Towards Intelligent Security Mechanisms for Connected Things

by

Paulo FREITAS DE ARAUJO FILHO

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE AND UNIVERSIDADE FEDERAL DE PERNAMBUCO (CO-TUTORSHIP)
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 30, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Prof. Georges Kaddoum, Thesis Supervisor
Department of Electrical Engineering, École de technologie supérieure

Prof. Divanilson R. Campelo, Thesis Co-Supervisor
Centro de Informática, Universidade Federal de Pernambuco

Prof. Chamseddine Talhi, President of the Board of Examiners
Department of Software Engineering and IT, École de technologie supérieure

Prof. Kim Khoa Nguyen, Member of the Jury
Department of Electrical Engineering, École de technologie supérieure

Prof. George Darmiton da Cunha Cavalcanti, External Examiner
Centro de Informática, Universidade Federal de Pernambuco

Prof. Susana Sargento, External Examiner
Departamento de Eletrónica, Telecomunicações e Informática, Universidade de Aveiro

Prof. Michele Nogueira Lima, External Examiner
Departamento de Ciência da Computação, Universidade Federal de Minas Gerais

Prof. Eduardo Coelho Cerqueira, External Independent Examiner
Faculdade da Engenharia de Computação e Telecomunicação, Universidade Federal do Pará

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON MARCH 15, 2023

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

The work contained in this thesis consists of the research outcomes accomplished during my doctoral degree under the supervision of Prof. Georges Kaddoum and Prof. Divanilson Campelo. This work was financially supported by the Fonds de recherche du Québec (FRQNT) B2X Scholarship, Mitacs Accelerate Fellowship, Microsoft Research Ph.D. Fellowship, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

The main topic of this thesis is the security of connected things against cyber-attacks and adversarial attacks. During my Ph.D. studies, I composed two published journal papers and two journal papers currently under review. Additionally, I was a co-author on three journal papers and two conference papers.

The first two chapters of the thesis include the introduction and the required background and literature review on intrusion detection systems, neural networks, and adversarial attacks. The following four chapters are based on the research journal papers I authored during my doctorate. Finally, Chapter six presents the conclusions of the thesis and future research directions.

ACKNOWLEDGEMENTS

I am deeply grateful and dedicate a special thanks to my beloved wife, Aline, who embarked with me on our journey to Canada and who is always by my side, providing me with support, companionship, and love.

I am forever and profoundly grateful to my parents, Ana Rosa and Paulo Araujo, and to my sister, Juliana Gusmão, who instilled in me a deep appreciation for my education and the courage to travel away, who supported me no matter what, and to whom I own everything. I am deeply grateful for my nieces and goddaughters, Letícia and Manuela, who bring me so much happiness and joy. I am also deeply grateful for all my uncles, aunts, cousins, relatives, and friends, who are always there for me no matter what.

I want to acknowledge my gratitude and appreciation to Prof. Georges Kaddoum and Prof. Divanilson Campelo for their mentorship and supervision, which made me grow as a person, professional, and researcher. They have been a constant source of encouragement, guidance, and inspiration throughout the duration of my Ph.D.. I sincerely hope we can continue our collaboration and relationship in the future.

I want to thank my collaborators, colleagues, and friends, to whom I am deeply grateful for their support. I am thankful to Prof. Cleber Zanchettin, who introduced me to the field of machine learning, and to João Victor Evangelista, who introduced me to Prof. Kaddoum and had an essential role in my journey to Canada. I dedicate a special thanks to Henrique Arcoverde and Tempest Security Intelligence, with whom I learned so much and who believed in me and gave me all the support and means to grow as a researcher and professional.

I want to acknowledge and thank the research structure and environment of ÉTS and CIn, and the financial support of Fonds de recherche du Québec, Mitacs, Ericsson, Microsoft, and CAPES.

To all of you, thank you so much.

Paulo

Vers des Mécanismes de Sécurité Intelligents pour les Objets Connectés

Paulo FREITAS DE ARAUJO FILHO

RÉSUMÉ

La nature de diffusion des communications sans fil et l'adoption généralisée des objets connectés augmentent les surfaces d'attaque et permettent aux attaquants de lancer plusieurs cyber-attaques. En outre, l'adoption croissante de l'apprentissage automatique (en anglais machine learning ou ML) dans de nombreuses applications, y compris les communications sans fil, introduit de nouveaux risques et vulnérabilités. Les attaques adverses conçoivent et introduisent de petites perturbations qui trompent les modèles de ML et les amènent à prendre de mauvaises décisions. Elles peuvent donc compromettre les tâches de communication sans fil basées sur l'apprentissage automatique et mettre en péril la disponibilité des communications et la sécurité des objets connectés. Par conséquent, les cyber-attaques et les attaques adverses peuvent compromettre les objectifs de sécurité, causer de graves dommages et pertes financières et même mettre la vie des gens en danger.

Dans cette thèse, nous faisons avancer l'état de l'art dans le domaine de la sécurité en considérant à la fois les problèmes de cyber-attaques et d'attaques adverses. Nous améliorons la sécurité des objets connectés en détectant de manière efficace et efficiente les cyber-attaques tout en défendant les systèmes qui reposent sur l'apprentissage automatique contre les attaques adverses.

Dans le chapitre 2, nous vérifions que, tandis que les systèmes de détection d'intrusion (en anglais intrusion detection systems ou IDS) basés sur l'e ML supervisé ne peuvent pas détecter les attaques inconnues et nécessitent des données d'entraînement étiquetées, ce qui est long, difficile et parfois impossible à obtenir, les approches non supervisées présentent généralement des taux élevés de faux positifs qui provoquent des interruptions de service et font dérailler les centres d'opérations de sécurité (en anglais security operation centers ou SOCs). De plus, nous vérifions que la plupart des IDS non supervisés ont du mal à gérer le temps nécessaire à la modélisation de systèmes très complexes et hétérogènes, de sorte qu'ils ne peuvent pas détecter les cyber-attaques assez rapidement pour les arrêter avant que des dommages ne soient causés. Nous proposons donc un nouvel IDS non supervisé qui détecte les attaques connues et inconnues à l'aide de réseaux adverses génératifs (en anglais generative adversarial networks ou GANs). Notre approche combine la sortie du discriminateur GAN avec une perte de reconstruction qui évalue si les échantillons de données sont conformes aux échantillons d'entraînement. Elle entraîne un réseau neuronal encodeur qui accélère le calcul de la perte de reconstruction, ce qui réduit considérablement les temps de détection par rapport au état de l'art.

Étant donné que de nombreuses attaques comportent plusieurs étapes et sont lancées à partir de différentes applications et de différents dispositifs, le chapitre 3 traite des différentes stratégies qui permettent de prendre en compte les dépendances temporelles des données dans la détection des cyber-attaques. Nous vérifions que si la plupart des IDS existants reposent sur des réseaux à mémoire à long et à court terme (en anglais long short-term memory ou LSTM), des études récentes montrent qu'ils présentent plusieurs inconvénients qui augmentent les temps de détection,

comme une capacité limitée à paralléliser les calculs. Ainsi, nous proposons un nouvel IDS non supervisé basé sur un réseau GAN qui utilise des réseaux convolutifs temporels (en anglais temporal convolutional networks ou TCNs) et l'auto-attention pour remplacer les réseaux LSTM afin de prendre en compte les dépendances temporelles des données. L'approche proposée remplace avec succès les réseaux LSTM pour la détection des attaques et obtient de meilleurs résultats de détection. En outre, elle permet différentes configurations des couches TCN et d'auto-attention pour obtenir différents compromis entre les taux et les temps de détection et satisfaire différentes exigences.

Contrairement aux chapitres 2 et 3, le chapitre 4 concerne les attaques adverses qui compromettent les classificateurs de modulation dans les récepteurs sans fil, mettant ainsi en péril la disponibilité des communications sans fil. Nous vérifions que les techniques d'attaque adverses existantes nécessitent une connaissance complète du modèle du classificateur, ce qui est une hypothèse irréaliste, ou prennent trop de temps pour créer des perturbations adverses, de sorte qu'elles ne peuvent pas altérer les signaux modulés reçus. Nous proposons donc une nouvelle technique d'attaque adverse de type boîte noire qui réduit la précision des classificateurs de modulation plus que les autres attaques adverses du même type et qui crée des perturbations adverses beaucoup plus rapidement qu'eux. La technique proposée est essentielle pour évaluer les risques liés à l'utilisation de classificateurs de modulation basés sur le ML dans les communications sans fil.

Enfin, étant donné les risques et les dommages que peuvent causer les attaques adverses, le chapitre 5 se concentre sur l'étude des techniques de défense contre ces menaces sophistiquées. Nous vérifions que seules quelques techniques de défense existent pour protéger les classificateurs de modulation contre ces attaques, la plupart d'entre elles ne réduisant que marginalement leur impact sur la précision du classificateur. Par conséquent, nous proposons une technique de défense pour protéger les classificateurs de modulation des attaques adverses afin que ces attaques ne nuisent pas à la disponibilité des communications sans fil. L'approche que nous proposons détecte et supprime les perturbations adverses tout en réduisant la sensibilité des classificateurs basés sur le ML. Par conséquent, elle diminue avec succès la réduction de la précision causée par différentes techniques d'attaques adverses.

Mots-clés: Internet des objets, 5G, 6G, Sécurité, Sécurité des réseaux, Systèmes de détection d'intrusion, Apprentissage automatique, Apprentissage profonde, Réseaux adversariaux génératifs, Classification de la modulation, Attaques adverses.

Towards Intelligent Security Mechanisms for Connected Things

Paulo FREITAS DE ARAUJO FILHO

ABSTRACT

The broadcast nature of wireless communications and the widespread adoption of connected things increase attack surfaces and enable attackers to launch several cyber-attacks. Moreover, the increasing adoption of machine learning (ML) in many applications, including wireless communications, introduces new risks and vulnerabilities. Adversarial attacks craft and introduce small perturbations that fool ML models into making wrong decisions. Hence, they may compromise wireless communications tasks based on ML and jeopardize communication availability and connected objects' security. Therefore, cyber-attacks and adversarial attacks may compromise security goals, causing severe damage and financial losses and even putting people's lives at risk.

In this thesis, we advance the state-of-the-art in the security field by considering both the cyber-attacks and adversarial attacks problems. We enhance the security of connected objects by effectively and efficiently detecting cyber-attacks while defending systems that rely on machine learning from adversarial attacks.

In Chapter 2, we verify that while supervised ML-based intrusion detection system (IDS) cannot detect unknown attacks and require labeled training data, which is time-consuming, challenging, and sometimes impossible to obtain, unsupervised approaches usually present high false positive rates that cause service disruptions and derail security operation centers (SOCs). Moreover, we verify that most unsupervised IDSs struggle with the time required to model highly complex and heterogeneous systems so that they cannot detect cyber-attacks quickly enough to stop them before damage is caused. Thus, we propose a novel unsupervised IDS that detects known and unknown attacks using generative adversarial networks (GANs). Our approach combines the GAN discriminator's output with a reconstruction loss that evaluates whether data samples comply with the training samples. It trains an encoder neural network that accelerates the reconstruction loss computation, significantly reducing detection times compared to state-of-the-art approaches.

Since many attacks have multiple steps and are launched from different applications and devices, Chapter 3 concerns different strategies for considering time dependencies among data in the detection of cyber-attacks. We verify that while most of the existing IDSs rely on long short-term memory (LSTM) networks, recent studies show that they present several drawbacks that increase detection times, such as a limited capacity to parallelize computations. Thus, we propose a novel unsupervised GAN-Based IDS that uses temporal convolutional networks (TCNs) and self-attention to replace LSTM networks for considering time dependencies among data. Our proposed approach successfully replaces LSTM networks for attack detection and achieves better detection results. Moreover, it allows different configurations of TCN and self-attention layers to achieve different trade-offs between detection rates and detection times and satisfy different requirements.

In contrast to Chapters 2 and 3, Chapter 4 concerns adversarial attacks that compromise modulation classifiers in wireless receivers, jeopardizing the availability of wireless communications. We verify that the existing adversarial attack techniques either require complete knowledge about the classifier's model, which is an unrealistic assumption, or take too long to craft adversarial perturbations, such that they cannot tamper with the received modulated signals. Thus, we propose a novel black-box adversarial attack technique that reduces the accuracy of modulation classifiers more than other black-box adversarial attacks and crafts adversarial perturbations significantly faster than them. Our proposed technique is essential for assessing the risks of using machine learning-based modulation classifiers in wireless communications.

Finally, given the risks and damage that adversarial attacks may cause, Chapter 5 focuses on studying defense techniques against such sophisticated threats. We verify that only a few defense techniques exist for protecting modulation classifiers from them, most of which only marginally reduce their impact on the classifier's accuracy. Therefore, we propose a defense technique for protecting modulation classifiers from adversarial attacks so that those attacks do not harm the availability of wireless communications. Our proposed approach detects and removes adversarial perturbations while reducing the sensitivity of machine learning-based classifiers to them. Hence, it successfully diminishes the accuracy reduction caused by different adversarial attack techniques.

Keywords: Internet of Things, 5G, 6G, Security, Network Security, Intrusion Detection Systems, Machine Learning, Deep Learning, Generative Adversarial Networks, Modulation Classification, Adversarial Attacks.

TABLE OF CONTENTS

Page

INTRODUCTION 1
0.1 Motivation 1
0.2 Problem Statement 4
0.3 Research Objectives 7
0.4 Contributions and Outline 7
0.5 Related Publications 9
0.6 Related Grants and Awards 10

CHAPTER 1 BACKGROUND AND LITERATURE REVIEW 11
1.1 Intrusion Detection Systems 11
1.1.1 Monitoring Environment 12
1.1.2 Placement Strategy 12
1.1.3 Operation Mode 13
1.1.4 Detection Method 13
1.2 Neural Networks 14
1.2.1 Convolutional Neural Networks 14
1.2.2 Recurrent Neural Networks / Long Short-Term Memory 15
1.2.3 Temporal Convolutional Networks 17
1.2.4 Self-Attention 18
1.3 Generative Adversarial Networks 19
1.4 Adversarial Attacks 21
1.4.1 Adversarial Attacks Taxonomy 22

CHAPTER 2 INTRUSION DETECTION FOR CYBER-PHYSICAL SYSTEMS USING GENERATIVE ADVERSARIAL NETWORKS IN FOG ENVIRONMENT 25
2.1 Abstract 25
2.2 Introduction 26
2.2.1 Related Works 28
2.2.2 Contributions 30
2.2.3 Organization 31
2.3 Proposed FID-GAN Architecture 31
2.3.1 GAN with LSTM-RNN 31
2.3.2 System Architecture and Fast Mapping Encoder 32
2.3.3 Attack Detection Score 34
2.3.4 Fog Architecture and System Model 36
2.4 Methodology and Performance Evaluation 38
2.4.1 Datasets Presentation 38
2.4.2 Simulation Experiments 39
2.5 Results and Discussions 41

2.5.1 Detection Rates 41
2.5.2 Detection Latency 45
2.6 Conclusions 47

CHAPTER 3 UNSUPERVISED GAN-BASED INTRUSION DETECTION SYSTEM USING TEMPORAL CONVOLUTIONAL NETWORKS AND SELF-ATTENTION 49
3.1 Abstract 49
3.2 Introduction 50
3.2.1 Related Works 52
3.2.2 Contributions 54
3.2.3 Organization 55
3.3 DDoS Threat Scenario 55
3.4 Proposed IDS Architecture 56
3.4.1 GAN-based IDSs 57
3.4.2 TCNs 58
3.4.3 Self-Attention 59
3.4.4 Proposed Detection Architecture 60
3.4.5 Proposed Deployment Architecture 62
3.5 Methodology and Performance Evaluation 63
3.5.1 Dataset Presentation 64
3.5.2 Simulation Experiments 65
3.6 Results and Discussions 66
3.6.1 Detection Rates 66
3.6.2 Detection Times 70
3.6.3 Complexity Analysis 72
3.6.4 Combining Protection Techniques 74
3.6.5 Strengths and Limitations 74
3.7 Conclusion 75

CHAPTER 4 MULTI-OBJECTIVE GAN-BASED ADVERSARIAL ATTACK TECHNIQUE FOR MODULATION CLASSIFIERS 77
4.1 Abstract 77
4.2 Introduction 77
4.3 Related Works 79
4.4 Adversarial Attacks Formulation 80
4.5 Proposed Adversarial Attack Technique 81
4.6 Methodology and Experimental Evaluation 84
4.7 Results and Discussion 86
4.8 Conclusion 90

CHAPTER 5 DEFENDING WIRELESS RECEIVERS AGAINST ADVERSARIAL ATTACKS ON MODULATION CLASSIFIERS 91
5.1 Abstract 91

5.2 Introduction 91
5.3 Related Works 95
5.4 Adversarial Attack Threat Model 97
5.5 Proposed Wireless Receiver Architecture 98
5.5.1 Adversarial Perturbation Preprocessor 100
5.5.2 Enhanced Modulation Classifier 101
5.5.3 Adversarial Samples for Training 102
5.6 Methodology and Experimental Evaluation 103
5.6.1 Dataset 103
5.6.2 DAE Experiments 104
5.6.3 EMC Experiments 105
5.6.4 Adversarial Attacks Considered 106
5.7 Results and Discussion 108
5.8 Conclusion 114

CONCLUSION AND RECOMMENDATIONS 117
6.1 Conclusion 117
6.2 Future Work 119
6.2.1 Diffusion-Based Intrusion Detection 119
6.2.2 Minimization of the Number of Training Data Required by Attack Classifiers 120
6.2.3 Security and Privacy of Digital Twins 120
6.2.4 Adversarial Attacks and Defenses on Regression-Based Applications 121

APPENDIX I APPENDIX OF CHAPTER 2 123

BIBLIOGRAPHY 129

LIST OF TABLES

Page

Table 2.1 Equal error rate (EER) 45

Table 3.1 Computational complexity 60

Table 3.2 Training, validation, and testing sets 65

Table 3.3 Malicious network flows per DDoS attack type 65

Table 3.4 Tukey's HSD pairwise group comparisons (95.0% confidence interval) 69

Table 3.5 Accuracy, precision, recall, and F1-scores of our IDS, ALAD, and FID-GAN 70

Table 3.6 Detection rates by DDoS attack type 70

Table 3.7 Computational complexity of each configuration of our IDS 72

Table 3.8 Number of parameters of each configuration of our IDS 73

Table 4.1 Hyper-parameters values 86

Table 4.2 Mean execution time for crafting adversarial samples 89

Table 5.1 Hyper-parameter values of the DAE 105

Table 5.2 Hyper-parameter values of the EMC 106

LIST OF FIGURES

Page

Figure 0.1 Summary of the thesis structure 8

Figure 1.1 Neural network and convolutional neural network architectures (obtained from [Stanford]) 15

Figure 1.2 Different layers of a CNN (obtained from [Amidi & Amidi]) 16

Figure 1.3 Recurrent neural network and LSTM architectures (obtained from [Mittal]) 17

Figure 1.4 General diagram of GANs (obtained from [Silva]) 21

Figure 1.5 Adversarial sample crossing decision boundary 22

Figure 2.1 Autoencoder used to train the proposed Encoder (adapted from [Flores]) 35

Figure 2.2 Encoder and Decoder mapping 35

Figure 2.3 Proposed FID-GAN system model 37

Figure 2.4 ROC curves of the proposed FID-GAN 42

Figure 2.5 ROC curves of the IDS in MAD-GAN [Li *et al.* (2019)] 43

Figure 2.6 ROC curves of the IDS in ALAD [Zenati, Romain, Foo, Lecouat & Chandrasekhar (2018b)] 44

Figure 2.7 Mean detection latency 46

Figure 3.1 The WGAN training framework used in our proposed IDS 58

Figure 3.2 The GAN generator and discriminator architectures 61

Figure 3.3 The TCN block architecture 62

Figure 3.4 The self-attention block architecture 62

Figure 3.5 Our proposed IDS's ROC curves 67

Figure 3.6 ALAD's ROC curve 68

Figure 3.7 FID-GAN's ROC curve 68

Figure 3.8 Detection times of our IDS, ALAD, and FID-GAN 71

Figure 4.1 Our attack model considers the adversarial attacker as malicious software on the wireless receiver 81

Figure 4.2 Our proposed training model 83

Figure 4.3 VT-CNN2 neural network architecture 85

Figure 4.4 GAN generator architecture 85

Figure 4.5 GAN discriminator architecture 85

Figure 4.6 Modulation classifier's accuracy versus PNR with and without our proposed adversarial attack technique 87

Figure 4.7 Waveform comparison of a 8PSK signal with SNR=10 dB before (clean sample) and after (adversarial sample) our proposed adversarial attack 88

Figure 4.8 Modulation classifier's accuracy versus PNR without and subject to different adversarial attack techniques 89

Figure 5.1 Adversarial sample crossing decision boundary 98

Figure 5.2 Adversary attack model as a perturbation transmitted over the air 99

Figure 5.3 Proposed wireless receiver architecture 99

Figure 5.4 DAE neural network architecture 104

Figure 5.5 EMC neural network architecture 106

Figure 5.6 VT-CNN2 modulation classifier's accuracy versus PNR 109

Figure 5.7 Cosine distance between clean and adversarial samples and their reconstructions 110

Figure 5.8 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the FGSM adversarial attack for a SNR of 10 dB 111

Figure 5.9 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the PGD adversarial attack for a SNR of 10 dB 111

Figure 5.10 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the MIM adversarial attack for a SNR of 10 dB 112

Figure 5.11 Modulation classifier's accuracy versus PNR against the FGSM adversarial attack for a SNR of 10 dB 113

Figure 5.12 Modulation classifier's accuracy versus PNR against the PGD adversarial attack for a SNR of 10 dB 113

Figure 5.13 Modulation classifier's accuracy versus PNR against the MIM adversarial attack for a SNR of 10 dB 114

LIST OF ALGORITHMS

Page

Algorithm 2.1 Novel attack detection system 36

Algorithm 4.1 Proposed adversarial attack technique 83

Algorithm 5.1 Proposed defense technique 101

LIST OF ABBREVIATIONS

5G	Fifth-Generation
6G	Sixth-Generation
ADC	Analog-to-Digital Converter
AMC	Automatic Modulation Classification
APP	Adversarial Perturbation Preprocessor
AUC	Area Under The Curve
AUCROC	Area Under The Receiver Operating Characteristic Curve
AWGN	Additive White Gaussian Noise
BIM	Basic Iterative Method
BO-GP	Bayesian Optimization-Based Gaussian Process
CAT	Customized Adversarial Training
CD	Cosine Distance
CE	Cross Entropy
CIC	Canadian Institute for Cybersecurity
CNN	Convolutional Neural Networks
CPS	Cyber-Physical Systems
DAE	Denoising Autoencoder
DDoS	Distributed Denial of Service
DNS	Domain Name System

DoS	Denial of Service
DT	Decision Tree
EER	Equal Error Rate
EMC	Enhanced Modulation Classifier
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Network
GNA	Gaussian Noise Augmentation
HIDS	Host-Based Intrusion Detection System
HTRD	Hybrid Training-Time and Run-Time Defense
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IoT	Internet of Things
LAN	Local-Area Network
LS	Label Smoothing
LSTM	Long Short-Term Memory
ML	Machine Learning
MGAN	Mixture Generative Adversarial Networks
MHA	Multi-Head Attention
MI-FGSM	Momentum Iterative Fast Gradient Sign Method

MIM	Momentum Iterative Method
MSE	Mean Squared Error
NIDS	Network-Based Intrusion Detection System
NR	Neural Rejection
PGD	Projected Gradient Descent
PNR	Perturbation-to-Noise Ratio
QoS	Quality of Service
R2L	Remote to Local
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SNR	Signal-to-Noise Ratio
SOC	Security Operation Center
SVM	Support Vector Machine
SWaT	Secure Water Treatment
TCN	Temporal Convolutional Network
TCP	Transmission Control Protocol
U2R	User to Root
UAP	Universal Adversarial Perturbation
UDP	User Datagram Protocol

UNB	University of New Brunswick
VAE	Variational Autoencoder
VoIP	Voice-Over-IP
VQA	Visual Question Answering
WADI	Water Distribution
WAF	Web Application Firewall
WGAN	Wasserstein GAN
WGAN-GP	WGAN Gradient Penalty

INTRODUCTION

0.1 Motivation

The increasing growth of connected devices, such as sensors, actuators, home appliances, vehicles, and many others, is changing the way we interact with our surroundings. This is reducing the gap between the physical and digital worlds and integrating devices into large-scale platforms that acquire and process data to produce automated decisions while also generating knowledge and information [Rodriguez (2015); Santos, Leroux, Wauters, Volckaert & De Turck (2018)]. This plethora of smart and connected devices compose smart-cities, industry 4.0, and, in general, the Internet of things (IoT). It creates a whole new world of possibilities and services, such as intelligent traffic lights, automated water treatment plants, and personal health monitoring applications [Li, Da Xu & Zhao (2018c); Osseiran, Monserrat & Marsch (2016)]. Moreover, this connected environment is expected to even further increase with the deployment of the fifth-generation (5G) and the development of the sixth-generation (6G) of wireless/mobile communications, which are expected to provide connectivity to a massive number of devices with highly diverse requirements [Illy, Kaddoum, Miranda Moreira, Kaur & Garg (2019); Sharma, Kalbarczyk, Barlow & Iyer (2011); Saad, Bennis & Chen (2020)].

On the other hand, the broadcast nature of wireless communications enables attackers to eavesdrop and inject malicious data into the network and launch several cyber-attacks to compromise the cyber-security goals, i.e., confidentiality, integrity, and availability [Finney (2014); Hachimi, Kaddoum, Gagnon & Illy (2020); Pourranjbar, Kaddoum & Saad (2022b)]. Confidentiality aims to protect information such that it can only be understood by the receiver and sender, i.e., third parties must not be able to understand the data even if they have access to it. Integrity, on the other hand, aims to ensure that data is not changed without authorization, i.e., that data is not tampered. Finally, availability aims to guarantee that data is available and accessible whenever it is needed, i.e., that systems are always fully functional and reliable.

Therefore, the widespread adoption of IoT introduces several security threats that may cause inaccurate sensing and control of systems [Alguliyev, Imamverdiyev & Sukhostat (2018); Han, Xie, Chen & Ling (2014)].

Among those threats, cyber-attacks targeting availability may completely interrupt the operation of systems, causing financial losses and putting at risk people's lives [Ali *et al.* (2020); Jia, Zhong, Alrawais, Gong & Cheng (2020); Ibitoye, Abou-Khamis, Matrawy & Shafiq (2019); Meftah, Kaddoum, Do & Talhi (2022); Pourranjbar, Kaddoum & Aghababaiyan (2022a)]. Denial of service (DoS) and distributed DoS (DDoS) attacks, for example, attempt to exhaust a system's or network's resources by forcing compromised devices to unnecessarily request resources until there is no resource left for legitimate users [Jia *et al.* (2020)]. Recently, a DDoS attack on a large domain name system (DNS) provider caused disruptions to many services, such as Airbnb, Netflix, PayPal, Visa, Amazon, The New York Times, and GitHub [Cloudfare; Nicholson]. Similarly, cyber-criminals have disrupted Internet service providers and voice-over-IP (VoIP) operations worldwide and threatened several organizations with DDoS incursions unless extortion demands are met [R. Dobbins and S. Bjarnason; Roland Dobbins and Steinthor Bjarnason]. Consequences can be even worse on critical systems. Autonomous vehicles cannot afford to lose access to their obstacle recognition or breaking systems [Baza *et al.* (2021)]; otherwise, accidents could occur. Likewise, one cannot afford their implantable medical devices, such as pacemakers and insulin pumps, to run out of battery due to the multiple message transmissions of DoS attacks, as such failures could be fatal [Vakhter, Soysal, Schaumont & Guler (2022)].

