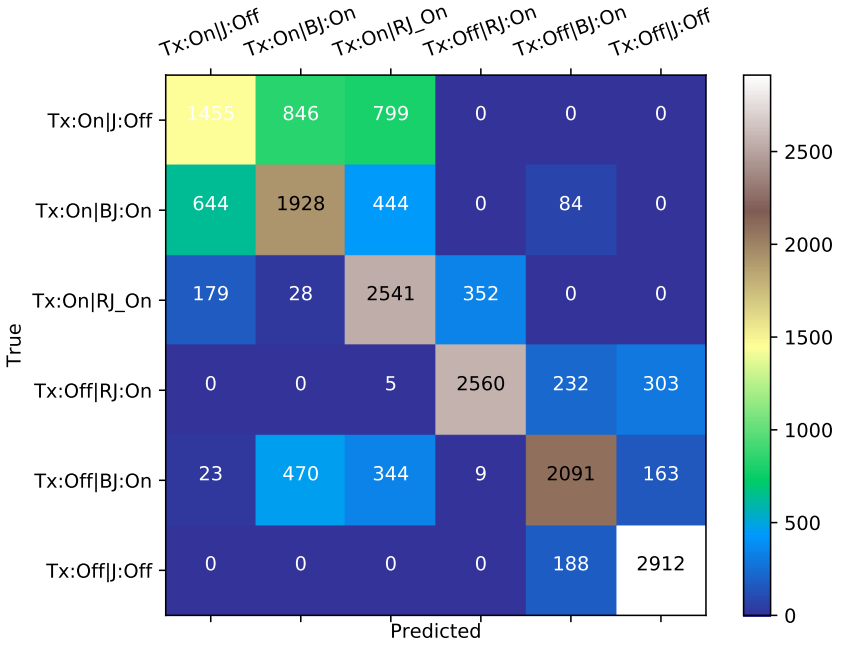

Figure 4.13 : Confusion matrix for SC-2, STFT, DRNN.

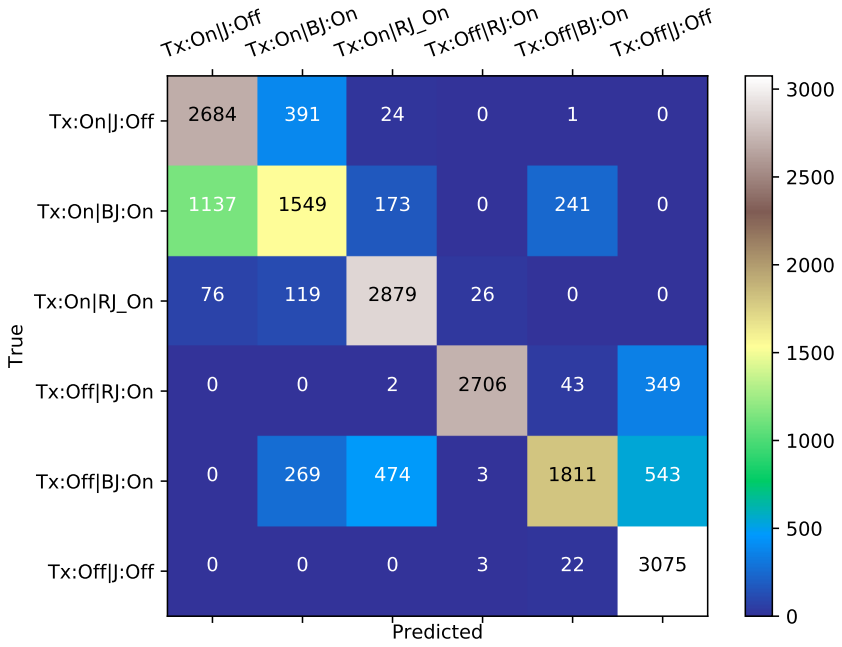

Figure 4.14 : Confusion matrix for SC-2, STFT, DCNN.

The results of DRNN and DCNN related to all scenarios and TFTs can be seen from Figures (4.7-4.14). The results show that both classifiers can predict the existence of transmission with a high accuracy while there is no attack. When the transmitter is closer than the attacker, classifiers performances increase. However, the proposed CNN based detection scheme achieves comparable performance to DRNN in overall.

Figures 4.7, 4.9, 4.11, 4.13 shows that the forecasting accuracy of DRNN is between 72 and 76 percentage. When STFT is preferred as a pre-processing method, DRNN's detection performance is superior. This is expected plausible because, DRNN architecture is more compatible with the time-series sequence and STFT focuses on the data with the same window size in time and frequency planes. In this way, DRNN sees and captures time-dependent data exchanges more clearly. However, we experienced after several trials, this does not mean that the number of units in LSTM layers has a linear relationship with the accuracy. Although DRNN indicates a higher performance with STFT, it is more robust with CGD wavelet to the different locations of the jammer. Figures 4.8, 4.10, 4.12 and 4.14 show that DCNN based attacker detection schemes outperforms DRNN. Especially, when the model trained with the pre-processed dataset with CGD wavelet transformation, DCNN correctly estimates an average of 85% as

Table 4.1 : Parameters.

Steps	**Parameters**	**Value**
Classification	# of scale factors	256
	# of data point for FFT	512
	Epoch # of DCNN	100
	Epoch # of DRNN	200
	Batch size	32
	Optimizer	ADAM
	Learning rate	0.00001
	γ_1	0.9
	γ_2	0.999
	Sample size	96×96
Test-bed	Carrier frequency	2.45 GHz
	Cyclic prefix	Normal
	Channel	Rician
	Modulation type	4 QAM
	Transmitter gains	5, 10, 15 dB
	Jammer gains	5, 10, 15 dB
	IQ rate	2 MS/s

shown in Figure 4.8 and 4.12. As the attacker gets to closer to the receiver in the absence of a transmission, the identification performance of DCNN classifier increases. However, the estimation accuracy does not increase depending on the attacker's position for the existence of the transmission and attack signals. Here, the factor effects the classification merit is the distance between transmitter and receiver.

DCNN's performance in Figure 4.10 and Figure 4.14 indicates that STFT provides fewer contributions to jointly detect and identify in the considered scenarios. It shows a reasonable performance to determine the type of attacker when there is only a transmitter or attacker. However, DCNN becomes more sensitive to the presence of transmitter and jammer.

4.6 Conclusion

A software defined radio node based test network is designed and implemented to detect the presence of jamming nodes, in both the presence and the absence of the legitimate transmission signal. The designed system is OFDM based. Two different jamming

attacks are considered, the classical wide-band barrage jamming and the reference signal jamming that is mainly targeting the pilot signals. The received base band data is then passed through a pre-processing step to improve detection and identification performances. To determine the presence of the jammers, two different neural networks are considered, namely deep convolutional neural networks and deep recurrent neural networks. Test results show that up to 86.1% accuracy can be reached under a realistic test environment. For future work, different channel conditions will be investigated, and the effect of fading on the detection performance will be studied.

5. CONCLUSIONS

In this thesis, a detection and identification scheme is proposed to realize the existence of an attacker and also to identify the type of attack signal. The proposed scheme has two main sections: the pre-processing and the classification. The performance of the presented approach has been tested on simulation and real-time environments. During the design of the system, the attacks of different jammer types and orthogonal frequency division multiplexing (OFDM) based communication systems were generated numerically and experimentally. In both systems, the power and position of the jammer are changed to investigate the success of the proposed scheme under different conditions.

In the first part of the proposed system, received complex signals data are represented on the time-frequency domain. Different transformations: Choi-Williams, Gabor wavelet, short-time Fourier (STFT), complex Gaussian derivative wavelet transformation are applied to received signals. In this way, the characteristics of the signals related to different types of attacks are more clearly demonstrated.

In the second part of the proposed system, deep learning approaches are explored to offer a more flexible and automated structure under changing environmental conditions and deep convolutional neural networks are presented. The proposed CNN architecture is compared with different classification techniques in the literature and the success of the approach is demonstrated.

In this context, firstly, LTE based communication system is formed in the simulation environment and raw signal datasets are obtained. The complex signal data are represented by using different time-frequency transformations. The designed deep CNN architecture is compared with the support vector machine (SVM) classifier in terms of performance. The deep learning based classifier is found more talented to identify the

type of jamming signals. In addition, it is observed that both methods show higher classification accuracies on the data processed by Gabor wavelet transform method.