Moreover, while machine learning (ML) is being largely adopted in many applications and domains due to its powerful classification capabilities, it also introduces new risks and vulnerabilities. Adversarial attacks craft and introduce small perturbations that fool machine learning models into making wrong decisions, which then may significantly impact the security of

applications [Chakraborty, Alam, Dey, Chattopadhyay & Mukhopadhyay (2018); Yuan, He, Zhu & Li (2019)]. Hence, just as cyber-attacks do, adversarial attacks may compromise the security of systems and networks, impacting their confidentiality, integrity, and availability. For instance, while deep learning models have been increasingly adopted for several wireless communication tasks [Gingras, Pourranjbar & Kaddoum (2020); Nguyen, Kaddoum, Selim, Srinivas & Freitas de Araujo-Filho (2022)], such as channel encoding and decoding [Liang, Shen & Wu (2018)], resource allocation [Sanguinetti, Zappone & Debbah (2018); Sun *et al.* (2017)], and automatic modulation classification (AMC) [O'Shea, Corgan & Clancy (2016); O'Shea, Roy & Clancy (2018)], adversarial attacks may compromise them and jeopardize the wireless communication's availability. The works in [Freitas de Araujo-Filho, Kaddoum, Naili, Fapi & Zhu (2022); Lin, Zhao, Ma, Tu & Wang (2021)] show, for example, that adversarial attacks compromise ML-based modulation classifiers used in wireless receivers to identify which scheme has been used to modulate signals in wireless transmitters. As a result, wireless receivers cannot correctly demodulate signals, and communication is interrupted [Freitas de Araujo-Filho *et al.* (2022); Lin *et al.* (2021)].

Despite numerous security solutions available on the traditional Internet, the IoT's physical constraints, highly heterogeneous environment, and the use of ML impose new security challenges [Miranda, Kaddoum, Boukhtouta, Madi & Alameddine (2022); Naeem, Ali & Kaddoum (2023); Illy, Kaddoum, Freitas de Araujo-Filho, Kaur & Garg (2022); Garg *et al.* (2020); Pourranjbar, Elleuch, Landry-pellerin & Kaddoum (2023)]. For instance, the heterogeneity brought by different access technologies, applications, and requirements significantly increases the attacks surfaces and the threat from new types of attacks [Abeshu & Chilamkurti (Feb, 2018); Papamartzivanos, Gómez Mármol & Kambourakis (2019); Midi, Rullo, Mudgerikar & Bertino (2017)]. On the other hand, the limited battery and computing power of most IoT devices thwart the deployment of most security mechanisms based on cryptography and authentication [Abeshu & Chilamkurti (Feb, 2018); Yang, Wu, Yin, Li & Zhao (2017)]. Finally, since

adversarial attacks have yet to be exploited in many fields, it is still not clear the extent to which they can compromise the availability of systems and how to make ML-based systems resistant to them.

To overcome these challenges, intrusion detection systems (IDSs), which detect attacks when other security mechanisms fail, have emerged as a fundamental component to protect and secure systems and networks [Chaabouni, Mosbah, Zemmari, Sauvignac & Faruki (2019); Li *et al.* (2019); Jia *et al.* (2020)]. In contrast to other approaches, anomaly-based IDSs detect attacks by measuring deviations between data patterns and what is considered to be a normal behavior [Nisioti, Mylonas, Yoo & Katos (2018)]. Unsupervised anomaly-based IDSs go one step further by not requiring any knowledge or previous occurrences of attacks. Thus, they can detect both known and unknown attacks, which is an essential feature as new types of attacks are launched daily [Nisioti *et al.* (2018)]. Although existing ML-based IDSs have been showing promising results at detecting attacks [Abeshu & Chilamkurti (Feb, 2018); Vigneswaran, Vinayakumar, Soman & Poornachandran (2018); Shone, Ngoc, Phai & Shi (Feb, 2018)], there are still several challenges and open issues that limit their efficiency and effectiveness. Furthermore, detecting adversarial attacks that jeopardize wireless communication tasks, such as the correct demodulation of signals, might not be enough, as their availability would still be compromised despite our awareness. Therefore, in addition to effectively and efficiently detecting cyber-attacks, it is also urgently necessary to protect ML-based systems from adversarial attacks.

0.2 Problem Statement

Since new cyber-attacks are constantly launched, IDSs must be able to detect both known and zero-day attacks. In addition, since obtaining labeled attack data is very challenging and time-consuming, if not impossible, e.g., for zero-day attacks, IDSs need to consider unlabeled data [Choi, Kim, Lee & Kim (Sep, 2019); Schlegl, Seeböck, Waldstein, Langs & Schmidt-Erfurth (May, 2019); Ozgumus (2019)]. Thus, unsupervised learning techniques are deemed best for

detecting cyber-attacks [Mitchell & Chen (Apr, 2014); Zarpelao, Miani, Kawakani & de Alvarenga (Apr, 2017); Nisioti *et al.* (2018)]. However, most existing unsupervised techniques are not able to deal with the non-linearity and inherent correlations in multivariate time series, which is the case of a considerable amount of real-world data, including data streams generated by sensors in IoT and cyber-physical systems (CPSs) [Li *et al.* (2019); Li & Wen (Jan, 2014); Goh, Adepu, Tan & Lee (2017)]. Moreover, even when using state-of-the-art deep learning algorithms, most existing unsupervised IDSs present high false positive rates, which can make the operation of security operation centers (SOCs) unfeasible as security analysts would have to analyze many false alarms [Prabavathy, Sundarakantham & Shalinie (2018)]. Therefore, it is necessary to investigate and propose novel unsupervised IDSs that simultaneously achieve low false positive and negative rates.

Moreover, since cyber-attacks need to be detected and stopped before causing damage, the detection time, i.e., the time between the start and the detection of an attack, needs to be as short as possible. However, most state-of-the-art IDSs have long detection times [Li *et al.* (2019)] because they rely on complex neural networks that have many layers, and on long short-term memory (LSTM) neural networks. Although LSTM networks improve detection results by considering time dependencies among data, their limited capacity to parallelize computations increases the detection time [Hollis, Viscardi & Yi (2018); Bai, Kolter & Koltun (2018); Vaswani *et al.* (2017); Huang *et al.* (2020)]. In addition, recent studies show that LSTM's sequential processing of data significantly increases the computational complexity and challenges LSTM's performance on devices with limited computational power and memory [Duc, Minh, Xuan & Kamioka (2020)]. Therefore, it is necessary to optimize detection algorithms and investigate other neural network architectures that consider time dependencies among data while allowing the fast detection of intrusions such that cyber-attacks are stopped before causing damage.

Finally, it has been shown that ML-based systems are vulnerable to adversarial attacks, which can cause severe security issues by putting at risk the availability of systems that rely on ML [Ibitoye *et al.* (2019)]. Adversaries can, for example, craft perturbations and manipulate legitimate inputs to force ML-based modulation classifiers to produce incorrect outputs and interrupt wireless communications [Chakraborty *et al.* (2018); Freitas de Araujo-Filho *et al.* (2022)]. Despite such risks, most studies on adversarial attacks are focused on image classifiers [Usama, Asim, Latif, Qadir *et al.* (2019); Samangouei, Kabkab & Chellappa (2018)]. Moreover, only a few works have proposed techniques to defend connected objects from such attacks, most of which only marginally reduce the impact of the attacks [Zhang *et al.* (2022); Zhang, Lambotharan, Zheng, AsSadhan & Roli (2021a)]. Therefore, further investigations are required to ensure the security of systems against adversarial attacks.

Despite the existing security solutions and the different approaches that have been presented to detect cyber-attacks and protect systems from adversarial attacks, there are still a variety of issues to be tackled. After conducting an extensive literature review, we reached a few concluding remarks and identified the following open challenges that our thesis aims to solve:

- While IDSs should not rely on labelled data, most of them present high false positive rates and struggle with the time required to detect intrusions. Thus, it is necessary to propose new detection solutions that reduce the detection time and achieve low false positive and false negative rates.
- While LSTM networks are heavily used by state-of-the-art intrusion detection systems, they present several drawbacks that put in doubt their status as the standard architecture for sequence modeling tasks. Thus, it is necessary to investigate novel strategies for considering time-dependencies among data.
- Although adversarial attacks may significantly compromise the security of systems that rely on ML, their study is still in its early stages. Thus, it is necessary to investigate the impact

of adversarial attacks on different application domains and propose techniques to enhance systems' security against them.

0.3 Research Objectives

Although cyber-attacks and adversarial attacks represent different techniques for compromising security, their effects are the same, as they can severely compromise confidentiality, integrity, and availability. Hence, given their potential impact, the hypothesis that guides our research is whether artificial intelligence enhances security by effectively and efficiently detecting attacks or harms security due to the vulnerabilities it adds. Therefore, in our research, we aim to advance the state-of-the-art in the security field by addressing the aforementioned identified challenges. Our main goal is to enhance the security of systems by effectively and efficiently detecting cyber-attacks while also defending systems that rely on ML from adversarial attacks. To achieve our goal, we define the following four specific objectives:

1. Propose an unsupervised IDS that reduces the detection time of the current state-of-the-art solutions, making it more suitable for latency-constrained applications.
2. Propose an unsupervised IDS that considers time-dependencies among data without relying on LSTM networks, such that their drawbacks are avoided.
3. Propose an adversarial attack technique and investigate the extent to which it may jeopardize security by compromising the availability of systems.
4. Investigate and propose a defense technique that protects ML-based systems from adversarial attacks.

0.4 Contributions and Outline

Our thesis is structured as shown in Fig. 0.1, and detailed as follows.

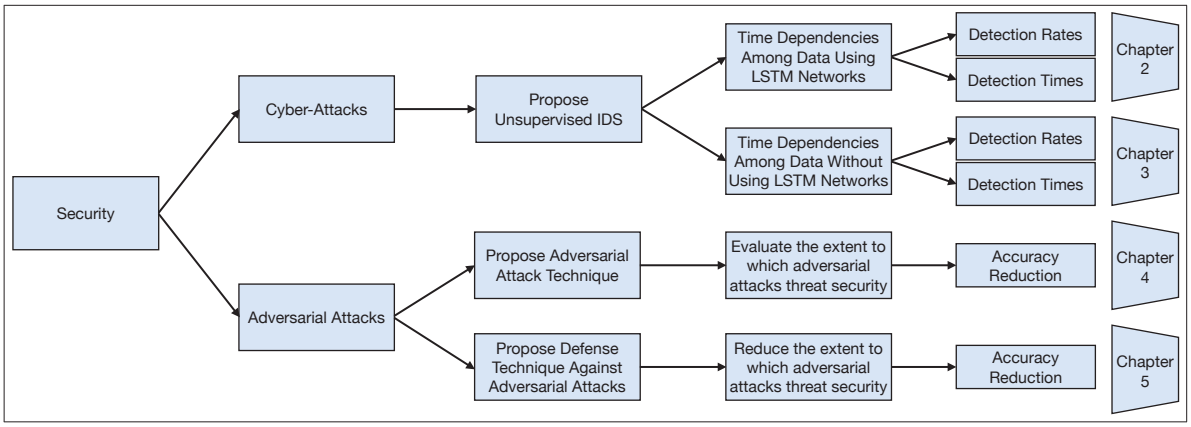


Figure 0.1 Summary of the thesis structure

Chapter 1 presents the technical background and literature review necessary for our work. We first introduce intrusion detection systems and their taxonomy. Then, we discuss neural network architectures and frameworks used in our work, such as convolutional neural networks (CNNs), LSTMs, temporal convolutional networks (TCNs), self-attention, and generative adversarial networks (GANs). Finally, we formulate adversarial attacks and introduce their taxonomy.

In Chapter 2, we evaluate the unsupervised detection of cyber-attacks problem using LSTM networks and GANs, which is a promising deep learning framework that simultaneously trains two neural networks: a generator and a discriminator. We show that we can combine the generator and discriminator networks to compute an anomaly detection score that indicates whether samples are malicious with higher detection rates than when only one of those networks is used. Moreover, we show that relying on an additional third neural network can accelerate the anomaly detection score computation, thus significantly reducing the detection time.

In Chapter 3, we focus on detecting DDoS attacks, which significantly impact the availability of systems, and investigating different neural network architectures that could replace LSTM networks for considering time dependencies among data in GAN-based IDSs. We show that

IDSs can combine TCN and self-attention layers to achieve different trade-offs between detection rates and detection times while outperforming IDSs that rely on LSTM networks.

In Chapter 4, we formulate adversarial attacks and show how they pose a serious security problem by significantly compromising the availability of wireless communications. We show that powerful adversarial perturbations can be crafted by modifying GANs and combining them to the multi-task loss [Kendall, Gal & Cipolla (2018)] so that they significantly reduce the accuracy of modulation classifiers in wireless receivers, consequently interrupting communication.

In Chapter 5, we review the existing techniques for defending modulation classifiers from adversarial attacks. Then, we show that it is possible to significantly diminish the impact of adversarial attacks by estimating and removing large adversarial samples with a specially trained denoising autoencoder (DAE).

Finally, we conclude our thesis by summarizing the conclusions from the main chapters and presenting recommendations for future works in Chapter 6.

0.5 Related Publications

The author's Ph.D. research contributed to the following published and submitted journal research articles.

P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macêdo and C. Zanchettin, "Intrusion Detection for Cyber–Physical Systems Using Generative Adversarial Networks in Fog Environment", in *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247-6256, April 15, 2021, doi: 10.1109/JIOT.2020.3024800.

P. Freitas de Araujo-Filho, M. Naili, G. Kaddoum, E. T. Fapi and Z. Zhu, "Unsupervised GAN-Based Intrusion Detection System Using Temporal Convolutional Networks and Self-Attention", in *IEEE Transactions on Network and Service Management*, doi: 10.1109/TNSM.2023.3260039.

P. Freitas de Araujo-Filho, G. Kaddoum, M. Naili, E. T. Fapi and Z. Zhu, "Multi-Objective GAN-Based Adversarial Attack Technique for Modulation Classifiers", in *IEEE Communications Letters*, April, 2022, doi: 10.1109/LCOMM.2022.3167368.

P. Freitas de Araujo-Filho, G. Kaddoum, M. C. B. Nasr, H. F. Arcoverde, and D. R. Campelo, "Defending Wireless Receivers Against Adversarial Attacks on Modulation Classifiers", submitted to *IEEE Internet of Things Journal*.

Besides the above articles, which contribute to the main contents of this thesis, the complete list of publications that the author was involved in during his Ph.D. research is given at the end of this thesis.

0.6 Related Grants and Awards

The author's Ph.D. research was recognized with the following grants and awards:

- Microsoft Research Ph.D. Fellowship (2022)
  One of the two 2022 Microsoft Research Ph.D. Fellowship recipients in Security, Privacy, and Cryptography, from a total of 36 recipients in all areas worldwide.
- Fonds de recherche du Québec - B2X Scholarship (2021-2022)
  First place in the FRQNT's B2X Doctoral Scholarship in the 2021-2022 competition.

CHAPTER 1

BACKGROUND AND LITERATURE REVIEW

In this chapter, we present the technical background and literature review necessary for the development and understanding of our thesis.

1.1 Intrusion Detection Systems

Intrusion Detection Systems are reactive systems that monitor the network traffic and system-level applications to detect and report malicious activities carried out by internal or external intruders. Internal intruders are users that already have some degree of legitimate access to a system or network and that are attempting to raise that access privilege and misuse it. On the other hand, external intruders do not have any access authorization to a system or network and attempt to gain and misuse it.

Despite the different security mechanisms used, systems and networks might still be subject to cyber-attacks. Therefore, intrusion detection systems have a fundamental role as a second line of defense and are the last resource when other security solutions fail. In order to be effective, IDSs need to meet several requirements. For instance, they need to allow dynamic reconfiguration, run continually with minimal human supervision, produce minimal overhead and degradation of service, and be scalable to serve a large number of users.

Typical IDSs are usually composed of three components: agents, directors, and notifier. Agents, or sensors, are responsible for collecting and sending data to the directors, which then analyze all data received and reach a decision of whether an intrusion is occurring or not. Finally, the notifier system receives the directors' decisions and generates an alert when an intrusion is detected. Depending on the number, distribution, and working mode of agents and directors, IDSs can be classified according to their monitoring environment, placement strategy, operation mode, and detection method.

1.1.1 Monitoring Environment

IDSs can have different monitoring environments, and then be classified as host-based IDSs (HIDSs) or as network-based IDSs (NIDSs). HIDSs monitor and analyze activities related to a single host machine. They detect intrusions by monitoring running processes, file-system changes, inter-process communications, application logs, and operating system logs. HIDSs are preferred for insider intrusions detection and benefit from lower volumes of traffic, overheads, and detection times. However, they only detect intrusions on a specific host, they become vulnerable when the host operating system is compromised, and they are more expensive and challenging to implement.

On the other hand, instead of monitoring and analyzing information from a single host, NIDSs detect intrusions by monitoring and analyzing the traffic that passes through a network. They are preferred against external intrusions and network-based attacks, such as the DoS attacks. In addition, they can protect the whole network and are less expensive and less complex to implement. However, NIDSs generate large amounts of data, large overheads, and cannot deal with encrypted network traffic.

1.1.2 Placement Strategy

Intrusion detection systems may have different placement strategies, depending on where they are deployed. In the centralized approach, IDSs are placed in a central location, such as a router or a dedicated host, to which all data is sent for analysis. In this context, a single node may offer more computing and battery resources for deploying the IDS; however, the IDS may be completely jeopardized is the node is compromised. On the other hand, in the distributed approach, IDSs are placed in different nodes to which all data or part of the data is sent for analysis. Since most of the nodes usually have constrained resources, e.g., computing power and battery, such IDSs must be very optimized. Finally, the hybrid strategy combines concepts of centralized and distributed placements to take advantage of their strengths and overcome their drawbacks. One possible hybrid approach is to organize a network in clusters, where the IDSs

placed in each cluster are responsible for monitoring the nodes within the cluster. Such strategy requires only a few nodes to have more resources for deploying the detection solutions.

1.1.3 Operation Mode

Regarding the operation, IDSs can work in an offline or a real-time manner. In the former, the detection of intrusions does not need to respect a deadline, i.e., intrusions can be detected whenever possible, regardless of possible damages. On the other hand, real-time IDSs are required to detect intrusions promptly, such that an alert is emitted while the intrusion is still occurring. Here, although more challenging, real-time detection is essential for stopping intrusions and preventing damages.

1.1.4 Detection Method

Depending on their detection method, IDSs can be classified into signature-based IDS, anomaly-based IDS, specification-based IDS, and hybrid IDS. Signature-based IDSs, also known as misuse IDSs, detect intrusions by comparing events and data patterns that correspond to the system or network behavior to signatures of known attacks stored in the IDS. If there is a match with a stored signature, an intrusion represented by that signature is detected. This approach is usually very accurate and effective for detecting known threats, and achieves low false positive rates. However, it has two major drawbacks, where it may require a large memory for storing signatures and can only detect known attacks, i.e., zero-day attacks for which there is no signature available cannot be detected.

On the other hand, instead of comparing events to signatures looking for a match, Anomaly-based IDSs compare events to a normal behavior profile, such that large deviations from this profile indicate attacks. The normal behavior profile, which corresponds to the system's or network's normal functioning, can be built through thresholds or ML algorithms that identify patterns corresponding to the normal behavior. Although anomaly-based IDSs usually present higher false positive rates, they are capable of detecting unknown attacks.

Specification-based IDSs also detect intrusions by comparing observed events to what is considered to be a normal system or network behavior. However, in addition to that, the input data is also compared to the specifications of the system. For instance, a vehicle manufacturer knows that the engine oil temperature of their car ranges between two values. Thus, in addition to the normal behavior profile, an IDS can also use that range to detect intrusions. The specification-based approach usually achieves lower false positive rates than the anomaly-based approach, at the expense of requiring knowledge of the system. Finally, hybrid-approaches combine the previous methods to take advantage of their strengths and overcome their limitations.

1.2 Neural Networks

Neural networks are composed of neurons that perform a dot product and apply an optional non-linear function to a received input. These neurons are organized and divided into one input layer, one or more hidden layers, and one output layer. Each neuron is fully connected to all neurons in the previous layer and is entirely independent of the other neurons in its own layer. Although useful for many problems, this architecture does not consider dependencies among data and does not scale well for high dimensional inputs, such as images, due to the large number of connections and parameters required. For instance, a single fully-connected neuron in the first hidden layer of a neural network would have 3072 (32*32*3) weights for input images of size 32x32x3 (32 wide, 32 high, 3 color channels), or 120,000 (200*200*3) weights for input images of size 200x200x3.

1.2.1 Convolutional Neural Networks

Just as regular neural networks, CNNs are composed of layers of neurons with learnable weights and biases. The whole network expresses a single differentiable score function, such that class scores are obtained from the input patterns. However, by encoding some properties into its architecture, the network may have much fewer parameters, and its forward function also becomes more efficient. In contrast to regular neural networks, the neurons of a CNN's layers are arranged in three dimensions: width, height, and depth. Accordingly, the inputs considered

for that architecture are also three dimensional volumes, instead of a single dimensional pattern. Figure 1.1a shows the architecture of a regular neural network with two hidden layers while Figure 1.1b exhibits the architecture of a CNN.

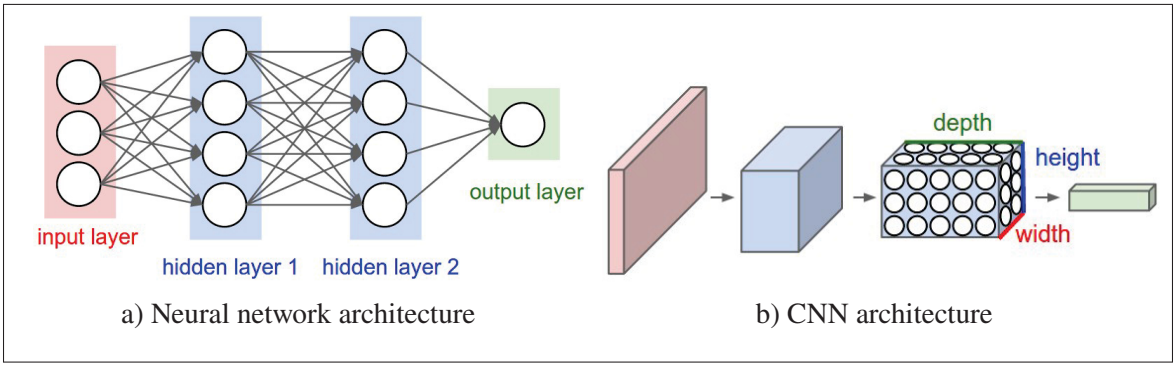


Figure 1.1 Neural network and convolutional neural network architectures (obtained from [Stanford])

Convolutional neural networks may include sequences of different layers, such as fully-connected layer, convolutional layer, and pooling layer. The neurons of fully-connected layers are connected to all neurons in the previous layers and work just as the layers of regular neural networks. On the other hand, the neurons of convolutional layers are connected to only small regions of the previous layer, which significantly reduces the network's number of parameters. These layers use filters and perform convolution operations, which reduces the network's size. Pooling layers perform a downsampling operation, by for example taking the maximum or average value along the network's width and height dimensions, consequently reducing the network's size. Finally, the output layer reduces the inputs into a single vector of class scores in the depth dimension. Figures 1.2a, 1.2b, and 1.2c show a fully-connected, convolutional, and max pooling layers, respectively.

1.2.2 Recurrent Neural Networks / Long Short-Term Memory

Traditional neural networks consider that all input patterns are independent of each other and cannot deal with dependencies among data. However, some applications have inherent dependencies among their data and need to consider sequences of data and how one pattern

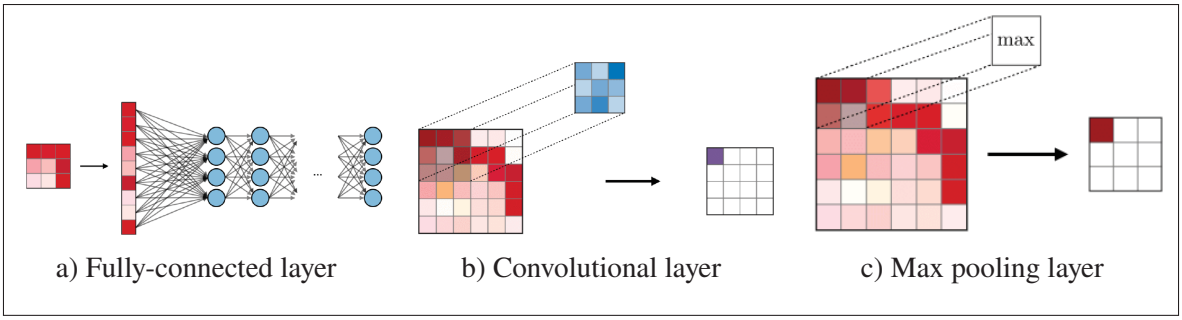


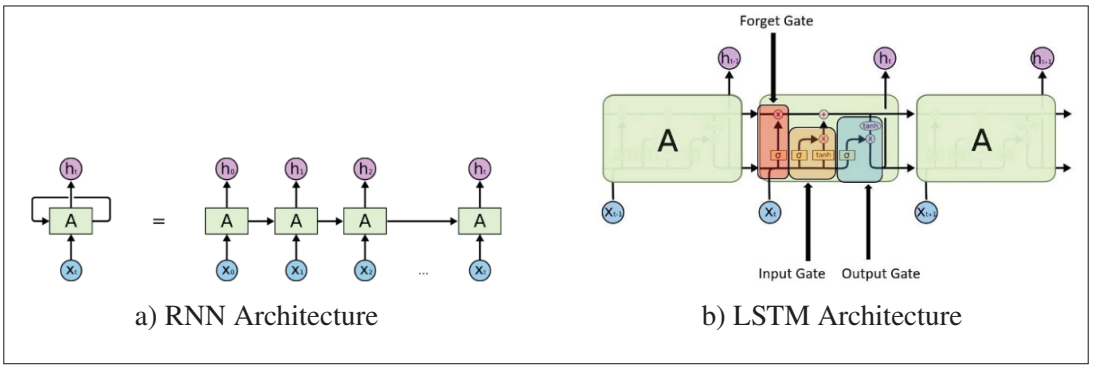
Figure 1.2 Different layers of a CNN (obtained from [Amidi & Amidi])

relates and affects the other. For instance, in natural language processing or speech recognition tasks, neural networks are required to process sequences of words and sounds to recognize meaningful information. Similarly, data streams generated by sensors in cyber-physical systems have time dependencies between them that could contribute to classification tasks if considered. In order to deal with sequences of data, recurrent neural networks (RNNs) allow the previous outputs to be used, and computations are performed for every element of a sequence such that the computation outputs for one element of the sequence serves as input for the computation of the following element in that sequence.

Besides considering dependencies among data, RNNs have the advantages of not increasing the model size with the input size and the possibility of processing inputs with any length. On the other hand, RNNs usually present longer computation times, difficulties in connecting previous information to the present task when there is a large gap between them, and vanishing and exploding gradient problems. Since the derivatives of the hidden layers are multiplied by each other, if they are too large, the gradient exponentially increases through the network and eventually explodes, making the model unstable. However, if the derivatives are too small, the gradient exponentially decreases through the model until it vanishes, and thus the model is unable to learn by not having its weights sufficiently updated.

In this context, LSTM units have been proposed to deal with the vanishing gradient problem, and with significant gaps between past and current data, i.e., they are capable of learning long-term dependencies. LSTM networks are a modified version of RNNs that have, for its repeating

module, a different structure composed of an input gate, a forget gate, and an output gate. The input gate decides what new information should be stored in the cell state. The forget gate decides what information should be discarded from the cell state. Finally, the output gate filters the cell state and decides what information should be outputted. Figures 1.3a and 1.3b depict the RNN and LSTM architectures, respectively.



a) RNN Architecture b) LSTM Architecture

Figure 1.3 Recurrent neural network and LSTM architectures (obtained from [Mittal])

1.2.3 Temporal Convolutional Networks

TCNs refer to modified convolutional architectures for sequence prediction tasks. They map input sequences to output sequences of the same length and use causal convolutions, i.e., convolutions that use only information from the past. Thus, an output at time t is convolved only with elements from times earlier than t in the previous layer. In addition, TCNs also use dilated convolutions to enable the architecture to look far into the past. Thus, for an input sequence $x \in \mathbb{R}^n$ and a filter $f : \{0, ..., k-1\} \rightarrow \mathbb{R}$, the dilated convolution on element s of the sequence is defined as

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) x_{s-di}, \tag{1.1}$$

where k is the filter size and d is the dilation factor. Finally, TCN networks allow a residual connection so the architecture learns what modifications are imposed on the data rather than only modifying it. This connection contributes to avoiding the gradient vanishing problem and

consists of adding the input x to the output of a series of transformations T. It is given by

$$O(x) = \Phi(x + T(x)), \tag{1.2}$$

where Φ is an activation function.

TCNs provide a powerful way to extract temporal dependencies from data and have been shown to have several advantages over LSTM networks for modeling sequences. For instanve, computations can be performed in parallel since the same filter can be used in all layers, and input sequences can be processed as a whole. This means TCNs do not need to store the partial results of computations and thus consume less memory during training. Finally, TCNs have been shown to have more stable gradients, which avoids the gradient vanishing and exploding problems [Bai *et al.* (2018); Duc *et al.* (2020)].

1.2.4 Self-Attention

Attention functions are defined as the mapping of a matrix of queries Q, a matrix of keys K, and a matrix of values V to an output. Scaled dot product attention is one type of attention function, which computes a context matrix C as

$$C = Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1.3}$$

where d_k is the dimension of Values. Matrices K and V usually correspond to input sequences x, whereas matrix Q is composed of randomly initialized trainable parameters. The dot product of Q and K^T gives a measure of the pairwise similarity between the query and key matrices, which results in an attention score. Thus, the matrix C represents the intrinsic dependencies between representations of a sequence.

Moreover, it has been shown that using linearly projected queries, keys, and values h times with learned linear projections contributes to extracting relationships between data [Li, Zhang, Lv & Wang (2021); Vaswani *et al.* (2017)]. Thus, multi-head attention (MHA) modules perform

attention functions in parallel on each of the projected versions of queries, keys, and values, and then concatenate their outputs as

$$MHA(Q, K, V) = W^0 Concat(head_1, ..., head_h), \tag{1.4}$$

where W^0 is a parameter matrix for the concatenation operation and $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. W_i^Q, W_i^K, and W_i^V are parameter matrices that project queries, keys, and values, respectively. Finally, self-attention considers that all the keys, values, and queries come from the same place, such as the output of the previous layer in a neural network. This allows modules to capture in-depth contextual information and relationships between data.

Similarly to TCNs, attention mechanisms make it possible to extract dependencies among data and have been shown to outperform LSTM networks in several sequence modeling tasks. They are more capable of extracting features than LSTM networks, which produces more accurate models [Li *et al.* (2021)]. In addition, they can process sequences as a whole and they enable more computation parallelization as MHA heads can run in parallel. Furthermore, while LSTM networks require $O(n)$ sequential operations, TCN, self-attention, and MHA layers have a constant number of sequentially executed operations.

1.3 Generative Adversarial Networks

Generative Models are a powerful method for learning the probabilistic distribution of a training set, such that it is possible to generate new samples of that same distribution. One of the most efficient generative models are GANs, which provide a powerful modeling framework able to cope with high-dimensional data.

GANs estimate generative models through an adversarial process by simultaneously training two competing networks: a generator G and a discriminator D. The generator network is trained to produce synthetic data examples that are similar to real data patterns by taking a random vector z, drawn from an input distribution $P_z(z)$ in a latent Z-Space. Thus, it captures the hidden distribution of the training samples and can be seen as an implicit model of the system.

On the other hand, the discriminator network is trained to distinguish and classify synthetic examples produced by the generator and real data samples from the training set. The two models are trained together in a zero-sum adversarial minimax game, in which the generator tries to maximize the probability of producing outputs recognized as real, and the discriminator tries to minimize that same probability [Goodfellow, Bengio & Courville (2016); Goodfellow *et al.* (2014); Schlegl *et al.* (May, 2019)]. Thus, they can be regarded as two agents playing a minimax game with value function $V(G, D)$ as in

$$\min_{\mathrm{G}}\max_{\mathrm{D}} V(D, G) = \mathcal{E}_{x\sim P_{\mathrm{data}}(x)}[\log D(x)] + \mathcal{E}_{z\sim P_z(z)}[1 - \log D(G(x))]. \tag{1.5}$$

Since GANs might be challenging to train and suffer from the gradient vanishing problem [Arjovsky, Chintala & Bottou (2017); Creswell *et al.* (2018)], researchers have proposed variations of the original GAN formulation to solve such drawbacks. Thus, the Wasserstein GAN (WGAN) trains a GAN by relying on the Wasserstein distance between two probability distributions [Creswell *et al.* (2018)]. Its discriminator estimates the Wasserstein distance by maximizing the difference between average critic score on real and fake samples, i.e., by minimizing the discriminator loss given by $L_D = D(G(z)) - D(x)$. On the other hand, the WGAN generator has the opposite goal of maximizing the average critic score on fake samples by minimizing the generator loss given by $L_G = -D(G(z))$ [Arjovsky *et al.* (2017); Creswell *et al.* (2018)]. Furthermore, as generative artificial intelligence is an active research field, other GAN formulations, such as WGAN Gradient Penalty (WGAN-GP) [Gulrajani, Ahmed, Arjovsky, Dumoulin & Courville (2017)] and Instance-Conditioned GAN [Casanova, Careil, Verbeek, Drozdzal & Romero Soriano (2021)], are being proposed to further advance the remarkable results that GANs have been achieving.

Regardless of the GAN formulation, since no labels are required, GANs are used in unsupervised problems to find an implicit probability distribution and model of the system, while also providing a model D to detect generated or fake samples. Thus, they present a promising approach to tackle the challenge of developing effective unsupervised anomaly detection methods for cyber-attacks

with probability distributions difficult to estimate. Figure 1.4 exhibits a general diagram of GANs.

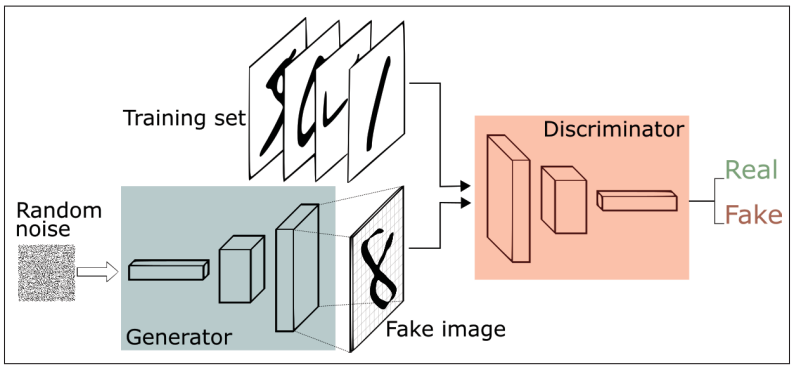


Figure 1.4 General diagram of GANs
(obtained from [Silva])

1.4 Adversarial Attacks

Although deep learning models may be trained with large amounts of data, it is impractical to train them to cover all possible input feature vectors. As a result, the decision boundary found by a trained model may differ from the real one. Such discrepancy creates room for a trained model to make mistakes [Lin *et al.* (2021)]. Thus, adversarial attacks craft perturbations to adulterate data samples so that they fall within that discrepancy area and are misclassified by a trained model, as shown in Figure 1.5. However, this is not a trivial task as those perturbations must be large enough to cause misclassifications but small enough not to be perceptible. Therefore, given a sample x, the goal of an adversarial attacker is to find a perturbation δ and construct an adversarial sample $x_{adv} = x + \delta$ while satisfying

$$\min ||x_{adv} - x|| < \rho \tag{1.6}$$

and

$$f(x_{adv}) \neq f(x), \tag{1.7}$$

where $||\cdot||$ represents a chosen distance metric, ρ is the maximum imperceptible perturbation according to that metric, and f is the already trained classifier target of the attack.

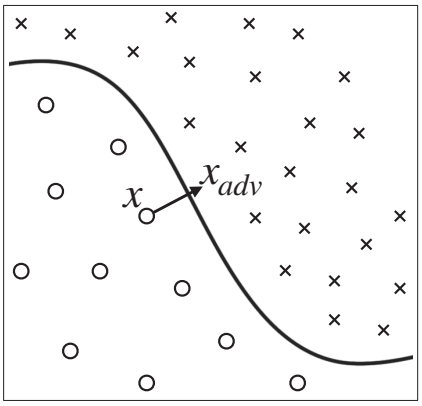


Figure 1.5 Adversarial sample crossing decision boundary

1.4.1 Adversarial Attacks Taxonomy

Adversarial attacks can be classified according to different criteria, such as knowledge, specificity, purpose, and impact. Moreover, based on the knowledge that they require about their target model f, adversarial attacks can be classified as white and black-box attacks. White-box attacks require complete knowledge of the classifier's model, such as training data, neural network architecture, learning algorithm, hyper-parameters, and learned model [Yuan *et al.* (2019)]. On the other hand, black-box attacks assume a more realistic and feasible scenario, in which the attacker has access to only the model's output [Yuan *et al.* (2019)]. Furthermore, in real-world systems, threat models might be even more restrictive. The work in [Ilyas, Engstrom, Athalye & Lin (2018)] defines three more realistic threat models: query-limited, partial-information, and label-only. The query-limited scenario considers that attackers have access to only a limited number of queries to the classifier, i.e., only a limited number of model's outputs may be accessed. The partial-information scenario considers that attackers have access to only the probabilities of some of the model's classes or scores that do not sum to one. Finally, the label-only or decision-based

scenario refers to when the attacker has access only to the model's decision, i.e., the class to which it assigns a given data sample.

According to their specificity, adversarial attacks can be classified as targeted or untargeted. The former refers to attacks that aim to induce ML models to make specific mistakes. In a classification problem, for example, targeted adversarial attacks want classifiers to assign data samples to a particular wrong class. On the other hand, untargeted attacks are only concerned with inducing wrong results, e.g., they do not care to what class classifiers assign data samples as long as it is not the correct one.

According to their purpose, the two main categories in which adversarial attacks can be classified are evasion and poisoning. Evasion attacks craft and introduce perturbations to data samples during inference time, i.e., the target of the attack is to tamper with data that is being sent to the ML model. On the other hand, poisoning attacks aim to craft and introduce adversarial perturbations to data samples that are used for training the ML model. Their goal is to compromise the model during training so that it produces wrong results once in operation.

Finally, as adversarial attacks create security issues, they can also be classified according to their impact on the confidentiality, integrity, and availability of data. Thus, adversarial attacks compromise the confidentiality of the data when the perturbations they introduce reveal confidential information by, for example, granting unauthorized access to a system. They compromise the integrity of the data when the adversarial perturbations introduced tamper with a data sample, such as a sensor measurement or the contents of a message transmitted through a wireless network. Adversarial attacks compromise the availability of the data when they interrupt the functioning of a system, such as when adversarial perturbations cause wrong classification results on modulation classifiers, causing an interruption of wireless communications.

CHAPTER 2

INTRUSION DETECTION FOR CYBER-PHYSICAL SYSTEMS USING GENERATIVE ADVERSARIAL NETWORKS IN FOG ENVIRONMENT

Paulo Freitas de Araujo-Filho[1,2] , Georges Kaddoum[1] , Divanilson R. Campelo[2] , Aline Gondim Santos[2] , David Macêdo[2] , Cleber Zanchettin[2]

[1] Electrical Engineering Department, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Centro de Informática, Universidade Federal de Pernambuco, Av. Jorn. Aníbal Fernandes, s/n, Recife, Pernambuco, Brazil 50740-560

Article published in IEEE Internet of Things Journal, April, 2021.

2.1 Abstract

Cyber-attacks on CPSs can lead to sensing and actuation misbehavior, severe damages to physical objects, and safety risks. Machine learning algorithms have been proposed for hindering cyber-attacks on CPSs, but the absence of labeled data from novel attacks makes their detection quite challenging. In this context, GANs are a promising unsupervised approach to detect cyber-attacks by implicitly modeling the system. However, the detection of cyber-attacks on CPSs has strict latency requirements, since the attacks need to be stopped before the system is compromised. In this paper, we propose FID-GAN, a novel fog-based, unsupervised IDS for CPSs using GANs. The IDS is proposed for a fog architecture, which brings computation resources closer to the end nodes and thus contributes to meeting low-latency requirements. In order to achieve higher detection rates, the proposed architecture computes a reconstruction loss based on the reconstruction of data samples mapped to the latent space. Other works that follow a similar approach struggle with the time required to compute the reconstruction loss, which renders them impractical for latency constrained applications. We address this problem by training an Encoder that accelerates the reconstruction loss computation. Experiments show that the proposed solution achieves higher detection rates and is at least 5.5 times faster than a baseline approach in the three studied datasets.

2.2 Introduction

Cyber-physical systems integrate computing and physical processes, such that effective control is performed through computation, efficient communication, and connected sensors and actuators [Ding, Han, Xiang, Ge & Zhang (2018)]. CPSs enable remote access and control of systems, devices, and machines, and thus are essential in industrial environments, especially for Industry 4.0 [Ding *et al.* (2018); Alguliyev *et al.* (2018)]. However, the widespread adoption of CPSs introduces several security threats that may cause inaccurate sensing and actuation. Such misbehavior can lead to severe damages to the controlled physical objects and harm the people that rely on them [Alguliyev *et al.* (2018); Han *et al.* (2014)].

Intrusion detection systems, which detect intrusions that other security mechanisms were not able to prevent, work as a second line of defense and have a significant role in securing cyber-physical systems [Li *et al.* (2019)]. IDSs based on anomaly detection build a normal behavior profile and classify behaviors that do not match this normal profile as attacks [Mitchell & Chen (Apr, 2014); Zarpelao *et al.* (Apr, 2017)]. In contrast to other approaches, anomaly-detection IDSs can detect unknown attacks, which is an essential feature for CPSs. CPSs connect a wide range of devices with different computation resources, communication technologies, battery capacity, software, and operating systems. Such heterogeneity challenges the deployment of security solutions and increases the attack surfaces, making CPSs more vulnerable to new and unknown attacks [Abeshu & Chilamkurti (Feb, 2018); Papamartzivanos *et al.* (2019); Midi *et al.* (2017)]. Traditional ML algorithms were shown to identify data patterns and detect cyber-attacks successfully in IDSs. However, they were also shown to not scale effectively with large datasets and to achieve low accuracy for the detection of cyber-attacks when network nodes are significantly distributed [Abeshu & Chilamkurti (Feb, 2018); Zenati *et al.* (2018b)]. On the other hand, advances in deep learning foment new IDS mechanisms capable of handling the current's cyber-attacks, level of sophistication and complexity [Abeshu & Chilamkurti (Feb, 2018); Vigneswaran *et al.* (2018); Shone *et al.* (Feb, 2018)].

Obtaining labels for attacks can be very time consuming, challenging, and sometimes even impossible. Therefore, unsupervised learning techniques, capable of detecting cyber-attacks without a need for labels, are deemed best for this task [Choi *et al.* (Sep, 2019); Schlegl *et al.* (May, 2019); Ozgumus (2019)]. However, most existing unsupervised techniques are not able to deal with the non-linearity and inherent correlations of multivariate time series, which represent a considerable amount of real-world data, including data streams generated by sensors in CPSs [Li *et al.* (2019); Li & Wen (Jan, 2014); Goh *et al.* (2017)]. Therefore, a new unsupervised technique independent from any prior knowledge of cyber-attacks is needed to detect intrusions in CPSs.

Moreover, the detection latency, i.e., the time interval between the start or penetration of an attack and its detection, is a critical challenge in the detection of attacks [Mitchell & Chen (Apr, 2014)], as they need to be detected quickly enough to be prevented. On the one hand, many devices and sensors within CPSs have limited battery and processing resources, which complicate the deployment of sophisticated security solutions [Papamartzivanos *et al.* (2019); Midi *et al.* (2017); Mourad, Laverdiere & Debbabi (2007); Mourad, Laverdière & Debbabi (2008)]. On the other hand, if an IDS is deployed in the cloud, the network transmission load, bandwidth requirement, and latency will significantly increase, and thus intrusions might not be detected in real-time [Aazam, Zeadally & Harras (2018); Yousefpour, Ishigaki, Gour & Jue (2018)]. However, a fog-based IDS architecture is well suited to meet low-latency detection requirements, as it provides computation, storage, and networking services to end users along the thing-to-cloud continuum for a better quality of service (QoS) [An, Zhou, Lü, Lin & Yang (2018); Moati, Otrok, Mourad & Robert (2014); Hu *et al.* (2017)]. In addition, through virtualization, fog nodes could use virtual machines to achieve higher efficiency and flexibility [Li, Jin, Yuan & Zhang (2017); Wahab, Bentahar, Otrok & Mourad (2020)]. Thus, a new unsupervised cyber-attack detection system could take advantage of the fog-computing paradigm and be deployed in the fog as a virtual function.

Generative adversarial networks estimate generative models through an adversarial process simultaneously training a generative model G and a discriminative model D. While the latter

estimates the probability that a sample came from the training rather than G, the former captures the data distribution without using labels. GANs are then used for unsupervised problems to find an implicit probability distribution and model of the system, while also providing a model D to detect generated or fake samples [Goodfellow *et al.* (2014); Salimans *et al.* (2016)]. Thus, they present a promising approach to tackle the challenge of developing effective unsupervised anomaly detection methods for multivariate time series, such as network attacks with probability distributions that are challenging to estimate.

2.2.1 Related Works

Recent artificial intelligence methods have a fundamental role in many domains. The work in [Peng *et al.* (2020)] proposed a novel Visual Question Answering (VQA) model to generate candidate answers and explore their semantics to facilitate the final answer prediction. In [Lu, Zhang, Xu, Li & Shen (2020)], a novel hashing method is suggested to overcome existing deep hashing approaches challenges. The work in [Xu, Lin, Lu, Gao & Shen (2020)] proposed to integrate multimodal feature synthesis, common space learning, and knowledge transfer for zero-shot cross-modal retrieval by using Wasserstein GANs. Finally, the limitations of artificial intelligence techniques and the promising potential of unsupervised learning are presented in [Lu, Li, Chen, Kim & Serikawa (2018)].

The work in [Li, Chen, Goh & Ng (2018a)] proposed a GAN-based anomaly detection (GAN-AD) method to detect deviant behaviors as possible attacks in complex networked CPSs. LSTM-RNN are used to capture the distribution of multivariate time series of the CPSs' sensors and actuators under normal working conditions. Anomalies are detected by combining the discriminator outputs to a reconstruction loss given by the residual between the actual data and its reconstruction through the generator. Experimental results demonstrated the high detection and low false-positive rates of this scheme compared to other existing methods. The GAN-AD approach was extended in [Li *et al.* (2019)], which proposed a Multivariate Anomaly Detection with GAN (MAD-GAN) framework to detect attacks using a novel anomaly score called DR-Score. This score exploits both the discriminator and generator networks, which are LSTM-RNN networks,

by computing and combining a reconstruction loss to the discrimination loss. In contrast to the LSTM-RNN approach, the work in [Schlegl, Seeböck, Waldstein, Schmidt-Erfurth & Langs (2017)] proposed AnoGAN, a deep convolutional GAN, and a scoring scheme that also combines the discrimination and reconstruction loss to detect anomalies in medical images.

Although GAN-AD [Li *et al.* (2018a)], MAD-GAN [Li *et al.* (2019)], and AnoGAN [Schlegl *et al.* (2017)] showed satisfactory performances in detecting anomalies, they all rely on an iterative approach to find a latent z by solving an optimization problem that minimizes the difference between the generated sample and the actual data. Since this optimization problem is solved for each data sample during the detection of intrusions, this strategy might take too long and not be feasible for latency constrained applications. In the face of this challenge, a few other works proposed alternative approaches to find latent representations of data samples. The works Efficient GAN (EGAN) [Zenati, Foo, Lecouat, Manek & Chandrasekhar (2018a)] and Adversarially Learned Anomaly Detection (ALAD) [Zenati *et al.* (2018b)] use a class of GANs that simultaneously learn a third network, which maps data samples to the latent space during training. However, they cannot be used to pre-trained GAN models as that third network is limited to be trained along with the GAN. The work fast AnoGAN (f-AnoGAN) [Schlegl *et al.* (May, 2019)] proposed three different architectures for mapping images to the latent space. However, it lacked an evaluation on the time efficiency of these architectures. In addition, the works in [Zenati *et al.* (2018a,b); Schlegl *et al.* (May, 2019)] are mainly focused on images and haven't been explored for multivariate time series, such as the data streams generated by CPSs, which present significant particularities and complexity. Thus, the literature still lacks a fast method to invert the GAN generator and find latent representations of multivariate time series data samples.

A few works explore fog computing and virtualization for IoT and Industry applications. The work in [Aazam *et al.* (2018)] presented an architectural overview of Industrial IoT and Industry 4.0, and discussed how the fog can provide local processing support with acceptable latency to actuators and robots. The work in [Zhang *et al.* (2019b)] presented a novel fog-based encryption-as-a-service architecture, which was shown to significantly improve security performance and

real-time communication of substation networks. The work in [An *et al.* (2018)] presented a new lightweight IDS called sample selected extreme learning machine (SS-ELM). This IDS showed, through experimental simulations, good performance in terms of accuracy and receiver operating characteristic (ROC). However, it followed a supervised approach and required labels. In [Li *et al.* (2017)], virtualization is investigated to overcome resource constraints on sensory-level nodes and network service provisioning. A case verification and quantitative analysis showed the mitigation of delay and jitter, as well as the achievement of low-latency and high scalability.

2.2.2 Contributions

In this paper, we propose FID-GAN, a novel low-latency unsupervised intrusion detection system for cyber-physical systems that uses a GAN and is deployed in the fog. The proposed architecture models data as multivariate time series and the GAN discriminator and generator as LSTM-RNN networks to acknowledge and deal with temporal dependencies among data. While the GAN discriminator already evaluates whether a data sample is an intrusion or not, the generator is used to compute a reconstruction loss and an intrusion score. In order to improve detection rates, we investigate the individual contributions of the discrimination and reconstruction losses and take advantage of both in the detection of cyber-attacks. Moreover, we improve the architecture of [Li *et al.* (2019)] by replacing their iterative GAN generator inversion technique, required for computing the reconstruction loss, for a trained Encoder. The Encoder accelerates the reconstruction loss computation and significantly reduces the detection latency by eliminating the need for solving an optimization problem during the detection of intrusions. Besides, the architecture proposed for training the Encoder allows pre-trained GAN models, since the Encoder is trained independently and after the GAN training, and also enhances the generator by updating its parameters. Furthermore, in order to achieve an even lower detection latency, our IDS architecture takes advantage of the fog-computing paradigm. Although the cloud, being more resourceful, is used to train the neural networks of our detection solution, the IDS itself is deployed in the fog as a virtual function.

In a nutshell, the main contributions of our work are:

1. An unsupervised anomaly-based IDS for CPSs using GAN, which is capable of detecting unknown attacks and overcomes the challenge of obtaining labels.
2. Evaluation of the individual contribution of the GAN discrimination and reconstruction losses in the detection of cyber-attacks to improve the detection rates.
3. Proposal of a novel and faster method for inverting the GAN generator, which is useful for latency constrained classification and retrieval tasks.
4. Proposal of a fog-based architecture for our IDS, which enables our security solution to take advantage of the low-latency of fog nodes-based applications.

2.2.3 Organization

The remainder of this paper is organized as follows. Section 2.3 introduces our proposed architecture by describing the system model, the GAN and Encoder training procedure, and the anomaly score strategy used to detect attacks. Section 2.4 explains the conducted experiments. In Section 2.5, we present and discuss the achieved results. Finally, Section 2.6 concludes the paper and proposes possible future extensions to this work.

2.3 Proposed FID-GAN Architecture

In this section, we briefly explain how GANs work and how they can be leveraged with LSTM-RNN networks to consider temporal dependencies among data. Moreover, we describe our system architecture and the trained Encoder, which accelerates the reconstruction loss computation and makes our system suitable for latency constrained applications. Finally, we define the attack detection score used to distinguish intrusions, and present the fog architecture proposed to deploy our IDS.

2.3.1 GAN with LSTM-RNN

Generative Adversarial Networks are powerful modeling frameworks for high-dimensional data that build two competing networks: a generator G and a discriminator D. The generator network is trained to produce synthetic data examples that are similar to real data patterns by taking a

random vector z, drawn from an input distribution $P_z(z)$ in a latent Z-Space. If trained only with normal data patterns, the generator captures the hidden multivariate distribution of the training sequences and can be seen as an implicit model of the system at normal status. On the other hand, the discriminator network is trained to distinguish between the generated synthetic examples and real data patterns, and then classify data patterns in one of these two classes. The two models are trained together in a zero-sum adversarial minimax game, in which the generator tries to maximize the probability of producing outputs recognized as real, and the discriminator tries to minimize that same probability [Goodfellow *et al.* (2016); Schlegl *et al.* (May, 2019)]. Thus, they can be regarded as two agents playing a minimax game with value function $V(G, D)$ as in

$$\min_{\mathrm{G}} \max_{\mathrm{D}} V(D, G) = \mathcal{E}_{x \sim P_{\mathrm{data}}(x)}[\log D(x)] + \mathcal{E}_{z \sim P_z(z)}[1 - \log D(G(x))]. \quad (2.1)$$

The continuous measurements of CPSs' sensors and actuators produce multivariate time series data streams, which are used to monitor the system working conditions. In order to deal with these intrinsically multivariate time series data, the discriminator and the generator are constructed as LSTM-RNN networks. Such networks assume that data samples are not independent of each other and that there is a temporal dependency among them. Thus, instead of dealing with isolated data samples, sequences of data are considered and stored in memory units. In this context, computations are performed for every element of a sequence such that the computation outputs for one element of the sequence serve as input for the computation of the following element in that sequence.