In the same context, the data set is built by using software defined radios in order to observe the performance of the proposed scheme regarding signals on real-time environments. Also in this study, datasets and scenarios are extended by investigating the presence of a jammer as well as the type of jammer. A complex Gaussian derivative wavelet transform is presented in the pre-processing and its performance is compared to the STFT. The performance of the proposed CNN architecture is compared with another state of the art technique, DRNN. As a result of the study, the proposed CNN-based approach in the real-time data set shows higher performance.

As a result of all studies, a scheme is proposed for the detection and identification of threats to secure communications. The systems that prevent or mitigate these attacks should be developed to ensure the security requirements of communication systems. In future studies, the proposed scheme will be extended by working on coping methods and countermeasures.

REFERENCES

[1] **Panagiotou, P.**, **Anastasopoulos, A. and Polydoros, A.** (2000). Likelihood ratio tests for modulation classification, *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155)*, volume 2, pp.670–674 vol.2.

[2] **Dobre, O.A.** (2015). Signal identification for emerging intelligent radios: classical problems and new challenges, *IEEE Instrumentation Measurement Magazine*, *18*(2), 11–18.

[3] **Balaban, S.** (2019). Deep learning and face recognition: the state of the art, *CoRR*, *abs/1902.03524*, `http://arxiv.org/abs/1902.03524`, `1902.03524`.

[4] **Todorov, P.** (2019). A Game of Dice: Machine Learning and the Question Concerning Art, *arXiv e-prints*, arXiv:1904.01957, `1904.01957`.

[5] **Briot, J.**, **Hadjeres, G. and Pachet, F.** (2017). Deep Learning Techniques for Music Generation - A Survey, *CoRR*, *abs/1709.01620*, `http://arxiv.org/abs/1709.01620`, `1709.01620`.

[6] **LeCun, Y.**, **Bottou, L.**, **Bengio, Y. and Haffner, P.** (1998). Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, *86*(11), 2278–2324.

[7] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.E.** (2017). ImageNet Classification with Deep Convolutional Neural Networks, *Commun. ACM*, *60*(6), 84–90, `http://doi.acm.org/10.1145/3065386`.

[8] **Zeiler, M.D. and Fergus, R.** (2013). Visualizing and Understanding Convolutional Networks, *CoRR*, *abs/1311.2901*, `http://arxiv.org/abs/1311.2901`, `1311.2901`.

[9] **Szegedy, C.**, **Liu, W.**, **Jia, Y.**, **Sermanet, P.**, **Reed, S.E.**, **Anguelov, D.**, **Erhan, D.**, **Vanhoucke, V. and Rabinovich, A.** (2014). Going Deeper with Convolutions, *CoRR*, *abs/1409.4842*, `http://arxiv.org/abs/1409.4842`, `1409.4842`.

[10] **Simonyan, K. and Zisserman, A.** (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv e-prints*, arXiv:1409.1556, `1409.1556`.

[11] **He, K.**, **Zhang, X.**, **Ren, S. and Sun, J.** (2015). Deep Residual Learning for Image Recognition, *CoRR*, *abs/1512.03385*, `http://arxiv.org/abs/1512.03385`, `1512.03385`.

[12] **Paszke, A.**, **Chaurasia, A.**, **Kim, S. and Culurciello, E.** (2016). ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation, *CoRR*, *abs/1606.02147*, `http://arxiv.org/abs/1606.02147`, `1606.02147`.

[13] **Iandola, F.N.**, **Moskewicz, M.W.**, **Ashraf, K.**, **Han, S.**, **Dally, W.J. and Keutzer, K.** (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size, *CoRR*, *abs/1602.07360*, `http://arxiv.org/abs/1602.07360`, `1602.07360`.

[14] **Elwekeil, M.**, **Jiang, S.**, **Wang, T. and Zhang, S.** (2018). Deep Convolutional Neural Networks for Link Adaptations in MIMO-OFDM Wireless Systems, *IEEE Wireless Communications Letters*, 1–1.

[15] **Ha, C.**, **You, Y. and Song, H.** (2019). Machine Learning Model for Adaptive Modulation of Multi-Stream in MIMO-OFDM System, *IEEE Access*, *7*, 5141–5152.

[16] **Luo, C.**, **Ji, J.**, **Wang, Q.**, **Chen, X. and Li, P.** (2018). Channel State Information Prediction for 5G Wireless Communications: A Deep Learning Approach, *IEEE Transactions on Network Science and Engineering*, 1–1.