2.3.2 System Architecture and Fast Mapping Encoder

The discriminator D has its weights initialized with the Xavier approach, and is trained with the Gradient Descent Optimizer to minimize the mean negative cross-entropy between its predictions and sequence labels. Its loss is thus given by

$$L_{\mathrm{D}} = \frac{1}{m}\sum_{i=1}^{m}[\log D(x_i) + \log(1 - D(G(z_i)))], \tag{2.2}$$

where m is the number of samples, $x_i\ \forall i \in \{1, ..., m\}$ are the training samples, which should be recognized as real and identified as normal by our IDS, and $z_i\ \forall i \in \{1, ..., m\}$ are samples from the latent Z-Space, such that $G(z_i)$ should be recognized as false and detected as intrusions by our IDS. On the other hand, the generator's weights are initialized with a truncated normal distribution, and it is trained with the Adam Optimizer to fool the discriminator into recognizing as many generated samples as possible as real. Its loss is given by

$$L_{\mathrm{G}} = \sum_{i=1}^{m}(1 - \log D(x_i)). \tag{2.3}$$

Although the GAN discriminator network learns to distinguish between real and synthetic data, the literature has shown that the generator can also play a fundamental role in classification tasks [Li *et al.* (2018a, 2019); Schlegl *et al.* (2017)]. Thus, our proposed architecture consists of a novel strategy to detect time series attacks with a GAN by computing an attack detection score through the combination of a discrimination loss L_D and a reconstruction loss L_R. The former corresponds to the discriminator's output, as it already indicates whether an evaluated data x_t is the result of an attack, while the latter corresponds to the residual difference between x_t and its reconstruction, i.e., the difference between an evaluated pattern and the generator's output when that pattern representation in the latent space is passed through the generator. Since the generator learns an implicit model of the system, patterns that lie far away from the patterns produced by the generator are likely the result of attacks. Thus, the reconstruction loss measures how much an evaluated pattern seems to be the result of an attack.

In order to compute L_R, it is first necessary to find the representation of a pattern x being evaluated in the latent Z-Space, i.e., the vector $z \in Z$ that, passed through the generator, provides the most similar pattern to x. Even though the GAN generator provides a mapping from the latent Z-Space to the data pattern space, it does not provide a direct mapping from the data pattern

space to the latent space. Such mapping is not trivial to achieve, as it requires the inversion of the generator, which is often a non-linear model with many layers [Creswell & Bharath (Jul, 2018)]. For this purpose, our architecture builds and trains an Encoder that maps data patterns to the latent space. In contrast to other approaches that find z by solving an optimization problem for every data pattern [Li *et al.* (2018a, 2019); Schlegl *et al.* (2017); Creswell & Bharath (Jul, 2018)], the mapping performed by our Encoder is fast and suitable for latency constrained applications, such as the detection of cyber-attacks.

Therefore, in our architecture, in addition to the GAN's discriminator and generator, we also train an Encoder E that maps data patterns x, from the data pattern space X, to representations z of those patterns in the latent space Z. Thus, E is designed to do the mapping: $E(x) : X \longmapsto Z$. The proposed Encoder, depicted in Figure 2.1, follows an autoencoder configuration and is obtained from the training of an autoencoder. The Encoder part of the autoencoder maps input data into the latent space. The Decoder part, on the other hand, corresponds to the GAN generator, which reconstructs the data from its representation in the latent space by performing the mapping: $G(z) : Z \longmapsto X$. Figure 2.2 shows the Encoder and Decoder space mappings. The purpose of the autoencoder is to ensure that x and $G(E(x))$, described in Figure 2.2, are as similar as possible. Thus, it is trained by minimizing the mean squared error (MSE) residual loss between the input data x and reconstructed data $x' = G(E(x))$ as

$$L_{\text{autoencoder}} = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} [x_i - G(E(x_i))]^2}, \tag{2.4}$$

where n is the data pattern dimension.

2.3.3 Attack Detection Score

Since both the discrimination and reconstruction losses, L_D and L_R, respectively, measure how much a data pattern seems to be the result of an attack, we define an Attack Detection Score AD_{Score} as a combination of these two values as

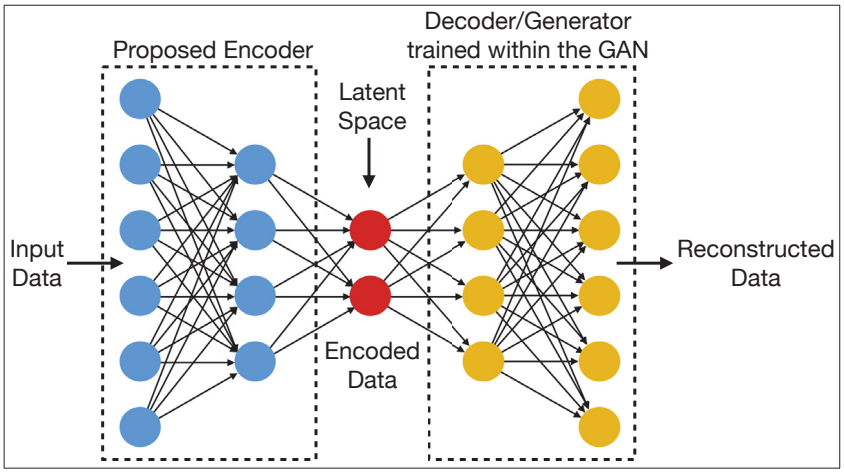


Figure 2.1 Autoencoder used to train the proposed Encoder (adapted from [Flores])

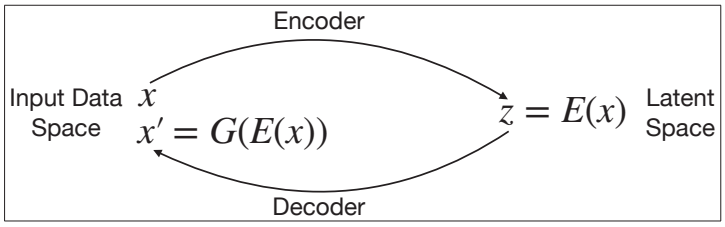


Figure 2.2 Encoder and Decoder mapping

$$AD_{\text{Score}} = \tau L_D + (1 - \tau) L_R, \tag{2.5}$$

where τ is a parameter that varies between 0 and 1 and balances the contributions of L_D and L_R in the attack detection score. Note that if τ is one, only the discrimination loss is considered for computing the anomaly detection score. In the same way, if τ is zero, only the reconstruction loss is considered. In a nutshell, the novel unsupervised strategy that we propose to detect time series attacks is described in Algorithm 2.1.

Algorithm 2.1 Novel attack detection system

1: Train the GAN D and G according to equations (2.1), (2.2), and (2.3)
2: Train the Encoder within the autoencoder using G as the Decoder and minimizing the loss function (2.4)
3: **for** Each evaluated data pattern x_t **do**
4: Compute $L_D(x_t)$
5: Obtain the latent representation of x_t by computing $E(x_t)$
6: Compute $L_R(x_t) = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} [x_i - G(E(x_i))]^2}$
7: Compute $AD_{\text{Score}} = \tau L_D + (1 - \tau) L_R, \tau \in [0, 1]$
8: Decide whether x_t is an intrusion using AD_{Score}
9: **end for**

2.3.4 Fog Architecture and System Model

The architecture of our proposed IDS is based on the fog-computing paradigm and deployed in three layers: End Point layer, Fog layer and Cloud layer. The End Point layer is where the cyber-physical systems are located. It is from this layer that the normal data patterns used to train the GAN and the Encoder come from. The unknown data patterns that are evaluated by our IDS also come from the CPSs in this layer. The Cloud layer is endowed with more computing resources and it is where the training of the GAN and the Encoder takes place. This layer tends to be distant from the CPS nodes, thus resulting in a higher latency. However, since there is no real-time requirement for the training, this is not an issue. Finally, the Fog layer is where the proposed detection system is deployed as a virtual function. Since it is closer to the CPSs in the End Point layer, a lower latency is achieved, which is suitable for the real-time requirements of attacks detection.

Figure 2.3a exhibits the architecture of the proposed IDS training model. The CPSs in the End Point layer send normal data to the cloud layer, as only normal data patterns are used for training. The GAN generator and discriminator are trained and then the Encoder is trained within the autoencoder architecture by using the trained generator as the Decoder. The architecture of the proposed IDS detection model is shown in Figure 2.3b. The CPSs in the End Point layer send unknown data patterns to be evaluated by the IDS in the Fog layer. To decide whether the

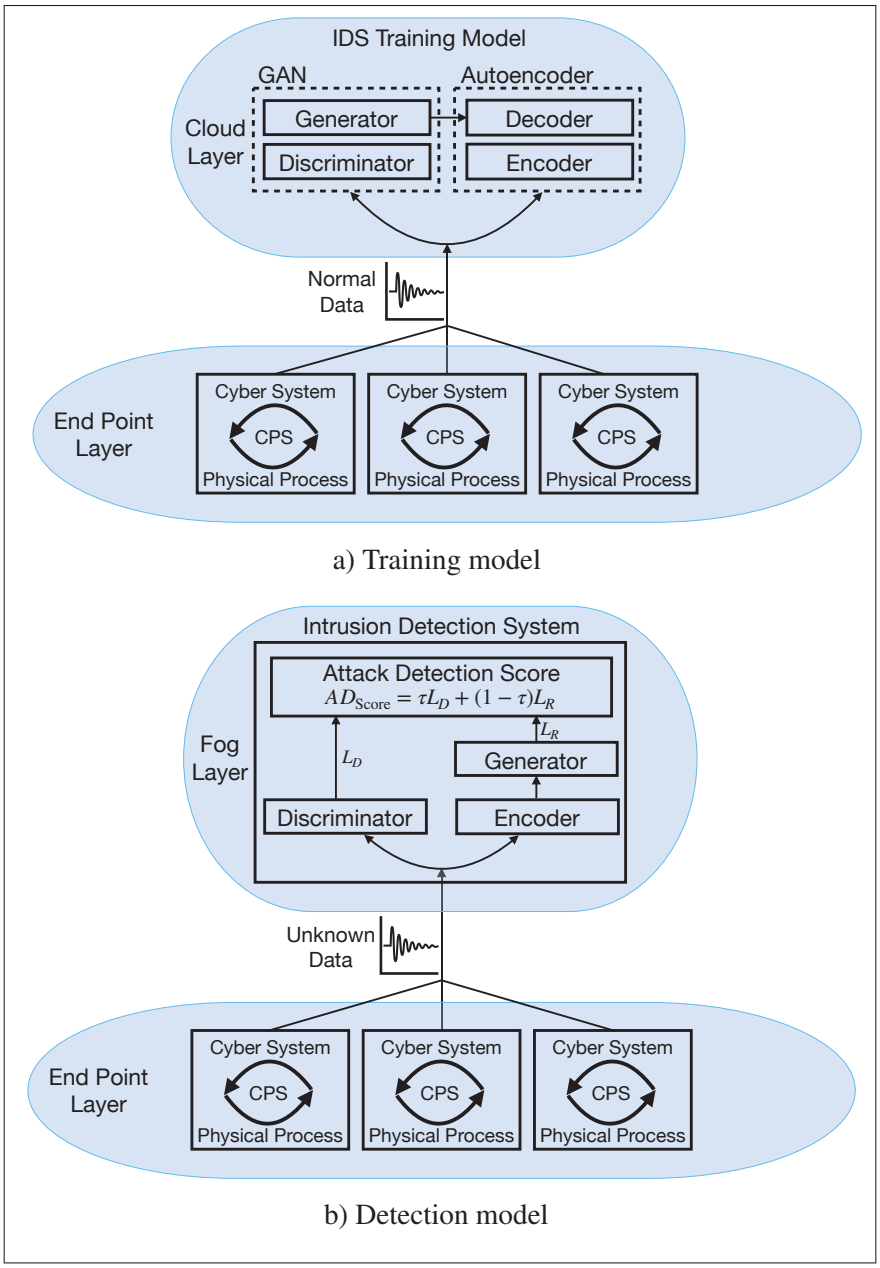


Figure 2.3 Proposed FID-GAN system model

evaluated pattern is an intrusion or not, the IDS computes the discrimination and reconstruction losses, and the attack detection score, as described in Algorithm 2.1. The total latency for this architecture is given by

$$T_{\text{Total}} = T_{\text{Comm}} + T_{\text{Net}} + T_{\text{Comp}}, \tag{2.6}$$

where T_{Comm} is the communication latency, T_{Net} is the network latency, and T_{Comp} is the computing latency. The communication latency corresponds to the propagation and transmission time of a packet, which depends on the physical medium and distance between nodes. The network latency corresponds to the network's delays, such as queuing delays caused by network congestion. These depend on the communication medium and network infrastructure. Thus, although they might impact the overall latency of the system, the communication and network latency do not affect the IDS accuracy and are therefore out of the scope of our work. On the other hand, the computing latency corresponds to the detection latency, i.e., the time taken to detect whether a data pattern is an intrusion. The Encoder trained in our architecture allows the latent representation of a data pattern to be quickly obtained, and thus the reconstruction loss computation is fast. This is an essential feature for our IDS to achieve low detection latency.

2.4 Methodology and Performance Evaluation

In this section, we briefly present the datasets used in our experiments, which contain both normal and attack data. Then, we explain the conducted experiments and metrics used for performance evaluation. Information on the platform and environment used as well as the github link for the reproduction of our experiments are given.

2.4.1 Datasets Presentation

We evaluated the proposed IDS, FID-GAN, using the Secure Water Treatment (SWaT) and the Water Distribution (WADI) datasets for CPSs, and the NSL-KDD dataset for network cyber-attacks. The SWaT dataset, built in the Singapore University of Technology and Design, represents a test-bed for a modern water treatment plant in which the water goes through a six-stage process equipped with several sensors and actuators. It contains 946,722 records, with 51 attributes of sensor and actuator data, of either normal or attack data, recorded over seven days of normal operation and four days in which 36 different attacks were conducted [iTrust Singapore University of Technology and Design (SUTD) (a)]. The WADI dataset is built by the same authors of the SWaT dataset and represents an extension of that system by considering a

complete and realistic water treatment, storage, and distribution network. It contains 1,209,610 data patterns with 126 features [iTrust Singapore University of Technology and Design (SUTD) (b)]. On the other hand, the NSL-KDD dataset is a refined version of the KDDCUP99 dataset setup by Lincoln Labs, which represents nine weeks of raw transmission control protocol (TCP) dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The LAN was operated as a true Air Force environment, affected by (1) DoS attacks, (2) unauthorized access from a remote machine (R2L) attacks, unauthorized access to local superuser with root privileges (U2R) attacks, and surveillance and other probing attacks [Canadian Institute for Cybersecurity]. For each dataset, we constructed a training, a validation, and a testing set. The former with only normal data and the other two with both normal and attack data. The training and validation sets are used to train the models and to find the optimal hyper-parameters of the algorithms, respectively. On the other hand, the testing set is used to evaluate the performance of our system in the detection of intrusions.

2.4.2 Simulation Experiments

The attack detection problem is for multivariate time series, where the temporal dependency between the data examples is considered. For this purpose, following [Li *et al.* (2019)], we assume a sliding window of size 30 across the raw data streams with shift length 10. We use LSTM networks with depth 3 and 100 hidden layers for the discriminator, Generator/Decoder, and Encoder. In addition, since [Li *et al.* (2019)] evaluated different dimensions for the latent space and found 15 to generate better samples, we also consider a latent space dimension of 15 in our study. We improve the work in [Li *et al.* (2019)] by introducing an Encoder that maps data to the latent space, and then we compare our results to that work. For a fair comparison, we follow many project decisions taken by [Li *et al.* (2019)], such as the number of hidden layers and the dimension of the latent space. However, by introducing an Encoder, our proposal is expected to improve both the detection rates and the detection latency T_{Comp}. Furthermore, we also compare our results to the work in [Zenati *et al.* (2018b)], which detects intrusions using a GAN and a third network that reconstructs data samples. The architecture in [Zenati *et al.* (2018b)] is much simpler than ours, and therefore might achieve a lower detection latency. On the other

hand, this simplicity might cause difficulties in the detection of intrusions on more complex datasets. Thus, our IDS is expected to achieve better detection rates. The detection rates are evaluated using the area under the curve (AUC) of the ROC. The detection latency is evaluated by measuring the mean computing time to detect whether a data sample is an intrusion.

Since it increases the detection latency, the reconstruction loss computation only makes sense if it improves the detection performance. Thus, we evaluate the individual contribution of the discrimination and reconstruction losses in the detection of attacks. This is done by varying the parameter τ in (2.5) from 0 to 1. If $\tau = 1$, AD_{Score} contains only the discrimination loss L_D. However, if $\tau = 0$, AD_{Score} only represents the reconstruction loss L_R. Finally, if $0 < \tau < 1$, AD_{Score} contains a combination of both the reconstruction and discrimination losses. We expect that a better detection rate can be achieved when considering a combination of both discrimination and reconstruction losses, such that the reconstruction loss computation enhances the detection results.

In our experiments, for each dataset, we trained the GAN for 100 epochs and saved the models for each epoch. Then, we consider $\tau = 1$ and compute the $AD_{\text{Score}(\tau=1)} = L_D$ for the samples within the validation set considering the 100 trained models saved. We save the model that achieves the highest AUC, considering only the discrimination loss. The generator's parameters of this model are then used to initialize the decoder part of the autoencoder. Following this, the autoencoder is trained for 300 epochs with the training set. Each trained model is saved and then used to compute the AUC for the validation set, considering only the reconstruction loss, i.e., with $\tau = 0$ and $AD_{\text{Score}} = L_R$. The autoencoder model that achieves the highest AUC is then saved. The discriminator of the first saved model and the encoder and generator of the second saved model are then used on the testing set to obtain the detection results considering only the discrimination loss, only the reconstruction loss, and a combination of both. All experiments were conducted on an AMD Ryzen Threadripper 1920X 12-Core Processor 2.2GHz with 64GB of RAM and an NVIDIA GeForce RTX 2080 under the Tensorflow 2.1 environment. The code to reproduce the experiments is available at https://github.com/pfreitasaf/FIDGAN.

2.5 Results and Discussions

In our experiments, we evaluated both the detection rate and latency using equation (2.5) for models with different contributions of discrimination and reconstruction losses. Specifically, we consider

1. $AD_{score(\tau=1)} = L_D$, which uses only the discrimination loss;
2. $AD_{score(\tau=0)} = L_R$, which uses only the reconstruction loss;
3. $AD_{Score(0<\tau<1)} = \tau L_D + (1-\tau)L_R$, which uses a combination of the discrimination and reconstruction losses.

The obtained results are compared to the results of the works in MAD-GAN [Li *et al.* (2019)] and ALAD [Zenati *et al.* (2018b)], which also compute discrimination and reconstruction losses to detect intrusions using a GAN.

2.5.1 Detection Rates

We use the AUC as the performance metric to evaluate the detection of intrusions and compare our results with [Li *et al.* (2019)] and [Zenati *et al.* (2018b)]. Thus, we obtain the ROC curves for the detection results of the data samples in the testing sets of the three considered datasets. Different contributions for the discriminant and reconstruction losses are investigated, and the model that achieves the highest AUC is considered the best one. Figures 2.4a, 2.4b, and 2.4c show the ROC curves obtained by our IDS for the SWaT, WADI, and NSL-KDD datasets, respectively. In the same way, Figures 2.5a, 2.5b, and 2.5c depict the ROC curves of the IDS proposed by MAD-GAN, and Figures 2.6a, 2.6b, and 2.6c exhibit the ROC curves of the IDS proposed by ALAD. In contrast to our IDS and MAD-GAN's IDS, ALAD's IDS explores anomaly detection scores that considers only L_D, only L_R and a combination of L_D and L_R without relying on a parameter τ.

These ROC plots demonstrate that the proposed FID-GAN achieves higher AUCs when combining both the discrimination and reconstruction losses. Moreover, the use of only the reconstruction loss achieves better detection results than the use of only the discrimination loss for the SWaT and

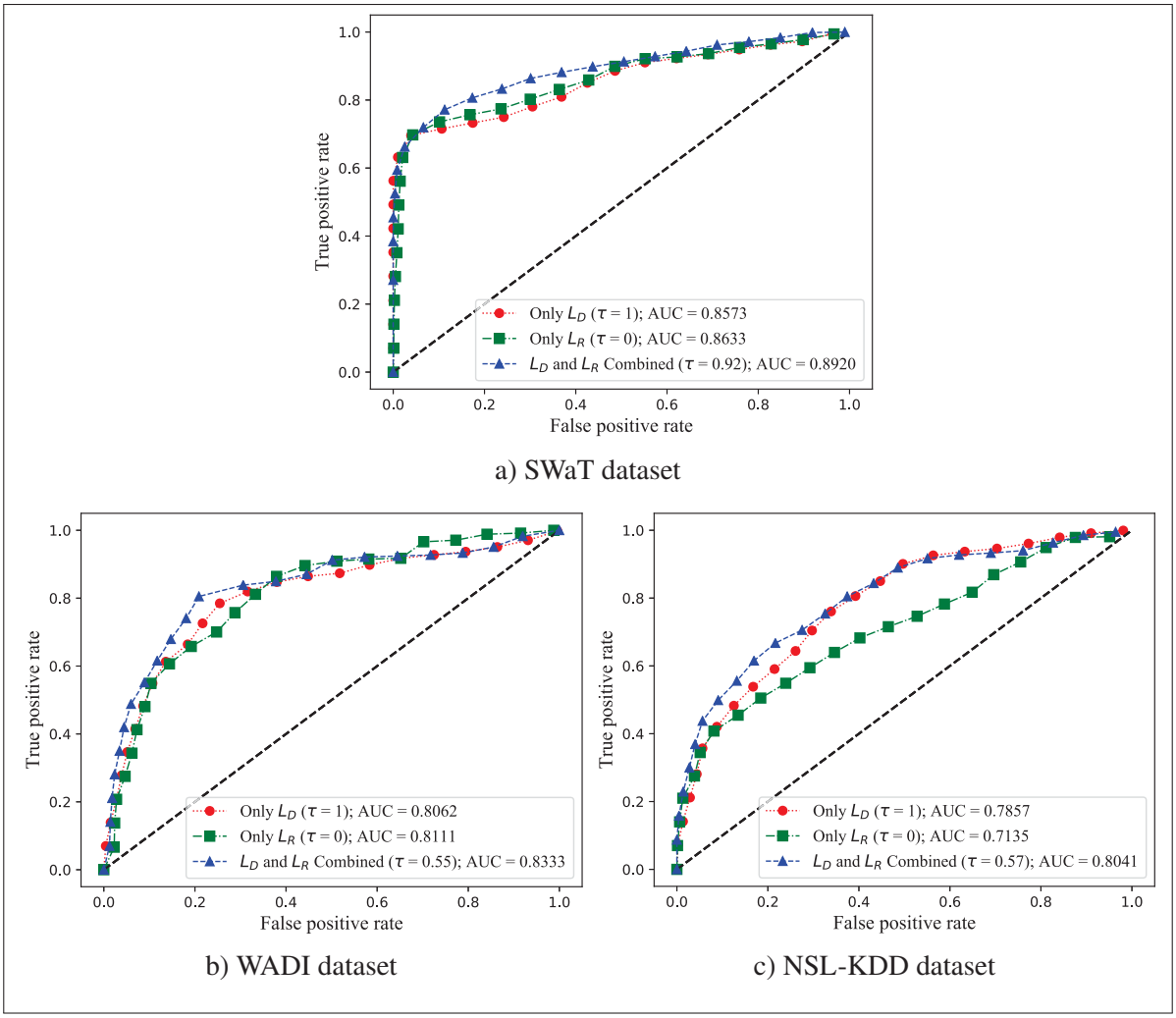


Figure 2.4 ROC curves of the proposed FID-GAN

WADI datasets. Therefore, the reconstruction loss computation is shown to enhance the detection performance of FID-GAN. In addition, the AUC results of FID-GAN are higher than the AUC results of MAD-GAN [Li *et al.* (2019)] for all considered models and datasets. Therefore, our IDS is shown to achieve better detection results than the IDS proposed by MAD-GAN. On the other hand, FID-GAN and ALAD essentially achieve the same AUCs for the SWaT dataset. However, FID-GAN achieves significantly better detection results than ALAD for the WADI and the NSL-KDD datasets, which are more complex and more challenging to detect intrusions from, since their AUCs are, in general, lower than the AUCs of the SWaT dataset. In addition, in contrast to our proposal, ALAD also does not support pre-trained GAN models, i.e., previously

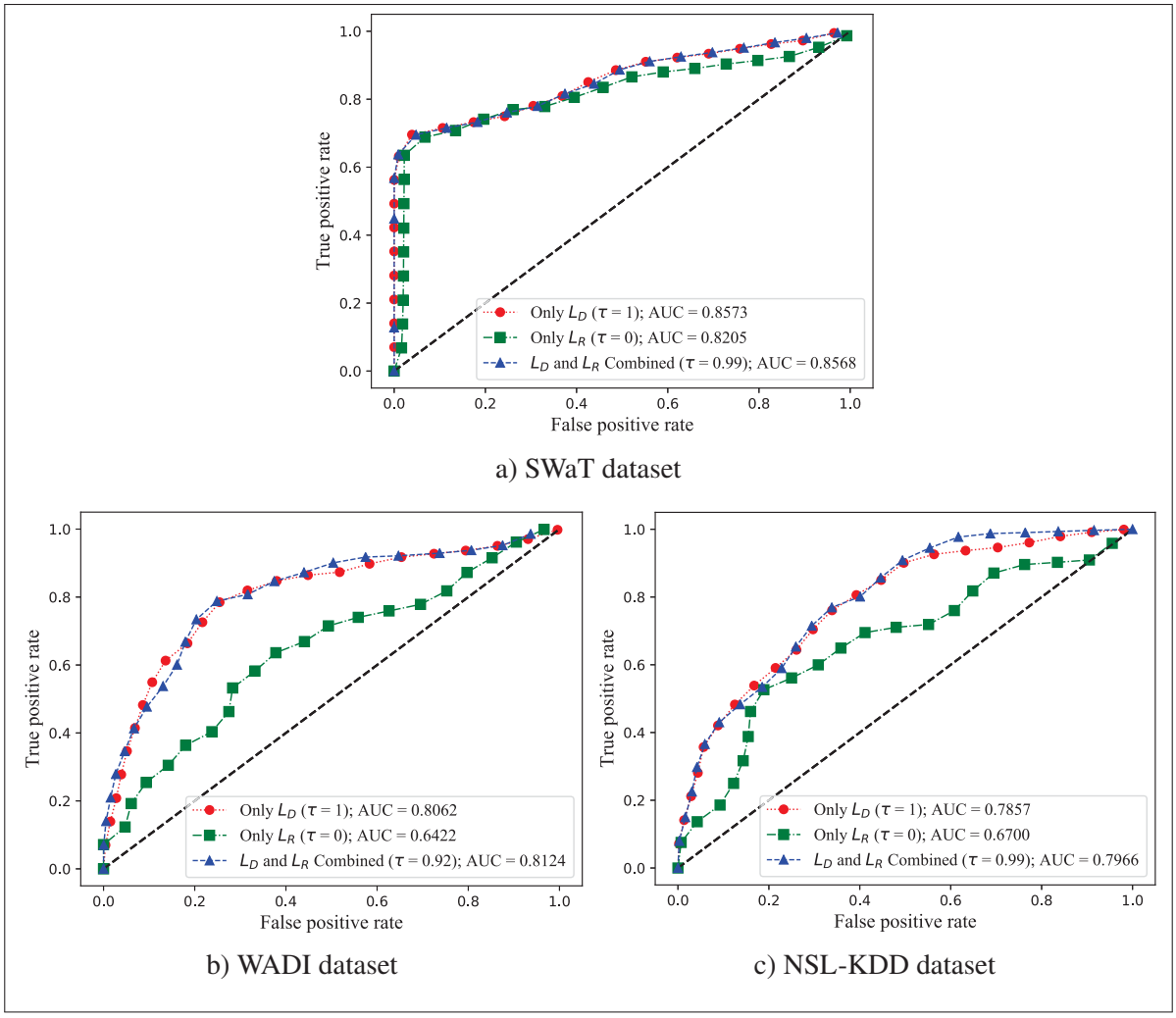


Figure 2.5 ROC curves of the IDS in MAD-GAN [Li *et al.* (2019)]

trained GANs. Precisely, the Encoder proposed by our architecture is trained independently from the GAN, and can thus be easily applied to enhance existing GAN based IDSs. On the other hand, ALAD requires their third network, which is responsible for reconstructing data samples, to be trained along with the GAN, such that previously trained GANs have to be re-trained. Since training GANs is not always an easy task due to mode collapse and stabilization issues [Arjovsky & Bottou (2017); Srivastava, Valkov, Russell, Gutmann & Sutton (2017); Salimans *et al.* (2016)], this is a disadvantage in the use of ALAD for improving existing GAN based IDSs.