[17] **Wang, Y.**, **Liu, M.**, **Yang, J. and Gui, G.** (2019). Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios, *IEEE Transactions on Vehicular Technology*, *68*(4), 4074–4077.

[18] **Tang, B.**, **Tu, Y.**, **Zhang, Z. and Lin, Y.** (2018). Digital Signal Modulation Classification With Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks, *IEEE Access*, *6*, 15713–15722.

[19] **Xie, J.**, **Liu, C.**, **Liang, Y. and Fang, J.** (2019). Activity Pattern Aware Spectrum Sensing: A CNN-Based Deep Learning Approach, *IEEE Communications Letters*, 1–1.

[20] **He, Y.**, **Zhang, Z.**, **Yu, F.R.**, **Zhao, N.**, **Yin, H.**, **Leung, V.C.M. and Zhang, Y.** (2017). Deep-Reinforcement-Learning-Based Optimization for Cache-Enabled Opportunistic Interference Alignment Wireless Networks, *IEEE Transactions on Vehicular Technology*, *66*(11), 10433–10445.

[21] **Zhou, P.**, **Fang, X.**, **Wang, X.**, **Long, Y.**, **He, R. and Han, X.** (2019). Deep Learning-Based Beam Management and Interference Coordination in Dense mmWave Networks, *IEEE Transactions on Vehicular Technology*, *68*(1), 592–603.

[22] **Challita, U.**, **Saad, W. and Bettstetter, C.** (2019). Interference Management for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach, *IEEE Transactions on Wireless Communications*, *18*(4), 2125–2140.

[23] **Yao, H.**, **Gao, P.**, **Wang, J.**, **Zhang, P.**, **Jiang, C. and Han, Z.** (2019). Capsule Network Assisted IoT Traffic Classification Mechanism for Smart Cities, *IEEE Internet of Things Journal*, 1–1.

[24] **Kim, M.**, **Lee, W. and Cho, D.** (2018). A Novel PAPR Reduction Scheme for OFDM System Based on Deep Learning, *IEEE Communications Letters*, *22*(3), 510–513.

[25] **Tang, F.**, **Mao, B.**, **Fadlullah, Z.M.**, **Kato, N.**, **Akashi, O.**, **Inoue, T. and Mizutani, K.** (2018). On Removing Routing Protocol from Future Wireless Networks: A Real-time Deep Learning Approach for Intelligent Traffic Control, *IEEE Wireless Communications*, *25*(1), 154–160.

[26] **Fadlullah, Z.M.**, **Mao, B.**, **Tang, F. and Kato, N.** (2018). Value Iteration Architecture based Deep Learning for Intelligent Routing Exploiting Heterogeneous Computing Platforms, *IEEE Transactions on Computers*, 1–1.

[27] **Mao, B.**, **Fadlullah, Z.M.**, **Tang, F.**, **Kato, N.**, **Akashi, O.**, **Inoue, T. and Mizutani, K.** (2017). Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning, *IEEE Transactions on Computers*, *66*(11), 1946–1960.

[28] **Zhang, J.**, **Li, Y. and Yin, J.** (2018). Modulation classification method for frequency modulation signals based on the time – frequency distribution and CNN, *12*(1), 244–249.

[29] **Khan, F.N.**, **Zhong, K.**, **Al-Arashi, W.H.**, **Yu, C.**, **Lu, C. and Lau, A.P.T.** (2016). Modulation Format Identification in Coherent Receivers Using Deep Machine Learning, *IEEE Photonics Technology Letters*, *28*(17), 1886–1889.

[30] **O'Shea, T.J.**, **Corgan, J. and Clancy, T.C.** (2016). Convolutional radio modulation recognition networks, *Communications in Computer and Information Science*, *629*, 213–226.

[31] **O'Shea, T.J.**, **Roy, T. and Clancy, T.C.** (2017). Over the Air Deep Learning Based Radio Signal Classification, *4553*(c), 1–13.

[32] **Zhang, M.**, **Diao, M. and Guo, L.** (2017). Convolutional neural networks for automatic cognitive radio waveform recognition, *IEEE Access*, *5*, 11074–11082.

[33] **Tang, B.**, **Tu, Y.**, **Zhang, Z. and Lin, Y.** (2018). Digital Signal Modulation Classification With Data Augmentation Using Generative Adversarial Nets in Cognitive Radio Networks, *IEEE Access*, *6*, 15713–15722.