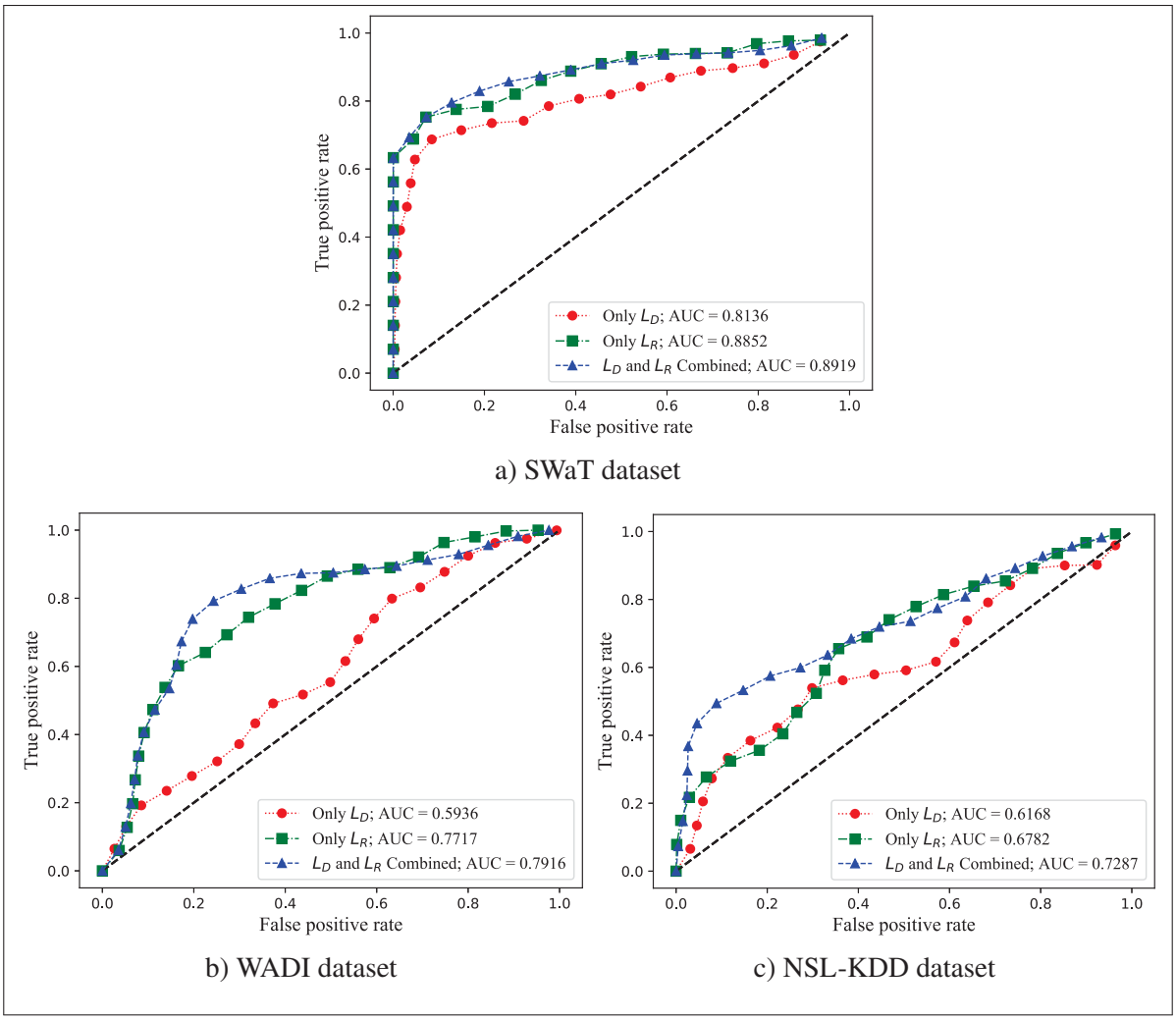


Figure 2.6 ROC curves of the IDS in ALAD [Zenati *et al.* (2018b)]

Furthermore, we also evaluate the equal error rate (EER), a performance metric derived from the ROC that represents the point where the false positive rate and the false negative rate are equal. Table 2.1 exhibits the EER values for the model that combines L_D and L_R. According to the AUCs, FID-GAN achieves lower EER than MAD-GAN for all considered datasets, and lower EER than ALAD for the two more complex datasets.

Table 2.1 Equal error rate (EER)

	SWaT	**WADI**	**NSL-KDD**
FID-GAN	0.1861	0.2049	0.2844
MAD-GAN	0.2416	0.2280	0.2921
ALAD	0.1768	0.2295	0.3485

2.5.2 Detection Latency

Since the detection of intrusions is a latency constrained application, the anomaly detection score needs to be computed in a short time. This time mainly depends on the computation time of the discrimination and reconstruction losses. Therefore, we compare the detection latency of our proposed IDS to that of the IDS in [Li *et al.* (2019)] and in [Zenati *et al.* (2018b)] when considering only the discrimination loss, only the reconstruction loss and a combination of both losses. Figures 2.7a, 2.7b, and 2.7c show the latency obtained for the SWaT, WADI, and NSL-KDD datasets, respectively.

For the three considered datasets, our IDS and the IDS in MAD-GAN achieved the same detection latency when considering only the discrimination loss. On the other hand, the latency increased when the reconstruction loss was also considered. This is because finding the latent representation of a sample and computing its reconstruction loss demands time. Although the detection latency has increased for these two IDSs, our IDS shows a much lower latency compared to that of MAD-GAN. While other works solve optimization problems during the detection of intrusions, the Encoder in our architecture enables a major reduction in the time taken to detect intrusions because it obtains the latent representation of patterns through a direct mapping. Our IDS is shown to achieve a detection latency at least 5.5 times lower than MAD-GAN's IDS when only the reconstruction loss is used. Therefore, it is much more suitable for latency constrained applications, such as the detection of intrusions in CPSs. On the other hand, the IDS proposed by ALAD achieves the shortest detection latency for the three considered datasets. In contrast to our proposed architecture, ALAD does not model data as time series or use RNN-LSTM networks to consider dependencies among data. In fact, ALAD uses neural networks with only fully-connected and convolutional layers, and therefore does not suffer

Figure 2.7 Mean detection latency

from the limited parallelization allowed by RNN-LSTM networks. Thus, it requires a lower computing time, and consequently a shorter detection latency than our solution. However, as already presented, ALAD's IDS is also the one that achieves the poorest AUCs for the WADI and NSL-KDD datasets, which indicates that it may not work well for more complex datasets and more sophisticated attacks. Thus, our IDS is more suitable than ALAD's IDS to detect intrusions in cyber-physical systems.

2.6 Conclusions

In this paper, we proposed FID-GAN, a novel unsupervised strategy to detect cyber-attacks in CPSs using a GAN. The detection is based on a combination of the discrimination and reconstruction losses, which requires the mapping of data samples to the latent space. In contrast to other works, that mapping is performed by a Encoder, such that the reconstruction loss computation is accelerated. Furthermore, to address the strict latency requirements that challenge the detection of cyber-attacks, our system is proposed within a fog architecture to benefit from the low-latency provided by fog nodes.

In our experiments, we evaluated both the detection performance and detection latency when the attack detection relied on (i) only the discrimination loss, (ii) only the reconstruction loss, and (iii) a combination of the discrimination and reconstruction losses. Three datasets were used: the SWaT and the WADI for CPSs, and the NSL-KDD for network cyber-attacks. We evaluated and compared the detection rates and latency of FID-GAN to the IDSs proposed in [Li *et al.* (2019)] and [Zenati *et al.* (2018b)]. Our proposed FID-GAN achieves significantly higher detection rates than [Zenati *et al.* (2018b)] for the WADI and NSL-KDD datasets. Moreover, our proposed solution is also shown to achieve higher detection rates and to be at least 5.5 times faster than the IDS proposed in [Li *et al.* (2019)] when considering only the reconstruction loss. Therefore, it is much more suitable for latency constrained applications, such as the detection of cyber-attacks in CPSs. In future works, we will investigate the use of Variational Autoencoders in the unsupervised detection of cyber-attacks and approaches to further reduce the detection latency of our IDS.

CHAPTER 3

UNSUPERVISED GAN-BASED INTRUSION DETECTION SYSTEM USING TEMPORAL CONVOLUTIONAL NETWORKS AND SELF-ATTENTION

Paulo Freitas de Araujo-Filho[1,2] , Mohamed Naili[3] , Georges Kaddoum[1,4] ,
Emmanuel Thepie Fapi[3] , Zhongwen Zhu[3]

[1] Electrical Engineering Department, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Centro de Informática, Universidade Federal de Pernambuco,
Av. Jorn. Aníbal Fernandes, s/n, Recife, Pernambuco, Brazil 50740-560
[3] Global Artificial Intelligence Accelerator, Ericsson Canada,
8275 Trans Canada Route, Saint-Laurent, Quebec, Canada H4S 0B6
[4] Cyber Security Systems and Applied AI Research Center, Lebanese American University

Article published in the IEEE Transactions on Network and Service Management, March, 2023.

3.1 Abstract

Fifth-generation networks provide connectivity to a massive number of devices and boost a plethora of applications in several different domains. However, the large adoption of connected devices increases attack surfaces and introduces several security threats that can severely damage physical objects and risk people's lives. Despite existing IDSs, there are still several challenges to be addressed in the detection of cyber-attacks. For instance, while unsupervised IDSs are required to detect zero-day attacks, they usually present high false positive rates. Moreover, most existing IDSs rely on LSTM networks to consider time-dependencies among data. However, LSTM networks have recently been shown to present several drawbacks and limitations, which put into question their performance on sequence modeling tasks. Thus, in this paper, we investigate GANs, a promising unsupervised approach to detecting attacks by implicitly modeling systems, and alternatives to LSTM networks to consider temporal dependencies among data. We propose a novel unsupervised GAN-based IDS that uses TCNs and self-attention to detect cyber-attacks. The proposed IDS leverages edge computing and is proposed for edge servers, which bring computation resources closer to end nodes. Experiment results show that our proposed IDS

can be configured to satisfy different detection rate and detection time requirements. Moreover, they show that our IDS is more accurate and at least 3.8 times faster than two state-of-the-art GAN-based IDSs that are used as baselines.

3.2 Introduction

The increasing growth of connected devices, such as sensors, actuators, home appliances and vehicles is changing how we interact with our surroundings. It is reducing the gap between the physical and digital worlds and integrating devices into large-scale platforms that acquire and process data to produce automated decisions while also generating knowledge and information [Rodriguez (2015); Santos *et al.* (2018)]. Smart and connected devices compose smart cities, Industry 4.0, and, in general, the IoT. They create a whole new world of services and applications, such as intelligent traffic lights, automated water treatment plants, and personal health monitoring applications [Li *et al.* (2018c); Osseiran *et al.* (2016)]. Moreover, they are expected to grow even further with the adoption of the 5G, since it can provide connectivity to a massive number of devices with highly diverse requirements [Illy *et al.* (2019); Sharma *et al.* (2011)].

On the other hand, the broadcast nature of wireless communications enables attackers to eavesdrop on the network, inject malicious data into it, and launch cyber-attacks [Ghafir *et al.* (2018)]. Therefore, the widespread adoption of IoT introduces several security threats that may impair network integrity and cause inaccurate sensing and control of systems. Such vulnerabilities could severely damage physical objects and risk people's lives [Alguliyev *et al.* (2018); Han *et al.* (2014)]. Despite numerous security solutions being available for the traditional Internet, the IoT's physical constraints and highly heterogeneous environment impose new security challenges. For instance, the heterogeneity brought by different access technologies, applications, and requirements significantly increases the attack surfaces and the threat from new types of attacks [Abeshu & Chilamkurti (Feb, 2018); Papamartzivanos *et al.* (2019); Midi *et al.* (2017)]. On the other hand, the limited battery and computing power of most IoT devices thwart the deployment of most cryptography- and authentication-based security mechanisms [Abeshu & Chilamkurti (Feb, 2018); Yang *et al.* (2017)].

To overcome these challenges, IDSs have emerged as a fundamental component to protect and secure networks and systems. They detect cyber-attacks when other security mechanisms fail [Chaabouni *et al.* (2019); Li *et al.* (2019); Jia *et al.* (2020)]. In contrast to other detection approaches, anomaly-based IDSs detect cyber-attacks by measuring deviations between data patterns and what is considered normal behavior. Although recent advances in ML foment new IDS mechanisms to detect cyber-attacks [Abeshu & Chilamkurti (Feb, 2018); Vigneswaran *et al.* (2018); Shone *et al.* (Feb, 2018)], there are still several challenges to be addressed.

First, sophisticated distributed cyber-attacks, such as modern DDoS attacks, significantly challenge current IDSs, as they might have multiple steps and be launched on different applications and devices. DDoS attacks attempt to exhaust a system's or network's resources by, for example, forcing multiple compromised devices to unnecessarily request resources so that there are no resources left for legitimate users. Google, Amazon Web Services, DNS providers, and many others have been the target of DDoS attacks. For instance, recently, a DDoS attack on a large DNS provider caused disruptions to many services, such as Airbnb, Netflix, PayPal, Visa, Amazon, The New York Times, and GitHub [Cloudfare; Nicholson]. In addition, cyber-criminals have threatened several organizations with DDoS incursions unless extortion demands are met. In 2021, such attacks disrupted internet service providers and VoIP operations worldwide [R. Dobbins and S. Bjarnason; Roland Dobbins and Steinthor Bjarnason].

Moreover, since new attacks are constantly being launched, IDSs must be able to detect zero-day attacks, for which there is no data available. Even for known attacks, it is challenging, time-consuming, and sometimes impossible to obtain labeled data. Therefore, unsupervised IDSs, which detect both known and zero-day attacks without relying on labeled attack data, are deemed the best to use [Choi *et al.* (Sep, 2019); Nisioti *et al.* (2018); Zarpelao *et al.* (Apr, 2017); Mitchell & Chen (Apr, 2014)]. However, most existing state-of-the-art unsupervised IDSs usually have high false positive rates and long detection times [Li *et al.* (2019); Nisioti *et al.* (2018); Freitas de Araujo-Filho *et al.* (2021)], which make them unsuitable for latency constrained applications.

Furthermore, most existing IDSs rely on LSTM networks to consider time dependencies among data, which are present in a considerable amount of real-world data, including network traffic. However, recent studies show that LSTM networks present several drawbacks, which put in doubt their status as the standard architecture for sequence modeling tasks [Hollis *et al.* (2018); Bai *et al.* (2018); Vaswani *et al.* (2017); Huang *et al.* (2020)]. For instance, they process data sequentially, which significantly increases their computational complexity and challenges their performance on devices with limited computational power [Duc *et al.* (2020)]. Moreover, LSTM networks can easily consume a lot of memory just to store the partial results of multiple cell gates during training [Bai *et al.* (2018)]. Therefore, it is urgently necessary to propose novel unsupervised IDSs that are capable of tackling the aforementioned challenges while avoiding the drawbacks of LSTM networks.

3.2.1 Related Works

The work in [Sayad Haghighi, Farivar & Jolfaei (2020)] proposes a learning firewall that automatically updates itself with new rules to minimize false negatives and eliminate false positives. The authors of [Jia *et al.* (2020)] propose FlowGuard, an intelligent defense mechanism that protects IoT networks against DDoS attacks. It identifies, classifies, and mitigates cyber-attacks by leveraging an edge-IoT architecture to meet the real-time requirements of IoT applications. The work in [Injadat, Moubayed & Shami (2020)] combines the Bayesian optimization-based Gaussian process (BO-GP) and the decision tree (DT) classification algorithm to detect botnet attacks on IoT devices. Similarly, the authors of [Moubayed, Injadat & Shami (2020)] rely on a genetic algorithm to optimize a random forest model that detects botnet attacks based on their DNS queries. However, the approaches proposed in [Jia *et al.* (2020)] and [Sayad Haghighi *et al.* (2020); Injadat *et al.* (2020); Moubayed *et al.* (2020)] follow a supervised learning approach so they cannot detect unknown attacks and require labeled attack data to detect known attacks.

Several other works propose unsupervised IDSs that leverage GANs to detect both known and unknown attacks without requiring labeled attack data. The works on generative adversarial

networks-based anomaly detection (GAN-AD) [Li *et al.* (2018a)] and multivariate anomaly detection with GAN (MAD-GAN) [Li *et al.* (2019)] propose GAN-based anomaly detection systems to find deviant behaviors resulting from cyber-attacks in CPSs. They detect anomalies by combining GAN discrimination and reconstruction losses. However, they detect attacks by solving an optimization problem for every data pattern under evaluation, which significantly increases detection time and makes them unsuitable for low-latency constrained applications. On the other hand, the work in [Freitas de Araujo-Filho *et al.* (2021)] proposes a low-latency unsupervised IDS for CPSs, called FID-GAN, that also uses GANs. It enhances MAD-GAN's architecture by training an encoder such that no optimization problem is solved at detection time and attacks are detected much faster than with MAD-GAN. However, it still presents considerable false positive rates.

Despite their interesting proposals, the works on FlowGuard [Jia *et al.* (2020)], GAN-AD [Li *et al.* (2018a)], MAD-GAN [Li *et al.* (2019)], and FID-GAN [Freitas de Araujo-Filho *et al.* (2021)] rely on LSTM networks to consider time dependencies among data. LSTM networks are heavily used by existing IDS solutions, which then result in several drawbacks [Duc *et al.* (2020); Bai *et al.* (2018); Vaswani *et al.* (2017)]. In contrast, the work on adversarially learned anomaly detection (ALAD) [Zenati *et al.* (2018b)] proposes a GAN-based anomaly detection system that uses only fully-connected and convolutional neural networks. However, it is significantly worse than FID-GAN at detecting attacks [Freitas de Araujo-Filho *et al.* (2021)].

To avoid LSTM's drawbacks, recent works have been proposing alternative architectures for considering time dependencies among data. The work in [Bai *et al.* (2018)] proposes TCNs by leveraging causal and dilated convolutions, and shows that TCNs can outperform LSTM networks in several sequence modeling tasks. The work in [Vaswani *et al.* (2017)] proposes transformers by replacing recurrent networks for attention mechanisms in sequence transduction models. The authors of [Huang *et al.* (2020)] propose an anomaly detection system for logs that uses transformers and show that transformers outperform LSTM networks in log sequences modeling. Finally, the work in [Tan, Iacovazzi, Cheung & Elovici (2019)] proposes an IDS that uses attention mechanisms adapted from the transformer's architecture and is more accurate

than an LSTM-based model. However, it follows a supervised learning strategy and cannot detect zero-day attacks.

3.2.2 Contributions

In this paper, we propose a novel unsupervised IDS that uses a GAN to detect both known and zero-day attacks. GANs simultaneously train two competing neural networks, namely, a generator and a discriminator. The generator learns the probabilistic distribution of a training set so that it can produce data similar to the training data. The discriminator, on the other hand, learns how to distinguish between real data and data produced by the generator. Thus, if the training set contains only normal data, the discriminator learns how to distinguish between normal data and anomalies regardless of whether they represent known or unknown attacks. Moreover, in contrast to most state-of-the-art unsupervised IDSs, which have high false positive rates and long detection times [Freitas de Araujo-Filho *et al.* (2021); Li *et al.* (2018a, 2019)], our proposed IDS does not rely on LSTM networks. Instead, it uses TCNs and self-attention in the GAN generator and discriminator networks. TCNs and self-attention enable more computation parallelization, have a constant number of sequentially executed operations, and have been shown to yield more accurate results than LSTM networks in specific sequence modeling tasks [Bai *et al.* (2018); Duc *et al.* (2020); Vaswani *et al.* (2017); Li *et al.* (2021)]. We conduct a comparative evaluation of different TCN and self-attention GAN architectures so that different trade-offs between detection rates and detection times are achieved and our IDS can be configured to satisfy different requirements. Furthermore, to achieve efficient service delivery with reduced end-to-end latency, our proposed system leverages edge computing by being deployed on edge servers closer to the network nodes under surveillance. In summary, the main contributions of our proposed TCN/self-attention GAN-based IDS are:

1. An unsupervised GAN-based IDS that is capable of detecting both known and zero-day attacks without relying on labeled attack data, which is difficult and sometimes impossible to obtain.

2. Experiments using TCNs and self-attention in a GAN to detect cyber-attacks with better detection results than existing GAN-based IDSs.
3. An evaluation of the trade-off between detection rates and detection times for different TCN and self-attention GAN architectures so that our proposed IDS can be configured to satisfy different requirements.

3.2.3 Organization

The remainder of this paper is organized as follows. Section 3.3 describes the DDoS threat scenario considered in our work. Section 3.4 presents our proposed architecture by describing the system model and the TCN and self-attention GAN architectures. Section 3.5 explains the experiments that were conducted. In Section 3.6, we present and discuss the results. Finally, Section 3.7 concludes the paper and proposes possible future extensions to this work.

3.3 DDoS Threat Scenario

While the goal of denial of service attacks is to prevent legitimate users from accessing specific network services and resources, they can achieve their goal by following different strategies: protocol exploration, network flooding, reflection amplification, and slow request/response. In our work, we consider DDoS attack types of all aforementioned strategies so that the adversaries' capabilities are described as follows.

Protocol exploration attacks rely on protocol features and implementation bugs, such as the three-way handshake mechanism of the TCP. An adversary can leverage this mechanism and send a large number of SYN messages to a server without transmitting ACK messages to acknowledge the server's responses. Thus, since the server persistently waits for the non-transmitted ACK messages, its limited buffer queue is exhausted and new connections cannot be processed. On the other hand, in network flooding attacks, the adversary sends many repetitive communication requests to fill the victim's buffer until the victim cannot accept new messages and legitimate requests are disrupted. Several network protocols may be used for flooding. For instance, the

adversary can send a large number of user datagram protocol (UDP) packets to random ports on the victim's host so that the victim is forced to send Internet control message protocol (ICMP) packets persistently and eventually reaches a resource-exhausted condition.

Similarly, reflection amplification attacks flood the victim by leveraging third-party servers, called reflectors, that respond to requests by transmitting large responses that significantly increase network traffic. Hence, the adversary sends many requests to reflectors by spoofing their source IP with the victim's IP so that reflectors send a large amount of traffic to the victim. Finally, slow request/response attacks exhaust a victim's resources by holding the communication channel for a long time. The adversary establishes multiple valid hypertext transfer protocol (HTTP) connections with a victim and segments legitimate HTTP packets into tiny fragments sent in each connection as slowly as possible within the maximum allowed communication time. Thus, as all victim's sockets are taken up, the victim becomes unavailable for legitimate connections.

In addition to the different strategies they can adopt, denial of service attacks become extra powerful and difficult to detect and trace back when they are launched from distributed sources with spoofed IPs. Moreover, adversaries usually take advantage of botnets, i.e., networks of computers infected by malware that can carry out commands under the attacker's control, to generate a significant amount of traffic from systems spread across the Internet. When large botnets are used, each system may only need to send out a small amount of traffic to produce enough volume to saturate the target network, making it extremely difficult to distinguish between DDoS and legitimate traffic. Therefore, DDoS attacks significantly impact network service and management while being very challenging to detect.

3.4 Proposed IDS Architecture

In this section, we briefly explain how GANs work and how they can leverage TCNs and self-attention to consider dependencies among data. Moreover, we describe the architecture of our proposed IDS and the different configurations it can adopt to achieve different trade-offs

between detection rates and detection times. Finally, we present our system's deployment architecture.

3.4.1 GAN-based IDSs

GANs are powerful frameworks for training generator and discriminator neural networks. When trained with only normal data, the generator implicitly models the system and learns how to produce data patterns similar to those of normal data. It learns to map random vectors z, drawn from a distribution $P(z)$ in a latent Z-space, to data patterns similar to those of normal data so that $x_{fake} = G(z)$. On the other hand, the discriminator learns to distinguish between real normal data patterns, x_{real}, and data patterns produced by the generator, x_{fake}. Thus, the discriminator's output, $D(x)$, indicates whether a data sample x is real or produced by the GAN generator, i.e., it measures deviations from normal behavior and hence detects cyber-attacks regardless of whether they are known or unknown.

The generator and discriminator neural networks are trained together in an adversarial process so that the generator tries to maximize the probability of producing outputs that are recognized as real and the discriminator tries to minimize that same probability. In our proposed system, we train a GAN according to the WGAN framework, in which the generator maximizes $G_{Loss} = D(G(z))$ and the discriminator minimizes $D_{Loss} = D(G(z)) - D(x)$. In contrast to the original GAN formulation, the WGAN is easier to train and does not suffer from the gradient vanishing problem [Arjovsky *et al.* (2017); Creswell *et al.* (2018)]. Figure 3.1 shows the adopted WGAN's training mechanism.

Existing GAN-based IDS solutions rely heavily on LSTM networks to consider temporal dependencies among data. However, since LSTM's sequential data processing significantly increases computational complexity and memory consumption during training, recent studies have been investigating alternative approaches for sequence modeling tasks. In our work, we investigate and propose replacing LSTM networks by TCNs and self-attention in both the GAN generator and discriminator for cyber-attack detection.