[34] **Ha, C. and Song, H.** (2018). Signal Detection Scheme Based on Adaptive Ensemble Deep Learning Model, *IEEE Access*, *6*, 21342–21349.

[35] **Ye, H.**, **Li, G.Y. and Juang, B.** (2018). Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems, *IEEE Wireless Communications Letters*, *7*(1), 114–117.

[36] **O'Shea, T.J.**, **Clancy, T.C. and McGwier, R.W.** (2016). Recurrent Neural Radio Anomaly Detection, *arXiv e-prints*, arXiv:1611.00301, 1611.00301.

[37] **Rajendran, S.**, **Meert, W.**, **Lenders, V. and Pollin, S.** (2018). SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features, *arXiv e-prints*, arXiv:1807.08316, 1807.08316.

[38] **Fragkiadakis, A.G.**, **Siris, V.A. and Petroulakis, N.** (2010). Anomaly-Based Intrusion Detection Algorithms for Wireless Networks, ***E. Osipov**, **A. Kassler**, **T.M. Bohnert and X. Masip-Bruin**, editors, Wired/Wireless Internet Communications*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.192–203.

[39] **Nakanishi, K.**, **Mori, H.**, **Kumagaya, T. and Aoki, T.** (2018). Signal Detection Method for Interference From Inverter Microwave Ovens in WLAN Systems, *IEEE Transactions on Cognitive Communications and Networking*, *4*(2), 368–378.

[40] **Politis, C.**, **Maleki, S.**, **Duncan, J.M.**, **Krivochiza, J.**, **Chatzinotas, S. and Ottesten, B.** (2018). SDR Implementation of a Testbed for Real-Time Interference Detection With Signal Cancellation, *IEEE Access*, *6*, 20807–20821.

[41] **Akeret, J.**, **Chang, C.**, **Lucchi, A. and Refregier, A.** (2017). Radio frequency interference mitigation using deep convolutional neural networks, *Astronomy and Computing*, *18*, 35 – 39, http://www.sciencedirect.com/science/article/pii/S2213133716301056.

[42] **Erpek, T.**, **Sagduyu, Y.E. and Shi, Y.** (2018). Deep Learning for Launching and Mitigating Wireless Jamming Attacks, *IEEE Transactions on Cognitive Communications and Networking*, 1–1.

[43] **Yuan, Z.**, **Niyato, D.**, **Li, H.**, **Song, J.B. and Han, Z.** (2012). Defeating Primary User Emulation Attacks Using Belief Propagation in Cognitive Radio Networks, *IEEE Journal on Selected Areas in Communications*, *30*(10), 1850–1860.

[44] **Khalaf, M.**, **Youssef, A. and El-Saadany, E.** (2018). Joint Detection and Mitigation of False Data Injection Attacks in AGC Systems, *IEEE Transactions on Smart Grid*, 1–1.

[45] **Gao, N.**, **Qin, Z. and Jing, X.** (2019). Pilot Contamination Attack Detection and Defense Strategy in Wireless Communications, *IEEE Signal Processing Letters*, 1–1.

[46] **El-Keyi, A.**, **Üreten, O.**, **Yanikomeroglu, H. and Yensen, T.** (2017). LTE for Public Safety Networks: Synchronization in the presence of jamming, *IEEE Access*, *5*, 20800–20813.

[47] **Shahriar, C.**, **McGwier, R. and Clancy, T.C.** (2013). Performance impact of pilot tone randomization to mitigate OFDM jamming attacks, *IEEE Consumer Communications and Networking Conference (CCNC)*, pp.813–816.

[48] **Akhlaghpasand, H.**, **Razavizadeh, S.M.**, **Björnson, E. and Do, T.T.** (2018). Jamming Detection in Massive MIMO Systems, *IEEE Wireless Communications Letters*, *7*(2), 242–245.

[49] **Navda, V.**, **Bohra, A.**, **Ganguly, S. and Rubenstein, D.** (2007). Using channel hopping to increase 802.11 resilience to jamming attacks, *Proceedings - IEEE INFOCOM*, 2526–2530.

[50] **Xiao, L.**, **Li, Y.**, **Dai, C.**, **Dai, H. and Poor, H.V.** (2018). Reinforcement Learning-Based NOMA Power Allocation in the Presence of Smart Jamming, *IEEE Transactions on Vehicular Technology*, *67*(4), 3377–3389.

[51] **Hoang, D.T.**, **Niyato, D.**, **Wang, P. and Kim, D.I.** (2015). Performance Analysis of Wireless Energy Harvesting Cognitive Radio Networks Under Smart Jamming Attacks, *IEEE Transactions on Cognitive Communications and Networking*, *1*(2), 200–216.

[52] **Xiao, L.**, **Lu, X.**, **Xu, D.**, **Tang, Y.**, **Wang, L. and Zhuang, W.** (2018). UAV relay in VANETs against smart jamming with reinforcement learning, *IEEE Transactions on Vehicular Technology*, *67*(5), 4087–4097.

[53] **Shahriar, C.**, **La Pan, M.**, **Lichtman, M.**, **Clancy, T.C.**, **McGwier, R.**, **Tandon, R.**, **Sodagari, S. and Reed, J.H.** (2015). PHY-Layer Resiliency in OFDM Communications: A Tutorial, *IEEE Communications Surveys Tutorials*, *17*(1), 292–314.

[54] **Aziz, F.M.**, **Shamma, J.S. and Stuber, G.L.** (2017). Jammer-Type Estimation in LTE with a Smart Jammer Repeated Game, *IEEE Transactions on Vehicular Technology*, *66*(8), 7422–7431.

[55] **Karagiannis, D. and Argyriou, A.** (2018). Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning, *Vehicular Communications*, *13*, 56–63.

[56] **Wang, L. and Wyglinski, A.M.** (2011). A combined approach for distinguishing different types of jamming attacks against wireless networks, *Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp.809–814.

[57] **Puñal, O.**, **Akta, I.**, **Schnelke, C.J.**, **Abidin, G.**, **Wehrle, K. and Gross, J.** Machine Learning-based Jamming Detection for IEEE 802.11: Design and Experimental Evaluation.

[58] **Zhang, L.**, **Ren, J. and Li, T.** (2012). Time-varying jamming modeling and classification, *IEEE Transactions on Signal Processing*, *60*(7), 3902–3907.

[59] **Lichtman, M.**, **Jover, R.P.**, **Labib, M.**, **Rao, R.**, **Marojevic, V. and Reed, J.H.** (2016). LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation, *IEEE Communications Magazine*, *54*(4), 54–61.

[60] **Cheng, M.**, **Ling, Y. and Wu, W.B.** (2018). Time Series Analysis for Jamming Attack Detection in Wireless Networks, *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, *2018-January*, 1–7.

[61] **Kurt, M.N.**, **Yilmaz, Y. and Wang, X.** (2018). Real-Time Detection of Hybrid and Stealthy Cyber-Attacks in Smart Grid, *IEEE Transactions on Information Forensics and Security*, *14*(2), 498–513.

[62] **Jahanshahi, J.A. and Eslami, M.** (2011). On the performance of SVM based jamming attacks detection algorithm in base station, *2011 IEEE Swedish Communication Technologies Workshop (Swe-CTW)*, pp.109–113.

[63] **Tugnait, J.K.** (2018). Pilot spoofing attack detection and countermeasure, *IEEE Transactions on Communications*, *66*(5), 2093–2106.

[64] **Goodfellow, I.**, **Bengio, Y. and Courville, A.** (2016). *Deep Learning*, MIT Press, `http://www.deeplearningbook.org`.

[65] **Raschka, S.** (2015). *Python Machine Learning*, Packt Publishing.

[66] **Khan, S.**, **Rahmani, H.**, **Shah, S.A.A.**, **Bennamoun, M.**, **Medioni, G. and Dickinson, S.** (2018). *A Guide to Convolutional Neural Networks for Computer Vision*, Morgan Claypool, `https://ieeexplore.ieee.org/document/8295029`.

[67] **Ioffe, S. and Szegedy, C.** (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *CoRR*, *abs/1502.03167*, `http://arxiv.org/abs/1502.03167`, `1502.03167`.

[68] **Srivastava, N.**, **Hinton, G.**, **Krizhevsky, A.**, **Sutskever, I. and Salakhutdinov, R.** (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *J. Mach. Learn. Res.*, *15*(1), 1929–1958, `http://dl.acm.org/citation.cfm?id=2627435.2670313`.