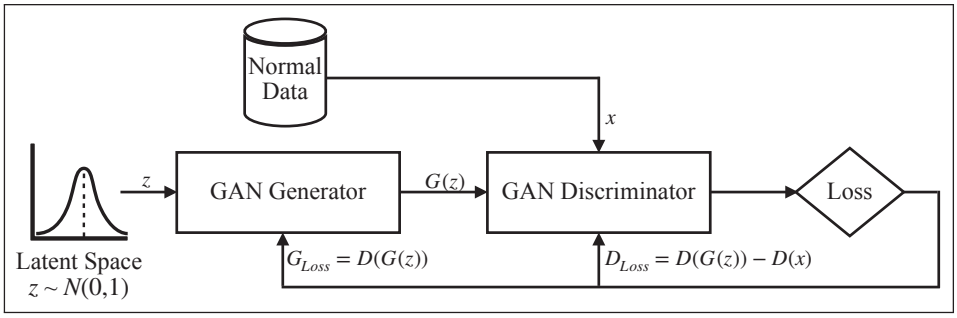


Figure 3.1 The WGAN training framework used in our proposed IDS

3.4.2 TCNs

TCNs refer to modified convolutional architectures for sequence prediction tasks. They map input sequences to output sequences of the same length and use causal convolutions, i.e., convolutions that use only information from the past. Thus, an output at time t is convolved only with elements from times earlier than t in the previous layer. In addition, since sequence modeling tasks may require more history, TCNs also use dilated convolutions to enable the architecture to look far into the past. Thus, for an input sequence $x \in \mathbb{R}^n$ and a filter $f : \{0, ..., k-1\} \rightarrow \mathbb{R}$, the dilated convolution on element s of the sequence is defined as

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) x_{s-di}, \tag{3.1}$$

where k is the filter size, d is the dilation factor, and $*_d$ is the dilated convolution operation. The dilation factor indicates how far into the past convolution operation $*_d$ looks. Hence, while $d = 1$ reduces Equation (3.1) to a regular convolution operation, the larger the dilation factor, the further back $*_d$ looks. Finally, TCN networks allow a residual connection so the architecture learns what modifications are imposed on the data rather than only modifying it. This connection contributes to avoiding the gradient vanishing problem and consists of adding the input x to the output of a series of transformations T. It is given by

$$O(x) = \Phi(x + T(x)), \tag{3.2}$$

where Φ is an activation function.

TCNs provide a powerful way to extract temporal dependencies from data and have been shown to have several advantages over LSTM networks for modeling sequences. More specifically, when our proposed solution uses TCNs, the filter f convolves across a sequence of incoming network flows by considering features only from network flows that have already occurred (causal convolution). A dilation factor is also considered (dilated convolution) so that network flows that occurred a long time ago are also taken into account. Computations can be performed in parallel since the same filter can be used in all layers, and input sequences can be processed as a whole. This means TCNs do not need to store the partial results of computations and thus consume less memory during training. Finally, TCNs have been shown to have stabler gradients, which avoids the gradient vanishing and exploding problems [Bai *et al.* (2018); Duc *et al.* (2020)].

3.4.3 Self-Attention

Attention functions are defined as the mapping of a matrix of queries Q, a matrix of keys K, and a matrix of values V to an output. Scaled dot product attention is one type of attention function, which computes a context matrix C as

$$C = Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3.3}$$

where d_k is the dimension of values. Matrices K and V usually correspond to input sequences x, whereas matrix Q is composed of randomly initialized trainable parameters. The dot product of Q and K^T gives a measure of the pairwise similarity between the query and key matrices, which results in an attention score. Thus, the C matrix represents the intrinsic dependencies between representations of a sequence.

Moreover, it has been shown that using linearly projected queries, keys, and values h times with learned linear projections contributes to extracting relationships between data [Li *et al.* (2021); Vaswani *et al.* (2017)]. Thus, MHA modules perform attention functions in parallel on each of

the projected versions of queries, keys, and values, and then concatenate their outputs as

$$MHA(Q, K, V) = W^0 Concat(head_1, ..., head_h), \tag{3.4}$$

where W^0 is a parameter matrix for the concatenation operation and $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. W_i^Q, W_i^K, and W_i^V are parameter matrices that project queries, keys, and values, respectively. Finally, self-attention considers that all the keys, values, and queries come from the same place, such as the output of the previous layer in a neural network. This allows modules to capture in-depth contextual information and relationships between data.

Similarly to TCNs, attention mechanisms make it possible to extract dependencies among data and have been shown to outperform LSTM networks in several sequence modeling tasks. They are more capable of extracting features than LSTM networks, which contributes to more accurate models [Li *et al.* (2021)]. In addition, they can process sequences as a whole and they enable more computation parallelization as MHA heads can run in parallel. Furthermore, while LSTM networks require $O(n)$ sequential operations, TCN, self-attention, and MHA layers have a constant number of sequentially executed operations. Table 3.1 summarizes the computational complexity of LSTM, TCN, self-attention, and MHA layers, where n is the sequence length, d is its depth, and k is the kernel size of convolutions [Vaswani *et al.* (2017); Kaiser (2017)].

Table 3.1 Computational complexity

Layer Type	Complexity per Layer	Sequential Operations
LSTM	$O(nd^2)$	$O(n)$
TCN	$O(knd^2)$	$O(1)$
Self-Attention	$O(n^2d)$	$O(1)$
MHA	$O(n^2d + nd^2)$	$O(1)$

3.4.4 Proposed Detection Architecture

Our proposed architecture consists of a GAN that relies on TCNs and self-attention to consider time dependencies among data. Since different applications may have different requirements and constraints, we propose different architectures for the GAN generator and discriminator neural

networks so that different trade-offs are achieved between detection rates and detection times. More specifically, we design generator and discriminator networks with one fully connected input layer, one fully connected output layer, and hidden layers of one or more TCN or self-attention blocks. Figure 3.2 shows the proposed high-level architectures of the GAN generator and discriminator.

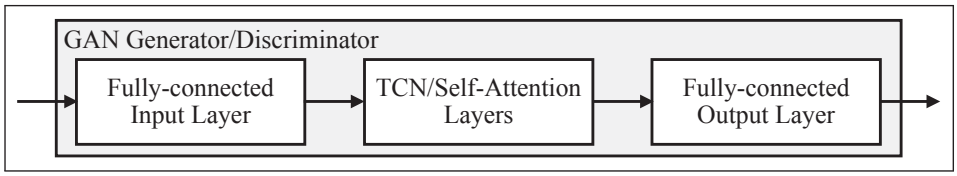


Figure 3.2 The GAN generator and discriminator architectures

The proposed TCN block allows the architecture to learn from experience and consists of a single dilated causal convolution and a rectified linear unit (ReLU) activation function. In addition, to avoid overfitting, it has a normalization layer and a dropout layer for regularization. This block can be replicated N times such that a single convolution layer is responsible for the TCN residual connection. The number of dilated causal convolutions, i.e., the value of N, directly impacts the detection rates and detection times. While higher values of N may increase our IDS's ability to learn and detect attacks, it also increases detection times, as the more layers there are, the longer the detection times. Figure 3.3 shows the TCN block architecture.

On the other hand, the proposed self-attention block consists of an MHA module that uses self-attention. Similarly to the TCN block, normalization and dropout layers are used to avoid overfitting. Moreover, a residual connection is included to help with the network's training, as it allows gradients to flow through the network. Finally, N self-attention blocks can be cascaded to increase our IDS's ability to learn and detect attacks, at the expense of also increasing detection time. Figure 3.4 shows the self-attention block architecture.

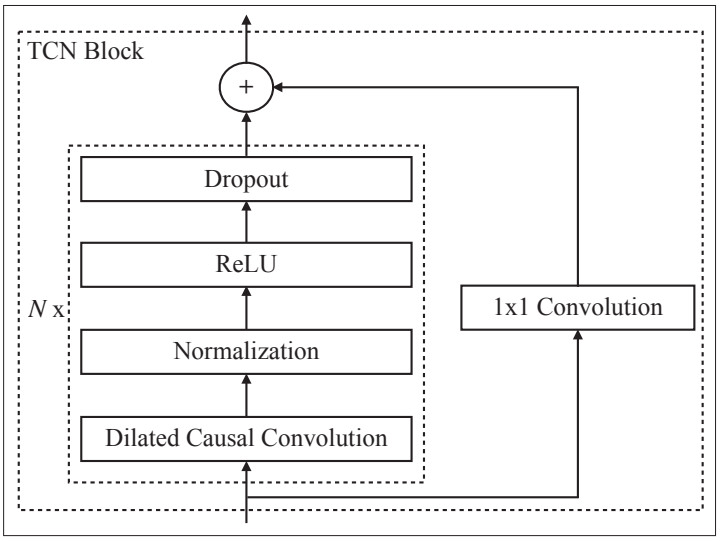


Figure 3.3 The TCN block architecture

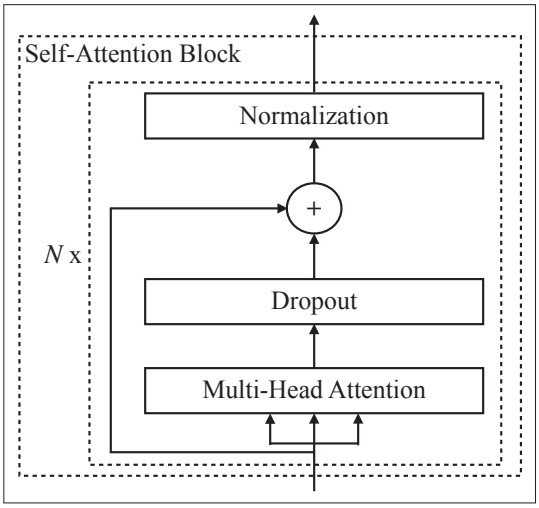


Figure 3.4 The self-attention block architecture

3.4.5 Proposed Deployment Architecture

Edge computing provides cloud computing capabilities closer to consumers and the data generated by applications. It is one of the main pillars for meeting low latency and bandwidth efficiency requirements [Kekki *et al.* (2018); Yousefpour *et al.* (2019)]. More specifically, edge computing architectures introduce edge servers, to which devices can offload computational

tasks and receive back their results in real time [Shi, Pallis & Xu (2019); Jia *et al.* (2020)]. Thus, our proposed IDS is deployed as an application on edge servers so that IoT devices can send their network flows for analysis and receive back attack detection alerts. Depending on their criticality and sensitivity, IoT devices may be configured to adopt different actions upon receiving alerts, such as dropping packets, resetting connections, or blocking the traffic from suspicious nodes.

We suggest using the open-source Kafka-ML [Martín, Langendoerfer, Zarrin, Díaz & Rubio (2022)] framework to deploy our IDS's ML models and transfer data between IoT devices and edge nodes. It uses data streams to manage ML pipelines and the Apache Kafka distributed publish/subscribe messaging system [Apache Kafka] to transfer large amounts of data with low latency. In addition, using the framework in all edge nodes and the cloud significantly reduces the ML models' response time [Carnero *et al.* (2021); Torres, Martín, Rubio & Díaz (2021)]. Finally, Kafka-ML relies on containerization and container orchestration platforms to ensure portability, easy distribution, and high availability.

Moreover, we propose that instances of our IDS that are deployed on edge servers in different regions interact with each other to exchange attack detection reports so that they become aware of attacks occurring in neighboring regions. In addition, they also send attack detection reports to a cloud service, which has a large-scale view and becomes aware of attacks that simultaneously target different locations. This awareness is essential for evaluating the potential risks and impacts of sophisticated distributed cyber-attacks to take appropriate countermeasures. Figure Figure-A I-1 in Appendix I shows the deployment of our proposed IDS on multiple edge servers.

3.5 Methodology and Performance Evaluation

In this section, we briefly present the dataset used in our experiments, which contains both normal and attack data. Then, we explain the models implemented and experiments conducted.

3.5.1 Dataset Presentation

In order to evaluate our proposed IDS, we use the CICDDoS2019 dataset provided by the Canadian Institute for Cybersecurity (CIC) and the University of New Brunswick (UNB) [for Cybersecurity; Sharafaldin, Lashkari, Hakak & Ghorbani (2019)]. This dataset contains benign traffic data and the most modern and common DDoS attacks, such as Syn, UDP, UDPLag, MSSQL, NetBIOS, LDAP, and Portmap, covering the four DDoS attack strategies presented in Section 3.3. The dataset provides 83 network flow features extracted from raw traffic data using the CICFlowMeter-V3 tool [Ahlashkari]. While many traditional network-based IDSs rely on deep packet inspection, this approach is computationally costly and challenging to implement when network traffic is encrypted [Umer, Sher & Bi (2017)]. Thus, like most state-of-the-art IDSs [Jia *et al.* (2020); Freitas de Araujo-Filho *et al.* (2021); Ozgumus (2019)], in our work, we rely on network flow features to detect malicious activities. We use the 35 most relevant network flow features from those defined in [Jia *et al.* (2020)], such as flow duration and the total number of packets in the forward and backward directions, as well as five features that identify network flows: source IP, destination IP, source port, destination port, and protocol. Table-A I-1 in Appendix 2 lists all the features used in our work.

To train and evaluate our IDS, we constructed a training, a validation, and a testing set. The training set, which is used to train the GAN, was constructed by sampling 80% of the normal network flows of a training day defined by the dataset. The validation set, which is used to optimize our models' hyper-parameters, was formed by the remaining 20% of the normal network flows of the training day in question and DDoS attacks sampled from the training day. Finally, the testing set, which is used to evaluate our IDS's performance, was constructed by sampling 50,000 normal network flows and 50,000 malicious network flows from a testing day defined by the dataset. Since our testing set contains samples collected from a different day and is only used after the training of all models has been completed, it provides unbiased results. Moreover, although the testing set's malicious network flows result from various types of DDoS attacks, our IDS relies on a binary classifier that distinguishes between normal and malicious network flows rather than classifying by attack type. Hence, the testing set is considered balanced. Table

3.2 depicts the number of normal and DDoS network flows in the constructed sets. Table 3.3 indicates the number of malicious network flows per DDoS attack type.

As shown in Tables 3.2 and 3.3, our proposed IDS relies only on normal network flows to train its neural networks. Although malicious network flows of the Syn, UDP, UDPLag, MSSQL, NetBIOS, and LDAP attacks are present in the validation set, they are used only to tune the models' hyper-parameters. Moreover, since the Portmap type of DDoS attack is present only in the testing set, it represents a zero-day attack for which no information is available.

Table 3.2 Training, validation, and testing sets

	Normal Network Flows	**DDoS Attacks Network Flows**
Training Set	45,408	0
Validation Set	11,342	68,052
Testing Set	50,000	50,000

Table 3.3 Malicious network flows per DDoS attack type

DDoS Attack Type	**Training Set**	**Validation Set**	**Testing Set**
Syn	0	11,342	8,021
UDP	0	11,342	8,021
UDPLag	0	11,342	1,873
MSSQL	0	11,342	8,021
NetBIOS	0	11,342	8,021
LDAP	0	11,342	8,021
Portmap	0	0	8,022

3.5.2 Simulation Experiments

We conducted multiple experiments by training and evaluating the GAN depicted in Figure 3.2 using different numbers of TCN and self-attention blocks, which are depicted in Figures 3.3 and 3.4, respectively. In addition, we trained the GAN using LSTM networks instead of the proposed blocks as hidden layers to have a baseline for comparing our IDS's performance. All models were optimized using the Optuna framework [Akiba, Sano, Yanase, Ohta & Koyama (2019)], which automatically searches for optimal hyper-parameter values by trial and error, and

employed the early stopping mechanism to avoid overfitting. Several hyper-parameters were tuned, such as learning rate, optimizer, batch size, kernel and dilation of convolutions, number of hidden units, and latent dimension. Moreover, we experimented with concatenating several layers of the TCN and self-attention blocks by varying the parameter N defined in Figures 3.3 and 3.4. Finally, we replicated our training experiments twenty times to reduce bias from the stochastic training. Table-A I-2 in Appendix 2 lists the hyper-parameter values used in our work.

3.6 Results and Discussions

In our experiments, we evaluated the detection rate, detection time, and complexity of our proposed IDS when using one or more TCN and self-attention blocks as hidden layers. The goal was to identify a trade-off between detection rates and detection times so that our IDS can employ different configurations and satisfy different requirements. In addition, we evaluated whether TCN and self-attention blocks outperform LSTM networks in our proposed GAN-based IDS. Finally, we compared our IDS to two state-of-the-art GAN-based IDSs: FID-GAN [Freitas de Araujo-Filho *et al.* (2021)] and ALAD [Zenati *et al.* (2018b)]. FID-GAN considers temporal dependencies among data by using LSTM networks in both the GAN generator and discriminator. ALAD, on the other hand, does not use LSTM networks or consider time dependencies among data. It relies only on fully connected and regular convolutional networks.

3.6.1 Detection Rates

We used the AUC of the ROC curve (collectively, AUCROC) as the metric to evaluate our proposed IDS's cyber-attack detection performance on the testing set samples. Each point on the curves represents both the true positive and false positive rates achieved for a threshold. Hence, the AUCROC metric allows us to evaluate our solution for many different thresholds at once rather than for only one at a time. Moreover, it shows which threshold yields the best results in terms of maximizing the true positive rate or minimizing the false positive rate.

Figure 3.5 shows the ROC curves obtained when using LSTM networks, one TCN block, two TCN blocks, and one self-attention block as the hidden layers in our proposed architecture. Other TCN and self-attention block configurations did not improve the AUCROC results but increased the detection times, as the IDS takes longer to detect attacks the more layers it has. The plots verify that our IDS achieves the highest AUCROCs results when using two TCN blocks or one self-attention block. Moreover, since the AUCROC values achieved are close to 1, our IDS ensures low false positive and false negative rates simultaneously. While minimizing false positives is essential for keeping the network operational, minimizing false negatives is essential for ensuring security.

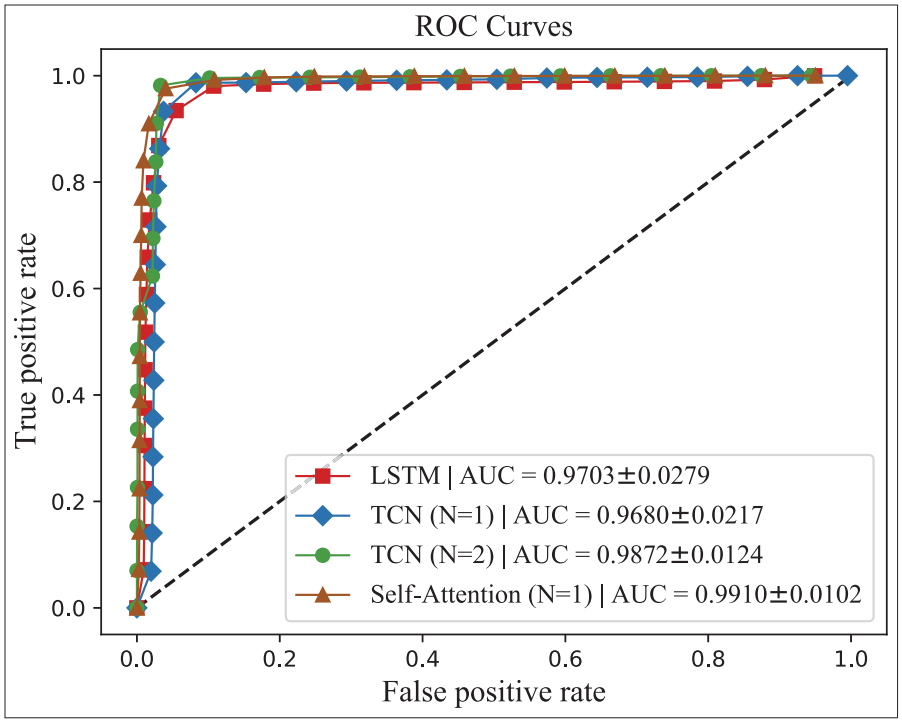


Figure 3.5 Our proposed IDS's ROC curves

Moreover, we compared the AUCROC results of our proposed IDS to those of ALAD [Zenati *et al.* (2018b)] and FID-GAN [Freitas de Araujo-Filho *et al.* (2021)]. Our proposed IDS outperformed ALAD in all its configurations, as ALAD uses only fully connected and regular convolutional layers, and does not consider dependencies among data. It also outperforms FID-GAN, which relies on LSTM networks, when it is configured with two TCN blocks or a

single self-attention block. ALAD's and FID-GAN's ROC curves are shown in Figures 3.6 and 3.7, respectively.

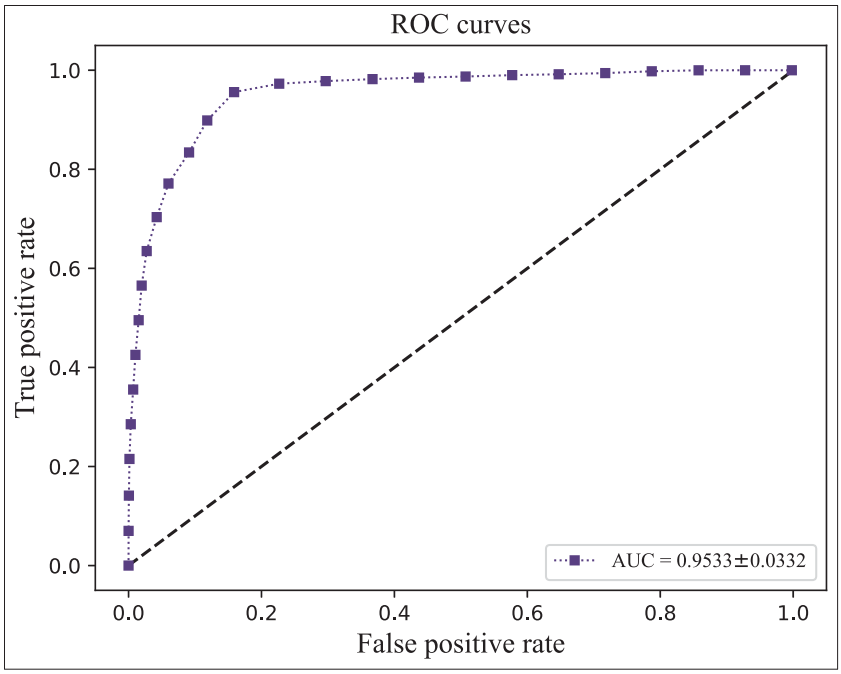


Figure 3.6 ALAD's ROC curve

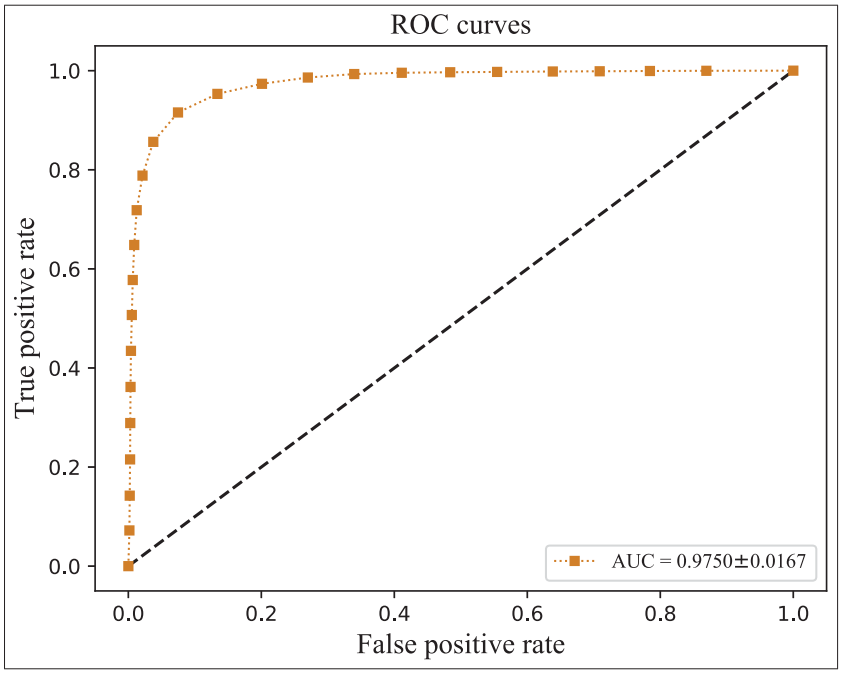


Figure 3.7 FID-GAN's ROC curve

Furthermore, we also conducted hypothesis tests to verify whether there were statistical differences between the AUCROCs of our proposed IDS, ALAD, and FID-GAN. Hence, we first conducted D'Agostino and Pearson's hypothesis test to verify whether a normal distribution could approximate the AUCROC values obtained for each model. This verification allowed us to conduct the one-way ANOVA hypothesis test to verify whether there was a significant difference between at least two of the models evaluated. The ANOVA test confirmed that at least one of the models differed from the others, such that there was a statistically significant difference among them. Since ANOVA cannot determine which model differed from the others, we conducted Tukey's honestly significant difference (HSD) post hoc test, which evaluates the models two-by-two. Table 3.4 shows the p-values of the models' comparison obtained from the post hoc test. We reject the null hypothesis that there is no significant difference between the model's AUCROCs whenever the post hoc test p-value does not exceed 0.05. Thus, our results show that when it uses self-attention or two TCN layers, our IDS is statistically different from its LSTM-based version. Therefore, self-attention and two TCN blocks successfully replace LSTM networks for attack detection and achieve better detection results.

Table 3.4 Tukey's HSD pairwise group comparisons (95.0% confidence interval)

Comparison		**p-value**
Self-Attention	TCN (N=2)	0.989
Self-Attention	TCN (N=1)	0.017
Self-Attention	LSTM	0.011
Self-Attention	FID-GAN	0.212
Self-Attention	ALAD	0.000
TCN (N=2)	TCN (N=1)	0.024
TCN (N=2)	LSTM	0.012
TCN (N=2)	FID-GAN	0.349
TCN (N=2)	ALAD	0.000
TCN (N=1)	LSTM	0.999
TCN (N=1)	FID-GAN	0.920
TCN (N=1)	ALAD	0.299
LSTM	FID-GAN	0.973
LSTM	ALAD	0.065
FID-GAN	ALAD	0.029

Finally, we evaluated our proposed IDS, ALAD, and FID-GAN at the EER, which corresponds to the threshold at which the false positive and false negative rates are equal. Table 3.5 shows the accuracy, precision, recall, and F1-scores of our IDS in four different configurations, ALAD, and FID-GAN. While our IDS outperforms ALAD and FID-GAN in all its configurations according to all the metrics used, it achieves the best results when it is configured with two TCN blocks. Finally, although our goal is not to identify different types of attacks, we provide in Table 3.6 the overall normal and attack (recall) detection rates as well as the detection rates for each type of DDoS attack in the testing set. Our IDS can detect the Portmap attack, which represents a zero-day attack, with a detection rate as high as 0.9993, which is higher than it can achieve for the other attack types. Therefore, our proposed IDS is considered able to detect unknown attacks.