[69] **Shahriar, C.**, **La Pan, M.**, **Lichtman, M.**, **Clancy, T.C.**, **McGwier, R.**, **Tandon, R.**, **Sodagari, S. and Reed, J.H.** (2015). PHY-layer resiliency in OFDM communications: A tutorial, *IEEE Communications Surveys & Tutorials*, *17*(1), 292–314.

[70] **Baker, M.** (2012). From LTE-advanced to the future, *IEEE Communications Magazine*, *50*(2).

[71] **Zou, Y.**, **Zhu, J.**, **Wang, X. and Hanzo, L.** (2016). A survey on wireless security: Technical challenges, recent advances, and future trends, *Proceedings of the IEEE*, *104*(9), 1727–1765.

[72] **Chen, H.**, **Zhang, Y.**, **Li, W.**, **Tao, X. and Zhang, P.** (2017). ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information, *IEEE Access*, *5*, 18066–18074.

[73] **Kong, S.H.**, **Kim, M.**, **Hoang, L.M. and Kim, E.** (2017). Automatic LPI radar waveform recognition using CNN, *IEEE Access*, *6*, 4207–4219.

[74] **Boashash, B.** (2015). *Time-frequency signal analysis and processing: a comprehensive reference*, Academic Press.

[75] **Bhunia, S.**, **Miles, E.**, **Sengupta, S. and Vázquez-Abad, F.** (2018). CR-Honeynet: A Cognitive Radio Learning and Decoy-Based Sustenance Mechanism to Avoid Intelligent Jammer, *IEEE Transactions on Cognitive Communications and Networking*, *4*(3), 567–581.

[76] **Zuba, M.**, **Shi, Z.**, **Peng, Z. and Cui, J.H.** (2011). Launching denial-of-service jamming attacks in underwater sensor networks, *Proceedings of the Sixth ACM International Workshop on Underwater Networks*, ACM, p. 12.

[77] **Shridhara, K.A.**, (2002), Jamming detection and blanking for GPS receivers, US Patent 6,448,925.

[78] **Mallat, S.G.** (1989). A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(7), 674–693.

[79] **Strang, G.** (1993). Wavelet transforms versus Fourier transforms, *Bulletin of the American Mathematical Society*, *28*(2), 288–305.

[80] **Lichtman, M.**, **Reed, J.H.**, **Clancy, T.C. and Norton, M.** (2013). Vulnerability of LTE to hostile interference, *arXiv preprint arXiv:1312.3681*.

[81] **ETSI, T.**, (2009), 136 211 V8. 7.0," LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 8.7. 0 Release 8),".

[82] **Hendrickson, R.C.**, (1999), Jamming detection in a wireless security system, US Patent 5,950,110.

[83] **Lichtman, M.**, **Czauski, T.**, **Ha, S.**, **David, P. and Reed, J.H.** (2014). Detection and mitigation of uplink control channel jamming in LTE, *Military Communications Conference (MILCOM)*, IEEE, pp.1187–1194.

[84] **Di Pietro, R. and Oligeri, G.** (2013). Jamming mitigation in cognitive radio networks, *IEEE Network*, *27*(3), 10–15.

[85] **Mietzner, J.**, **Nickel, P.**, **Meusling, A.**, **Loos, P. and Bauch, G.** (2012). Responsive communications jamming against radio-controlled improvised explosive devices, *IEEE Communications Magazine*, *50*(10), 38–46.

[86] **Cohen, L.** (1989). Time-frequency distributions-a review, *Proceedings of the IEEE*, *77*(7), 941–981.

[87] **Figueiredo, A.**, **Nave, M. and contributors, E.J.** (2004). Time–frequency analysis of nonstationary fusion plasma signals: a comparison between the Choi–Williams distribution and wavelets, *Review of scientific instruments*, *75*(10), 4268–4270.

[88] **Choi, H.I. and Williams, W.J.** (1989). Improved time-frequency representation of multicomponent signals using exponential kernels, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(6), 862–871.

[89] **Vodpl, I.D.Q.G.**, **Divkdu, P.M.**, **Vex, P.**, **Lu, D.F.**, **Hvodpl, P.R.K.**, **Df, V.E.X.**, **Sursrvh, E.H.**, **Dwwdfnv, D.M. and Dojrulwkp, G.** (2011). On the performance of SVM based Jamming Attacks Detection Algorithm in Base Station, *On the performance of SVM based Jamming Attacks Detection Algorithm in Base Station*, 109–113.