Table 3.5 Accuracy, precision, recall, and F1-scores of our IDS, ALAD, and FID-GAN

	Accuracy	**Precision**	**Recall**	**F-1**
LSTM	0.9405	0.9405	0.9405	0.9405
TCN Block (N=1)	0.9588	0.9588	0.9588	0.9588
TCN Block (N=2)	**0.9707**	**0.9705**	**0.9710**	**0.9707**
Self-Attention Block (N=1)	0.9682	0.9682	0.9682	0.9682
FID-GAN	0.9203	0.9203	0.9203	0.9203
ALAD	0.8860	0.8860	0.8860	0.8860

Table 3.6 Detection rates by DDoS attack type

	Normal	**Attack**	**Syn**	**UDP**	**UDPLag**	**MSSQL**	**NetBIOS**	**LDAP**	**Portmap**
LSTM	0.9405	0.9405	0.7290	0.9994	0.9215	0.9999	0.9479	0.9996	0.9718
TCN Block (N=1)	0.9588	0.9588	**0.9728**	0.9946	0.6610	0.9868	0.935544	0.9994	0.9333
TCN Block (N=2)	**0.9704**	**0.9710**	0.8242	**1.0000**	**0.9856**	**1.000**	**0.9990**	**1.000**	**0.9993**
Self-Attention Block (N=1)	0.9682	0.9682	0.8897	**1.0000**	0.9770	**1.000**	0.9439	0.9998	0.9739
FID-GAN	0.9203	0.9203	0.6052	0.9516	0.8655	0.9898	0.9925	0.9998	0.9960
ALAD	0.8860	0.8860	0.5726	0.8193	0.8831	0.9662	0.9736	0.9999	0.9848

3.6.2 Detection Times

To evaluate how long our IDS takes to detect attacks, we measured its mean detection time when using LSTM networks, TCN blocks, and self-attention blocks. The configuration with a single

TCN block had the shortest detection time, hence it is the preferred configuration for latency constrained applications. Moreover, we compared the detection times our solution achieved to those of ALAD and FID-GAN, which took longer than our proposed solution to detect attacks. Particularly, FID-GAN has a much longer detection time than our IDS because it relies on a more complex GAN architecture and computes a reconstruction loss using an encoder neural network. Similarly, such complexity and the need for training an encoder neural network in addition to the GAN make the training time of FID-GAN much longer than that of our IDS. Therefore, our IDS is considered the best IDS of the three. Figure 3.8 shows the detection times of our proposed IDS, ALAD, and FID-GAN.

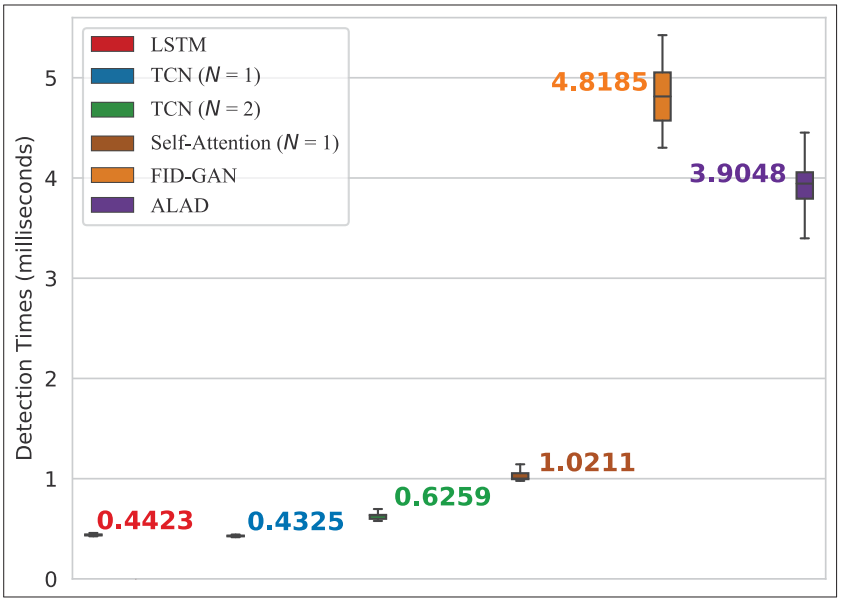


Figure 3.8 Detection times of our IDS, ALAD, and FID-GAN

Furthermore, the results in Figures 3.5 and 3.8 verify a trade-off between detection rates and detection times. For instance, our IDS achieves the highest AUCROC value and the longest detection time when configured with a single self-attention block. On the other hand, it achieves the lowest AUCROC value and the shortest detection time when configured with a single TCN block. Therefore, depending on the application's requirements and whether it is more important to achieve higher detection rates or shorter detection times, our IDS can be configured with different blocks as hidden layers and satisfy different constraints.

3.6.3 Complexity Analysis

To evaluate the complexity of our IDS, we present in Table 3.7 the number of epochs trained, the mean training time per epoch, the total convergence time, and the number of parameters in the GAN discriminator and generator. In addition, we detail in Table 3.8 the number of parameters of each model's input layer, hidden layers, and output layer. The models are trained for different numbers of epochs due to the early stopping mechanism, the LSTM model trains for the fewest number of epochs and converges fastest. However, the model with a single TCN block has the lowest number of parameters and the shortest training time per epoch, which is reflected in the detection time results in Figure 3.8, as it has the shortest detection times. Finally, even though the self-attention model converges slowest, it achieves the best detection rates.

Table 3.7 Computational complexity of each configuration of our IDS

	LSTM	**TCN (N=1)**	**TCN (N=2)**	**Self-Attention (N=1)**
Number of Epochs	33	50	45	42
Training Time (s/epoch)	12.03	10.07	13.64	19.42
Convergence Time (s)	397.00	503.29	613.80	815.80
Number of Parameters on Discriminator	4,234	3,101	4,961	5,921
Number of Parameters on Generator	3,618	2,330	8,180	5,520

Usually, the more parameters a neural network has, the longer it takes to converge. However, although a few works have proposed a correlation between the number of parameters of a neural network and its convergence time, they are not too accurate and depend on many other variables, such as the optimizer and the number and complexity of training samples [Ronen, Jacobs, Kasten & Kritchman (2019); Bach & Chizat (2021)]. For that reason, dynamic learning techniques and the early stopping mechanism have been developed and largely adopted to let the

Table 3.8 Number of parameters of each configuration of our IDS

		LSTM	TCN (N=1)	TCN (N=2)	Self-Attention (N=1)
Number of Parameters on Discriminator	**Input Layer**	1,640	1,640	1,640	1,640
	Hidden Layers	2,568	1,450	3,300	3,440
	Output Layer	26	11	21	841
Number of Parameters on Generator	**Input Layer**	840	440	4,040	440
	Hidden Layers	2,568	1,450	3,300	3,440
	Output Layer	210	440	840	1,640

models decide when they have learned enough and must stop training to reduce the training time and avoid overfitting [Prechelt (1998); Caruana, Lawrence & Giles (2000)]. Therefore, although all models evaluated were given the same computational resources and training time, so they had the same computational budget, they were allowed to stop training earlier, i.e., after different numbers of epochs using the early stopping mechanism as it is commonly done in machine learning. Precisely, all the models had a maximum time of 15 minutes (900 seconds) to train on an AMD Ryzen Threadripper 1920X 12-core processor 2.2GHz with 64GB of RAM and an NVIDIA GeForce RTX 2080 in a Pytorch environment. On the other hand, providing less time than what is required for training the models, i.e., forcing them to stop training before convergence, would unnecessarily compromise their detection results as we adopt an offline training procedure that deploys models only after they have been trained. Moreover, even though we consider 15 minutes a very reasonable maximum training time, since the models will only work and detect malicious network flows after they have been trained, our primary concern is not the training time but the detection time.

3.6.4 Combining Protection Techniques

Despite our solution's results, security is usually constructed in layers to enhance protection against cyber-attacks. Thus, our proposed IDS may be combined with other techniques to protect against DDoS attacks. For instance, we can limit attack surface areas by not exposing applications and resources to ports and protocols from which they do not expect to receive any communication [Amazon Web Services]. Moreover, we can rely on firewalls, web application firewalls (WAFs), and traditional signature-based IDSs, which are rule-based, to reduce the burden on our proposed IDS [Praseed & Thilagam (2022)]. Finally, we can rely on scalable architectures that quickly adjust their resources to accommodate high traffic volumes and maintain availability in critical systems [Amazon Web Services].

Furthermore, since DDoS attacks create large volumes of traffic, they are usually launched from botnets, i.e., networks of computers infected by malware that can carry out commands under the attacker's control. Therefore, a fundamental aspect of protecting against DDoS attacks is thwarting botnet recruitment, which requires protection techniques, such as limiting attack surfaces, and using firewalls and botnet detection systems [Garcia, Grill, Stiborek & Zunino (2014); Sriram, Vinayakumar, Alazab & KP (2020)]. In future works, we will combine our proposed IDS with botnet detection techniques.

3.6.5 Strengths and Limitations

One of the main strengths of our proposed IDS is its ability to be more accurate and at least 3.8 times faster than the two state-of-the-art GAN-based IDSs we used as baselines. Moreover, it can use different hidden layers to satisfy different requirements depending on whether it is more important to have higher detection rates or shorter detection times. Finally, our solution follows an unsupervised approach so that it does not require labeled attack data and can detect unknown attacks, such as Portmap attacks.

On the other hand, our IDS is limited to detecting attacks, as mitigating them is outside of the scope of our work. In addition, it has not been combined with other protection mechanisms,

such botnet detection techniques, which we will investigate in future works. Finally, our IDS uses only the discriminator's output. However, the reconstruction loss, which is computed using the GAN generator, could improve detection rates at the expense of increasing the detection time as is noted in [Li *et al.* (2019); Freitas de Araujo-Filho *et al.* (2021)].

Furthermore, another limitation of our proposed IDS is its offline training procedure. Since the normal behavior of systems and networks under surveillance may change with time, our IDS needs to be retrained from time to time so that it keeps up to date. However, since large environments usually have a massive amount of network data, which may reach several gigabytes per hour, constantly retraining our IDS might be operationally challenging. In addition, acquiring training data from multiple nodes may raise privacy issues as such data may contain sensitive information that must not be shared. In the face of those limitations, in future works, we will investigate and propose an online federated training procedure for our IDS that leverages federated learning to preserve privacy while always being up to date.

3.7 Conclusion

In this paper, we propose a novel unsupervised GAN-based IDS that is capable of detecting both known and zero-day attacks without relying on labeled attack data. In contrast to most existing IDSs, which rely on LSTM networks, our proposed architecture considers dependencies among data by relying on TCNs and self-attention. In our experiments, we verify the trade-off between detection rates and detection times for different configurations of our IDS. Our solution can be configured to satisfy different requirements depending on whether it is more important to achieve higher accuracies or shorter detection times. Moreover, our simulation experiments show that our proposed IDS achieves higher AUCROC values and shorter detection times than two state-of-the-art GAN-based IDSs. Therefore, not only does our IDS achieve better detection rates than LSTM-based IDSs, it is also more suitable than them for latency constrained applications.

Finally, although variational autoencoders (VAEs) are conceptually different than GANs, they have also yielded promising results in terms of learning data representations and detecting

malicious activities [Zavrak & İskefiyeli (2020); Xu, Li, Yang & Shen (2021)]. Thus, in future works, we will investigate the use of VAEs and combinations of VAEs and GANs for unsupervised attack detection, and combine them with botnet detection techniques. Moreover, we will propose an online federated training procedure so that our IDS is constantly retrained and kept up to date while preserving privacy by sharing only the weights of neural networks in different nodes instead of sensitive data. Furthermore, we will construct a new DDoS dataset with more DDoS attack types compared to the existing datasets to better evaluate our IDS's generalization and performance. For instance, we will consider the detection of DoS attacks caused by adversarial attacks that compromise the functionality of machine learning models, such as the one proposed in Freitas de Araujo-Filho *et al.* (2022), which interrupts wireless communications by compromising machine learning-based modulation classifiers on wireless receivers.

CHAPTER 4

MULTI-OBJECTIVE GAN-BASED ADVERSARIAL ATTACK TECHNIQUE FOR MODULATION CLASSIFIERS

Paulo Freitas de Araujo-Filho[1,2] , Georges Kaddoum[1] , Mohamed Naili[3] ,
Emmanuel Thepie Fapi[3] , Zhongwen Zhu[3]

[1] Electrical Engineering Department, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Centro de Informática, Universidade Federal de Pernambuco,
Av. Jorn. Aníbal Fernandes, s/n, Recife, Pernambuco, Brazil 50740-560
[3] Global Artificial Intelligence Accelerator, Ericsson Canada,
8275 Trans Canada Route, Saint-Laurent, Quebec, Canada H4S 0B6

Article published in IEEE Communications Letters, July, 2022.

4.1 Abstract

Deep learning is increasingly being used for many tasks in wireless communications, such as modulation classification. However, it has been shown to be vulnerable to adversarial attacks, which introduce specially crafted imperceptible perturbations, inducing models to make mistakes. This letter proposes an input-agnostic adversarial attack technique that is based on GANs and multi-task loss. Our results show that our technique reduces the accuracy of a modulation classifier more than a jamming attack and other adversarial attack techniques. Furthermore, it generates adversarial samples at least 335 times faster than the other techniques evaluated, which raises serious concerns about using deep learning-based modulation classifiers.

4.2 Introduction

Due to its success in the most diverse fields, deep learning has been increasingly investigated and adopted in wireless communications. It has been recently used for channel encoding and decoding [Liang *et al.* (2018)], resource allocation [Sanguinetti *et al.* (2018); Sun *et al.* (2017)], and AMC [O'Shea *et al.* (2016, 2018)]. More specifically, deep learning-based modulation classifiers have been replacing traditional AMC techniques because they achieve better classification

performance without requiring manual feature engineering [Flowers, Buehrer & Headley (2020); Lin *et al.* (2021); Sahay, Brinton & Love (2022)].

However, deep learning models have been shown to be vulnerable to adversarial attacks, which puts into question the security and reliability of wireless communication systems that rely on such models [Lin, Zhao, Tu, Mao & Dou (2020); Manoj, Sadeghi & Larsson (2021); Sadeghi & Larsson (2019); Ibitoye *et al.* (2019); Flowers *et al.* (2020)]. Adversarial attacks introduce specially crafted imperceptible perturbations that cause wrong classification results. Thus, they can force a deep learning-based modulation classifier on a receiver to misidentify the modulation mode used so that a signal is not correctly demodulated and the communication compromised.

Adversarial attacks can be classified as white or black-box attacks, depending on the knowledge they require from their target models. White-box attacks require a complete knowledge of the classifier's model, such as training data, architecture, learning algorithms, and hyper-parameters [Yuan *et al.* (2019)]. Black-box attacks, on the other hand, assume a more feasible scenario in which the attacker has access to only the model's output [Yuan *et al.* (2019)]. Furthermore, the authors of [Ilyas *et al.* (2018)] define three more restrictive and realistic black-box threat models: query-limited, partial-information, and decision-based. The query-limited scenario considers that attackers have access to only a limited number of the model's outputs. The partial-information scenario considers that attackers have access to only the probabilities of some of the model's classes. Finally, the decision-based scenario considers that attackers have access to only the model's decision, i.e., the class to which it assigns a given data sample.

Although existing adversarial attacks pose risks to the use of deep learning in wireless communications, they require a complete knowledge about the target model [Lin *et al.* (2021); Zhao, Lin, Gao & Yu (2020)] or take too long to craft adversarial perturbations [Brendel, Rauber & Bethge (2017); Moosavi-Dezfooli, Fawzi, Fawzi & Frossard (2017); Sadeghi & Larsson (2019)]. In this letter, we propose a novel input-agnostic decision-based adversarial attack technique that reduces the accuracy of modulation classifiers more and crafts perturbations significantly faster than

existing techniques. Our technique is necessary for assessing the risks of using deep learning-based AMC in the more realistic scenario of decision-based black-box attacks. Moreover, it can significantly contribute to developing classifiers that are robust against adversarial attacks. The main contributions of our work are as follows: First, we combine GANs [Goodfellow *et al.* (2014)] and multi-task loss [Kendall *et al.* (2018)] to generate adversarial samples, by simultaneously optimizing their ability to cause wrong classifications and not being perceived. Second, we reduce the accuracy of modulation classifiers more and craft adversarial samples in a shorter time than existing techniques while following the decision-based black-box scenario. Third, we propose an input-agnostic adversarial attack technique that does not depend on the original samples to craft perturbations. It allows adversarial perturbations to be prepared in advance, further reducing the time for executing the adversarial attack. Finally, our work verifies that modulation classifiers are at an increased risk and urgently need to be enhanced against adversarial attacks.

4.3 Related Works

Although adversarial attacks were initially explored in computer vision applications, they have recently been investigated for wireless communication applications, such as AMC. The authors of [Lin *et al.* (2021)] and [Zhao *et al.* (2020)] evaluate the robustness of a modulation classifier against four white-box adversarial attack techniques: fast gradient sign method (FGSM), projected gradient descent (PGD), basic iterative method (BIM), and momentum iterative method (MIM). The works show that the classifier's accuracy is significantly compromised. However, they do not measure the extent of the perturbation or the time it takes to craft adversarial samples. The work in [Manoj *et al.* (2021)] extends the white-box techniques FGSM, momentum iterative fast gradient sign method (MI-FGSM), and PGD to a power allocation application. It shows that adversarial attacks also pose a significant risk to regression-based applications, such as power allocation.

Several other works focus on black-box attacks, as they are more realistic for not requiring complete knowledge about the model [Yuan *et al.* (2019)]. The authors of [Brendel *et al.* (2017)]

propose a boundary attack technique that requires access to only the classifier's decision. It relies on a probabilistic distribution to iteratively craft adversarial samples and reduce their distance to the original sample. Although it compromises the accuracy of classifiers, it takes more than a minute to craft a single adversarial sample. The authors of [Moosavi-Dezfooli *et al.* (2017)] propose an iterative algorithm to produce universal perturbations and show that state-of-the-art image classification neural networks are highly vulnerable. However, it takes more than 20 seconds to craft each adversarial sample. The authors of [Sadeghi & Larsson (2019)] propose an algorithm to craft adversarial attacks that is shown to require significantly less power than conventional jamming attacks to compromise the performance of a modulation classifier. Although the algorithm reduces the craft time of adversarial perturbations, it still requires hundreds of milliseconds to craft each adversarial sample.

4.4 Adversarial Attacks Formulation

Although deep learning models may be trained with a large amount of data, it is impractical to train them to cover all possible input feature vectors. As a result, the decision boundary found by a trained model may differ from the real one. The discrepancy creates room for a trained model to make mistakes [Lin *et al.* (2021)]. Adversarial attacks craft perturbations to corrupt data samples so that they fall within that discrepancy area and are misclassified by a trained model. However, this is not a trivial task as the perturbations must be large enough to cause misclassifications but small enough to not be perceived. Therefore, given a sample x, the goal of an adversarial attacker is to find a perturbation δ and construct an adversarial sample $x_{adv} = x + \delta$ while satisfying

$$\min ||x_{adv} - x|| < \rho \tag{4.1}$$

and

$$f(x_{adv}) \neq f(x), \tag{4.2}$$

where $||\cdot||$ represents a chosen distance metric, ρ is the maximum imperceptible perturbation according to that metric, and f is the trained classifier target of the attack.

4.5 Proposed Adversarial Attack Technique

In our work, we consider that our proposed adversarial attack technique is deployed as a malicious software on software-defined wireless receivers, an essential piece of modern wireless communication and 5G/6G. Although injecting such malicious software is out of the scope of our work, it may be done by infecting software-defined radios with malware [Li *et al.* (2018b)]. The malware can send samples to the receiver's modulation classifier and has access to its decisions. It intercepts incoming signals, craft perturbations δ, add the perturbations to original samples to form adversarial samples $x_{adv} = x + \delta$, and forward adversarial samples x_{adv} to the modulation classifier. Thus, the receiver's modulation classifier f identifies the modulation mode of x as $f(x_{adv})$. Since $f(x_{adv}) \neq f(x)$, the signal is not correctly demodulated, and the communication is compromised. Figure 4.1 shows our attack model. The analog-to-digital converter (ADC) forwards clean samples to the modulation classifier, but they are tampered by the adversarial attacker.

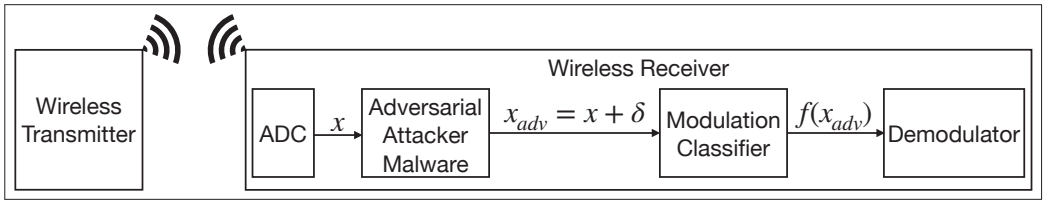


Figure 4.1 Our attack model considers the adversarial attacker as malicious software on the wireless receiver

We propose a novel multi-objective adversarial attack technique by combining a GAN and multi-task loss. GANs estimate generative models by simultaneously training two competing neural networks: generator and discriminator [Freitas de Araujo-Filho *et al.* (2021)]. The generator learns the probabilistic distribution of training data, and the discriminator learns how to distinguish between real data and data produced by the generator. We train a GAN so that its generator produces adversarial perturbations $\delta = G(z)$ from random latent vectors z and its discriminator learns to distinguish between clean samples x and adversarial samples $x_{adv} = x + G(z)$. We adopt the WGAN, which minimizes the Wasserstein distance between two probability distributions. It is easier to train than the original GAN, and does not suffer from the

gradient vanishing problem [Arjovsky *et al.* (2017); Creswell *et al.* (2018)]. Although other GAN formulations, such as WGAN-GP [Gulrajani *et al.* (2017)[, try to overcome WGAN's difficulty in enforcing the Lipschitz constant, the work in [Lucic, Kurach, Michalski, Gelly & Bousquet (2017)] shows that WGAN-GP does not necessarily outperform WGAN. In future work, we will evaluate our technique with other GAN formulations, such as WGAN-GP.

The WGAN discriminator estimates the Wasserstein distance by maximizing the difference between average critic score on real and fake samples. Besides, since we want the generator to produce perturbations rather than adversarial samples, fake samples are designated as $x + G(z)$ instead of $G(z)$. Thus, we minimize the discriminator loss given by $L_D = D(x + G(z)) - D(x)$. On the other hand, the WGAN generator has the opposite goal of maximizing the average critic score on fake samples. Hence, we minimize the generator loss given by $L_G = -D(x + G(z))$. However, such a L_G only accounts for minimizing the difference between x and x_{adv}, which corresponds to the condition of equation (4.1). It does not consider the condition of equation (4.2), which is to ensure that x and x_{adv} are assigned to different classes.

To ensure that our GAN considers the conditions of both equation (4.1) and equation (4.2), we modify the generator's loss to simultaneously optimize two objective functions that are given by L_{G1} and L_{G2}. L_{G1} represents the task of minimizing the difference between x and x_{adv} and is given by the original generator loss, hence $L_{G1} = -D(x + G(z))$. L_{G2} represents the task of ensuring that x and x_{adv} are assigned to different classes. It is given by the cross entropy loss between the class f assigns to x_{adv} and the label of x, hence $L_{G2} = CE(f(x + G(z)), y)$, where CE stands for the cross entropy loss largely adopted in classification problems and y is the label of x. During training, our technique leverages its access to the classifier's decisions to simultaneously optimize its ability to cause wrong classifications and not being perceived.

While most works that simultaneously learn multiple tasks manually tune a weighted sum of losses, we leverage the multi-task loss proposed in [Kendall *et al.* (2018)]. That work uses aleatoric uncertainty, which is a quantity that stays constant for all input data and varies between different tasks, to simultaneously optimize any two losses by optimally balancing their

contributions as

$$L = \frac{1}{2\sigma_1^2}L_1 + \frac{1}{2\sigma_2^2}L_2 + \log \sigma_1\sigma_2, \tag{4.3}$$

where L_1 and L_2 are any two losses, and σ_1 and σ_2 are learnable weights automatically tuned when training a neural network. Thus, while we train the GAN discriminator with

$$L_D = D(x + G(z)) - D(x), \tag{4.4}$$

we combine L_{G1} and L_{G2} with equation (4.3), where $L_1 = L_{G1}$ and $L_2 = L_{G2}$, so that our generator loss becomes

$$L_G = \frac{-D(x + G(z))}{2\sigma_1^2} + \frac{CE(f(x + G(z)), y)}{2\sigma_2^2} + \log \sigma_1\sigma_2. \tag{4.5}$$

Figure 4.2 shows the training model, and Algorithm 4.1 shows the execution steps of our proposed adversarial attack technique.

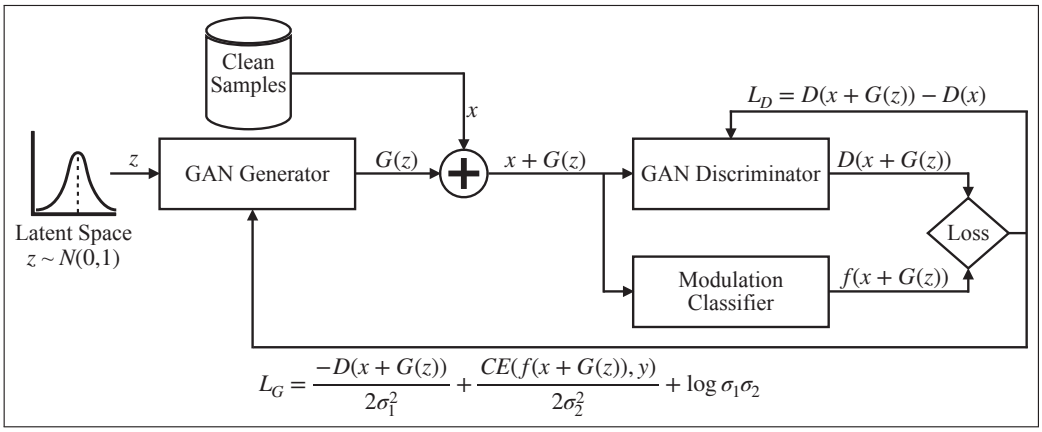


Figure 4.2 Our proposed training model

Algorithm 4.1 Proposed adversarial attack technique

1: Train a GAN according to equations (4.4) and (4.5)
2: **for** Each incoming sample x **do**
3: Compute $G(z)$
4: Construct the adversarial sample $x_{adv} = x + G(z)$
5: **end for**

4.6 Methodology and Experimental Evaluation

We use the RADIOML 2016.10A dataset and VT-CNN2 modulation classifier designed by DeepSiG and publicly available in [O'Shea *et al.* (2016); O'Shea & West (2016)] to evaluate our proposed adversarial attack technique. The dataset is constructed by modulating and exposing signals to an additive white Gaussian noise (AWGN) channel that includes sampling rate offset, random process of center frequency offset, multipath, and fading effects, as described in [O'Shea *et al.* (2016); O'Shea & West (2016)]. Since our technique crafts adversarial samples on receivers, it is not subject to channel effects. In future work, we will consider them to enhance our proposed technique so that it sends adversarial samples over the air.

After modulation and channel modeling, the signals are normalized and packaged into 220,000 samples of in-phase and quadrature components with length 128, each associated with a modulation scheme and a signal-to-noise ratio (SNR). SNR is a measure of a signal's strength. It is the ratio between the power of the signal and of the background noise, i.e., $SNR_{[dB]} = 10\log(\frac{P_{signal}}{P_{noise}})$, where P is the signal power. Eleven different modulation schemes (eight digital and three analog) are possible: 8PSK, BPSK, QPSK, QAM16, QAM64, CPFSK, GFSK, PAM4, WBFM, AM-DSB, and AM-SSB. Twenty different SNRs, ranging from -20 dB to 18 dB in steps of 2 dB, are possible. Twenty percent of the samples are reserved as a testing set to measure the VT-CNN2 modulation classifier's accuracy on clean and adversarial samples.