[90] **Wu, Z.**, **Zhao, Y.**, **Yin, Z. and Luo, H.** (2017). Jamming signals classification using convolutional neural network, *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, pp.062–067.

[91] **Mahmoud, H.A. and Arslan, H.** (2009). Error vector magnitude to SNR conversion for nondata-aided receivers, *IEEE Transactions on Wireless Communications*, *8*(5).

[92] **Géron, A.** (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, O'Reilly Media.

[93] **Pal, M.** (2008). Multiclass Approaches for Support Vector Machine Based Land Cover Classification, *arXiv prePrint*, *arXiv:0802.2411*.

[94] **Jolliffe, I.**, (2011). Principal component analysis, International Encyclopedia of Statistical Science, Springer, pp.1094–1096.

[95] **Bengio, Y.**, **{LeCun}, Y. and Lecun, Y.** (2007). Scaling Learning Algorithms towards AI, *Large Scale Kernel Machines*, (1), 321–360.

[96] **Gardner, W.A.** (1994). Cyclostationarity in communications and signal processing, **Technical Report**, Statistical Signal Processing Inc Yountville Ca.

[97] **Spooner, C.M.** (2001). On the utility of sixth-order cyclic cumulants for RF signal classification, *Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256)*, volume 1, pp.890–897 vol.1.

[98] **Xu, W.**, **Trappe, W.**, **Zhang, Y. and Wood, T.** (2005). The feasibility of launching and detecting jamming attacks in wireless networks, *6th ACM international symposium on Mobile ad hoc networking and computing*, 46.

[99] **Tsang, C.S.** (2011). Jamming detection and SNR/SNJR estimation, *2011 Aerospace Conference*, pp.1–7.

[100] **Lin, Y. and Li, M.** (2009). Distributed detection of jamming and defense in wireless sensor networks, *2009 43rd Annual Conference on Information Sciences and Systems*, pp.829–834.

[101] **Zhang, X.**, **Luo, P. and Hu, X.** (2017). A hybrid method for classification and identification of emitter signals, *2017 4th International Conference on Systems and Informatics (ICSAI)*, (Icsai), 1060–1065.

[102] **Nachmani, E.**, **Marciano, E.**, **Lugosch, L.**, **Gross, W.J.**, **Burshtein, D. and Be'ery, Y.** (2018). Deep Learning Methods for Improved Decoding of Linear Codes, *IEEE Journal of Selected Topics in Signal Processing*, *12*(1), 119–131.

CURRICULUM VITAE

Name Surname: Selen Geçgel

Place and Date of Birth: Istanbul, Turkey | 13 November 1993

E-Mail: selengecgel@gmail.com

EDUCATION:

- **B.Sc.:** 2016, Yildiz Technical University, Electric-Electronics Faculty, ELECTRONICS and COMMUNICATION ENGINEERING

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- "Jammer Detection based on Artificial Neural Networks: A Measurement Study ", ***Selen Geçgel****, C. Goztepe, G. K. Kurt,* ACM Conference on Security and Privacy in Wireless Mobile Networks (***ACM WiSec***).

- "Identification of Smart Jammers: A Wavelet Based Approach", *O.A. Topal,* ***Selen Geçgel****, E.M. Eksioglu, G. Karabulut Kurt,* ***IEEE TCNN***.

- "PHY Based Jamming Attacks and Countermeasure: An Observation via Real-Time Test System", IEEE International Workshop on Signal Processing Advances in Wireless Communications (***SPAWC***).

- "Identification of Non-Conforming Cordless Phone Signals in Licensed Bands", ***Selen Geçgel****, M.A. Durmaz, H. Alakoca, G. Karabulut Kurt, and C. Ayyıldız*, International Telecommunications Conference ***(ITELCON)***.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- "Transmit Antenna Selection for Massive MIMO-GSM with Machine Learning", ***Selen Geçgel****, C. Goztepe, G. K. Kurt,* IEEE Wireless Communication Letters (***IEEE WCL***).

- "Antenna Selection on Spatial Modulation: A Machine Learning Approach", ***Selen Geçgel****, C. Goztepe, G. K. Kurt,* IEEE Signal Processing and Communications Applications Conference (***IEEE SIU***).