The VT-CNN2 modulation classifier relies on deep convolutional neural networks and classifies samples among the eleven modulation schemes in the dataset. Figure 4.3 shows VT-CNN2's architecture. Although the softmax layer gives the probability of membership for each class, we consider the classifier's output to be only its final decision, i.e., the modulation class that has the highest probability. Thus, $f(x + G(z))$ is the predicted label of one of the modulation schemes considered.

Finally, Figures 4.4 and 4.5 show the GAN's generator and discriminator architectures. They were optimized using the Optuna framework [Akiba *et al.* (2019)], which automatically searches for the optimal hyper-parameters, and the early stopping mechanism to avoid overfitting. Table 4.1

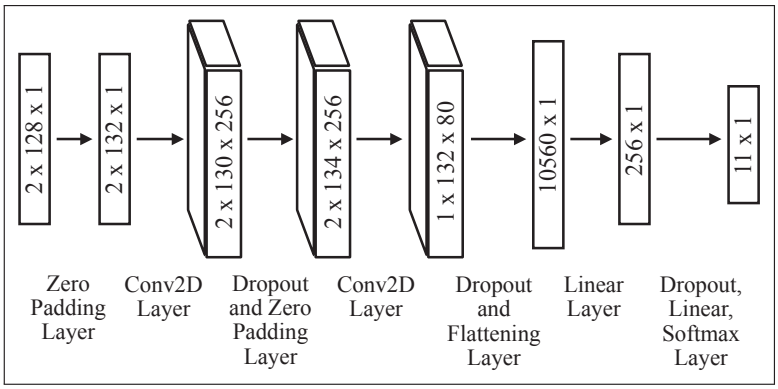


Figure 4.3 VT-CNN2 neural network architecture

shows the hyper-parameter values used in the GAN after tuning. All experiments were conducted using an AMD Ryzen Threadripper 1920X 12-core 2.2GHz processor with 64GB of RAM and an NVIDIA GeForce RTX 2080 in a Pytorch environment.

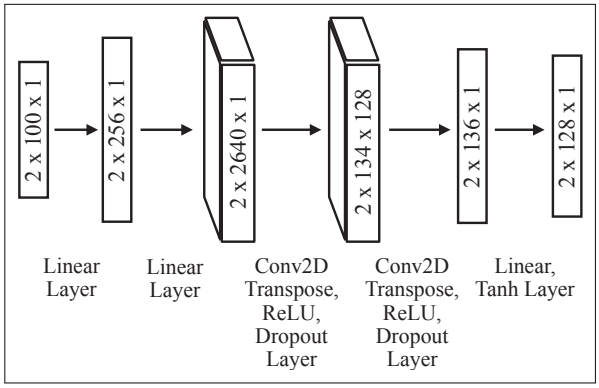


Figure 4.4 GAN generator architecture

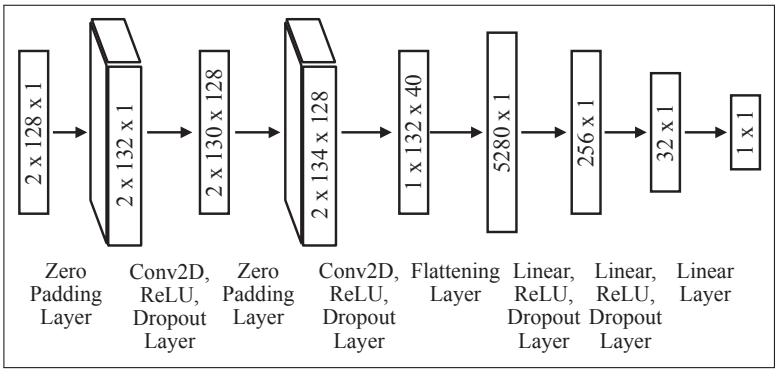


Figure 4.5 GAN discriminator architecture

Table 4.1 Hyper-parameters values

Hyper-Parameter	Value
Optimizer	Adam
Generator Learning Rate	0.00049
Discriminator Learning Rate	0.00055
Batch Size	128
Latent Dimension	100
Dropout Rate	0.10

4.7 Results and Discussion

As previously mentioned, the goal of adversarial attacks is to introduce imperceptible perturbations capable of reducing the accuracy of a modulation classifier. Therefore, we evaluated our proposed attack technique by measuring the VT-CNN2's accuracy on clean and adversarial samples, and the perturbation-to-noise ratio (PNR). PNR measures the ratio between the perturbation and noise power levels so that $PNR_{[dB]} = 10\log(\frac{P_{perturbation}}{P_{noise}})$, where P is the signal power. The larger the PNR, the larger the perturbation is in comparison to the noise, becoming more distinguishable and more likely to be detected. Perturbations are considered imperceptible when they are in the same order as or below the noise level, i.e., PNR < 0 dB.

Figure 4.6 shows the VT-CNN2's accuracy versus PNR for SNRs of 10, 0, and -10 dB. Without attacks, the classifier achieves different accuracy depending on the SNR because larger noises make it harder for the classifier to achieve correct results. Under our proposed adversarial attack, the classifier's accuracy is significantly reduced in all cases. At 0 dB PNR, our technique reduces the accuracy by 37% for 10 dB SNR, 56% for 0 dB SNR, and 7% for -10 dB SNR. Our technique reduces the accuracy more for 0 dB than for 10 dB SNR because, for signals with the same strength, larger SNRs mean lower noise levels so that it is more challenging to produce imperceptible perturbations that still significantly compromise the accuracy. However, although the noise at -10 dB SNR is the highest, allowing our technique to produce larger perturbations, the accuracy reduction is not as significant as at 0 dB SNR or 10 dB SNR. If $f(x + G(z))$ in equation (5) gives too many wrong results regardless of the adversarial perturbation $G(z)$, it is harder for our technique to find what perturbation would reduce the classifier's accuracy the

most. Thus, the fact that our technique relies on the classifier's decisions to train the GAN diminishes its capacity to produce wrong classifications when the classifier's accuracy is low. Since the classifier's accuracy is around only 22% at -10 dB SNR, the adversarial perturbations that our proposed technique crafts are less effective. Nevertheless, our proposed adversarial attack technique still significantly reduces the classifier's accuracy.

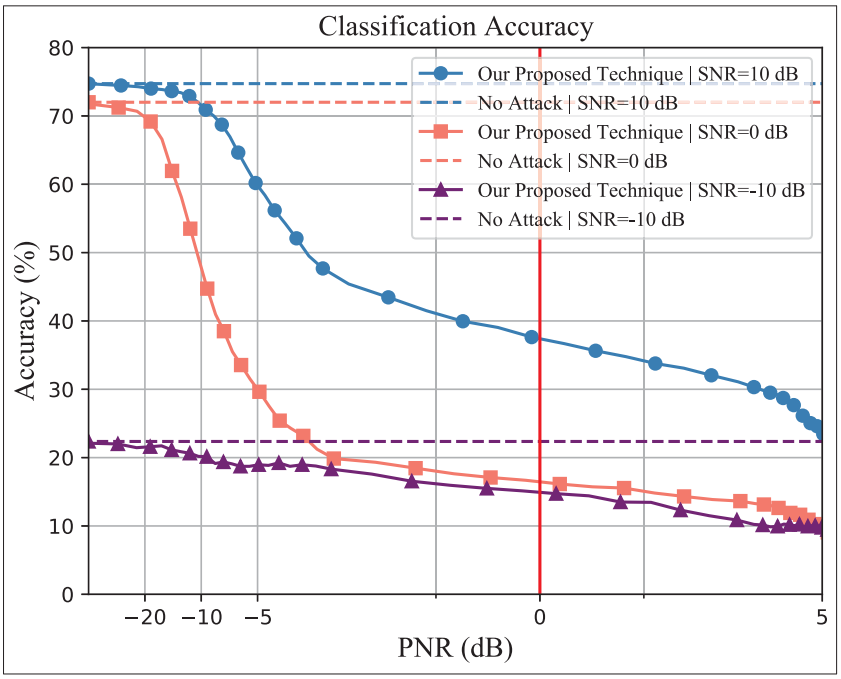


Figure 4.6 Modulation classifier's accuracy versus PNR with and without our proposed adversarial attack technique

We further examine the influence of perturbations on signal waveforms. We verify that the signal waveform after perturbation (adversarial sample) is consistent with the original waveform (clean sample), i.e., amplitude, frequency, and phase do not significantly change. Thus, while our technique's perturbations mislead the classifier, they are not easily recognized by human eyes. Figure 4.7 illustrates the time domain waveform of an 8PSK signal before and after the perturbation is introduced. Similar results were achieved for the other modulation schemes considered, such that clean and adversarial samples always have very similar waveforms without significant changes in their amplitude, frequency, and phase.

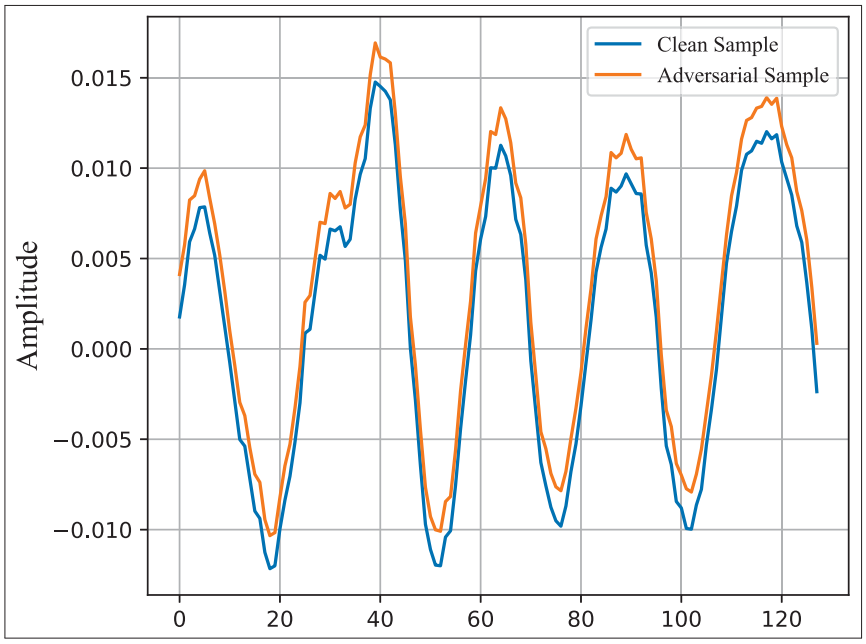


Figure 4.7 Waveform comparison of a 8PSK signal with SNR=10 dB before (clean sample) and after (adversarial sample) our proposed adversarial attack

Moreover, we compare our results to those of a jamming attack, which adds Gaussian noise to signals, and two other adversarial attack techniques: those proposed in [Moosavi-Dezfooli *et al.* (2017)] and [Sadeghi & Larsson (2019)]. Figure 4.8 shows the VT-CNN2's accuracy on clean samples and adversarial samples produced by the jamming attack and the three adversarial attack techniques evaluated for SNR=10 dB. Perturbations introduced by adversarial attacks are specially crafted to reduce the classifier's accuracy the most while not being perceived. Thus, our technique and the techniques from [Moosavi-Dezfooli *et al.* (2017)] and [Sadeghi & Larsson (2019)] are significantly more harmful than attacks that introduce random noises, such as the jamming attack. Moreover, our proposed attack technique is the one that reduces the accuracy the most.

Finally, we evaluate how long it takes for each technique to craft adversarial samples. Table 4.2 shows the mean execution time for crafting adversarial samples. Our proposed technique achieves significantly shorter times than the other two techniques by crafting adversarial samples in less than 0.7 ms. Thus, it is more than 335 times faster than the second-fastest attack technique.

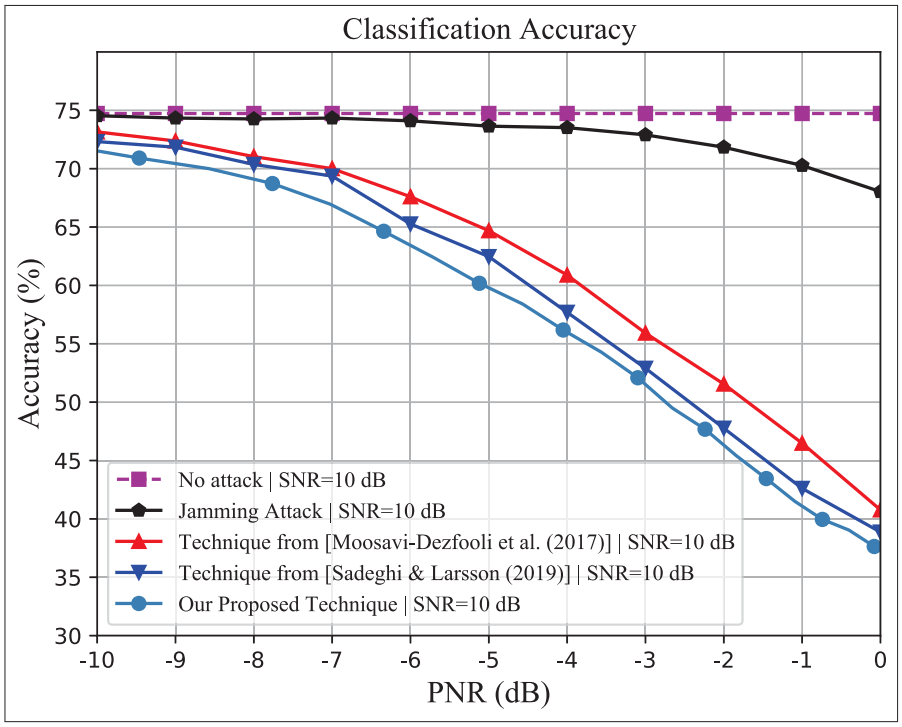


Figure 4.8 Modulation classifier's accuracy versus PNR without and subject to different adversarial attack techniques

Techniques that take too long to craft perturbations might be too late so that the signals they aim to perturb have already been correctly demodulated. Thus, such a time reduction is essential to compromise fast modulation classifiers and is a great advantage of our technique. Moreover, since our technique is input-agnostic, it can prepare perturbations in advance and just add them to incoming signals. Therefore, our proposed technique represents a severe risk to using deep learning-based modulation classifiers.

Table 4.2 Mean execution time for crafting adversarial samples

Adversarial Attack Technique	**Mean Execution Time per Sample**
Technique from [Moosavi-Dezfooli *et al.* (2017)]	20189 *ms*
Technique from [Sadeghi & Larsson (2019)]	234 *ms*
Our Proposed Technique	0.6980 *ms*

4.8 Conclusion

In this letter, we verified that deep learning is exposed to security risks that must be considered despite its advantages. Our results showed that it is possible to quickly craft small imperceptible perturbations that completely compromise modulation classifiers' accuracy and hence wireless receivers' performance. Therefore, it is urgently necessary to enhance deep learning-based modulation classifiers' robustness against adversarial attacks. As future work, we will evaluate the use of other GAN formulations, such as WGAN-GP, modify our attack model to consider adversarial attacks transmitted over the air, and investigate adversarial attack defense strategies.

CHAPTER 5

DEFENDING WIRELESS RECEIVERS AGAINST ADVERSARIAL ATTACKS ON MODULATION CLASSIFIERS

Paulo Freitas de Araujo-Filho[1,2,3] , Georges Kaddoum[1] , Mohamed Chiheb Ben Nasr[1] , Henrique F. Arcoverde[2,3] , Divanilson R. Campelo[2]

[1] Electrical Engineering Department, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Centro de Informática, Universidade Federal de Pernambuco,
Av. Jorn. Aníbal Fernandes, s/n, Recife, Pernambuco, Brazil 50740-560
[3] Tempest Security Intelligence,
Paço Alfândega Shopping - Loja 216A - 1 Piso, R. Alfândega 35, Recife, Pernambuco, Brazil 50030-030

Article submitted to the IEEE Internet of Things Journal, November, 2022.

5.1 Abstract

Deep learning has been adopted for a wide range of wireless communication tasks, including modulation classification, because of its great classification capability. However, deep learning models have been shown to also introduce risks and vulnerabilities. For instance, adversarial attacks craft and introduce imperceptible perturbations that compromise the accuracy of deep learning-based modulation classifiers on wireless receivers. Therefore, in this paper, we propose a novel wireless receiver architecture that enhances deep learning-based modulation classifiers to defend them against adversarial attacks. Our experimental results show that our defense technique significantly diminishes the accuracy reduction that is caused by adversarial attacks by protecting modulation classifiers at least 18% more than existing defense techniques.

5.2 Introduction

The recent increase in connected devices and wireless communication traffic, which has been boosted by 5G/6G technology, has made the radio spectrum overcrowded and inefficient [Sahay *et al.* (2022); Lin *et al.* (2021)]. To mitigate this issue, modern wireless transmitters dynamically change how the radio spectrum is shared by automatically switching between different modulation

schemes. As a result, wireless receivers must automatically recognize what modulation schemes are being used; otherwise, signals will not be demodulated correctly and communication will be compromised. Automatic modulation classification, which is an essential piece of cognitive and software-defined radio, has therefore become crucial to wireless communications [Flowers *et al.* (2020); Sahay *et al.* (2022); Sahay, Love & Brinton (2021)].

Deep learning has been increasingly investigated for many tasks in wireless communications, such as channel encoding and decoding [Liang *et al.* (2018)], resource allocation [Sanguinetti *et al.* (2018); Sun *et al.* (2017)], and AMC [O'Shea *et al.* (2016, 2018)]. Deep learning-based modulation classifiers, for example, have been found to perform better than traditional techniques that usually rely on statistical approaches [Sahay *et al.* (2022)]. Moreover, they do not require manual feature engineering, which significantly reduces the cost of involving an expert [Flowers *et al.* (2020); Lin *et al.* (2021); Sahay *et al.* (2022)]. Hence, deep learning has been gaining ground and is being widely adopted for AMC [O'Shea *et al.* (2016, 2018); Lin *et al.* (2021); Sahay *et al.* (2022)].

However, deep learning models have recently been shown to introduce vulnerabilities and security risks. While wireless communications' shared and broadcast nature allows attackers to tamper with signals transmitted over the air, adversarial attacks introduce small imperceptible perturbations that fool ML models into making wrong decisions. Unlike jamming attacks, which tamper signals by adding Gaussian noise, adversarial attacks craft precisely the right perturbation to compromise a classifier's accuracy the most. Hence, they are much more harmful than jamming attacks and present a severe risk to modulation classifiers that could significantly compromise wireless communications [Freitas de Araujo-Filho *et al.* (2022); Lin *et al.* (2020); Manoj *et al.* (2021); Sadeghi & Larsson (2019); Ibitoye *et al.* (2019); Flowers *et al.* (2020)].

Adversarial attacks can be classified as white- or black-box attacks depending on what knowledge they require from the target models. White-box attacks represent the worst-case scenario in which the attacker has complete knowledge about the target model, such as training data, architecture, learning algorithm, and hyper-parameters [Yuan *et al.* (2019)]. Black-box attacks, on the other

hand, are more feasible and realistic as they assume that the attacker has access to only the model's output or decision [Yuan *et al.* (2019)]. Both types of attacks have been shown to severely compromise the accuracy of modulation classifiers.

The works in [Lin *et al.* (2021)] and [Zhao *et al.* (2020)] show that four white-box adversarial attack techniques significantly compromised the accuracy of modulation classifiers: the FGSM, the PGD, the BIM, and the MIM. The work in [Sadeghi & Larsson (2019)], on the other hand, proposes a black-box adversarial attack technique that requires significantly less power than other attacks techniques to compromise the performance of a modulation classifier. Finally, the work in [Freitas de Araujo-Filho *et al.* (2022)] combines GANs and multi-task loss to generate adversarial samples that can simultaneously optimize their ability to cause wrong classifications and not be perceived. That technique reduces the accuracy of a modulation classifier more and crafts adversarial samples much faster than other adversarial attack techniques. It is therefore urgently necessary to enhance the resistance of deep learning-based classifiers to adversarial attacks.

In this paper, we propose a novel wireless receiver architecture that enhances the resistance of the receiver's modulation classifier to adversarial attacks. Our proposed defense technique is threefold. First, the amount of adversarial perturbation in a modulated signal is estimated by relying on a DAE that has been specially trained to remove Gaussian noise and adversarial perturbations. Then, signals with considerable perturbations are preprocessed using the DAE to remove those undesirable attributes. Signals with small amounts of noise and adversarial perturbations, on the other hand, are not preprocessed as the DAE could introduce errors that are more significant than the perturbations. Finally, the signal's modulation scheme is identified with an enhanced classifier that has been trained using noisy and adversarial samples to make it resistant to sample variation.

In contrast, most existing defense techniques do not effectively remove adversarial perturbations as they focus only on detecting adversarial samples and improving the classifier. Thus, compared to existing defense schemes, our proposed solution's first major technical improvement is our

technique for estimating and removing adversarial perturbations, which significantly alleviates the burden on the classifier. Moreover, while most existing defense schemes enhance the classifier's resistance to adversarial attacks by including adversarial samples in training, they are effective only against the adversarial attacks whose samples were considered. On the other hand, our proposed defense technique relies on our previous work in [Freitas de Araujo-Filho *et al.* (2022)] to quickly craft and include in training adversarial samples that generalize other adversarial attacks. Therefore, our proposed solution's second major technical improvement is its ability to enhance modulation classifiers' resistance to various adversarial attack techniques while requiring only adversarial samples crafted using a single fast attack technique [Freitas de Araujo-Filho *et al.* (2022)]. These improvements enable our technique to diminish the accuracy reduction that is caused by adversarial attacks by at least 18% more than existing defense techniques. In a nutshell, the main contributions of our work are as follows:

- We propose a DAE that has been specially trained to estimate and remove noise and adversarial perturbations from modulated signals.
- We propose an enhanced modulation classifier (EMC) that is resistant to a variety of adversarial attack techniques.
- We propose a novel wireless receiver architecture that is resistant to adversarial attacks by combining our proposed DAE and EMC to remove adversarial perturbations and make the classifier less affected by them.

The remainder of this article is organized as follows. Section 5.3 reviews the existing techniques to defend against adversarial attacks on modulation classifiers. In Section 5.4, we formulate adversarial attacks and describe the threat model and assumptions considered in our work. In Section 5.5, we present our proposed wireless receiver architecture by describing our proposed DAE and EMC. Section 5.6 describes the dataset used, the experiments conducted, and the adversarial attacks considered in our evaluation. In Section 5.7, we present and discuss our solution's results and compare them to the results of state-of-the-art defense techniques. Finally, Section 5.8 concludes our paper and proposes future extensions to our work.

5.3 Related Works

Despite the severe risks adversarial attacks on deep learning-based modulation classifiers pose to wireless communications, only a few techniques have been proposed to defend modulation classifiers against them. The work in [Sahay *et al.* (2022)] proposes a wireless transmission receiver architecture that reduces the risks of a modulation classifier experiencing adversarial attacks. The defense technique consists of using an ensemble of eight classifiers to recognize modulation schemes as it is more challenging for an attacker to simultaneously fool several classifiers than just one. However, considering many classifiers significantly increases the computational resources required and the time it takes to recognize a signal's modulation scheme.

The work in [Shtaiwi *et al.* (2022)] proposes a defense technique that discards adversarial samples before they are sent to the modulation classifier. It relies on mixture generative adversarial networks (MGAN) and trains a GAN for each modulation scheme considered. However, the technique proposed in [Shtaiwi *et al.* (2022)] also significantly increases the computational resources required as one GAN is trained for each modulation scheme. Moreover, the authors of [Shtaiwi *et al.* (2022)] evaluate their proposal against only adversarial samples that are crafted using the FGSM technique and do not indicate the size of adversarial perturbations or if they are imperceptible.

The authors of [Sahay *et al.* (2021)] propose a defense technique for modulation classifiers that detects large adversarial perturbations by computing a reconstruction loss with an autoencoder. Moreover, it includes adversarial samples that have been crafted using the FGSM technique when training the classifier so that the classifier learns how to classify them correctly. However, while large perturbations are detected but not correctly classified, small perturbations are correctly classified only if they were crafted using the FGSM technique, as only those types of perturbations are considered when training the classifier. Similarly, the work in [Kim, Sagduyu, Davaslioglu, Erpek & Ulukus (2021)] enhances a modulation classifier's resistance by augmenting its training data with Gaussian noise. However, it does not significantly prevent

adversarial attacks from reducing the classifier's accuracy as they optimally find perturbations that compromise classifiers.

The authors of [Manoj, Santos, Sadeghi & Larsson (2022)] evaluate the performance of modulation classifiers enhanced using three defense techniques: randomized smoothing, hybrid PGD adversarial training, and fast adversarial training. Randomized smoothing augments the classifier's training data with Gaussian noise as it is also done by the work in [Kim *et al.* (2021)]. The hybrid PGD adversarial training and fast adversarial training techniques, on the other hand, augment the classifier's training data with adversarial samples crafted using the PGD and universal adversarial perturbation (UAP) techniques, respectively. However, the results in [Manoj *et al.* (2022)] show that none of those three techniques is effective as they do not prevent white-box attacks from reducing the classifier's accuracy to less than 20%.

The work in [Zhang, Lambotharan, Zheng & Roli (2021b)] proposes a neural rejection (NR) system that detects adversarial attacks on modulation classifiers. It trains a support vector machine (SVM) model for each modulation scheme considered so that they detect samples that differ from the clean samples used in training and consider them to contain adversarial perturbations. The authors of [Zhang *et al.* (2022)] propose a hybrid training-time and run-time defense (HTRD) technique that combines the customized adversarial training (CAT) technique with the NR system developed in [Zhang *et al.* (2021b)]. The CAT technique enhances classifiers by augmenting their training samples with adversarial samples crafted using a modified PGD attack. Since adversarial samples are used in training, the classifier learns how to classify them correctly. However, the authors of [Zhang *et al.* (2021b)] and [Zhang *et al.* (2022)] evaluate their defense techniques against only one adversarial attack technique. Moreover, their NR system significantly increases the computational resources needed as one SVM model is required for each modulation class.

Finally, the authors of [Zhang *et al.* (2021a)] propose a defense mechanism that combines the NR mechanism proposed in [Zhang *et al.* (2021b)] with two techniques that enhance the classifier's resistance to adversarial attacks: Gaussian noise augmentation (GNA) and label

smoothing (LS). The GNA technique adds Gaussian noise to training samples. The LS technique converts labels that were encoded using the one-hot encoding technique, such as (1, 0, 0, 0), into smoothed vectors that reduce the classification confidence, such as (0.91, 0.03, 0.03, 0.03). These techniques help the neural network classifier to better generalize by not being overconfident. Although the work in [Zhang *et al.* (2021a)] diminishes the degree to which FGSM adversarial samples reduce the classifier's accuracy, it does not consider other types of adversarial attacks. In addition, as in [Zhang *et al.* (2021b)] and [Zhang *et al.* (2022)], it also significantly increases the computational resources needed since it employs the NR mechanism.

5.4 Adversarial Attack Threat Model

Deep learning-based classifiers are trained to find decision boundaries between the decision regions of each class. However, as shown in Figure 5.1, adversarial attacks craft and introduce perturbations that modify data samples and force them to cross decision boundaries and lie in other decision regions. However, these perturbations must be small enough to not be perceived. Thus, adversarial attacks aim to find a perturbation δ that, when added to a sample x, modifies it just enough so that the adversarial sample $x_{adv} = x + \delta$ satisfies

$$\min ||x_{adv} - x|| < \rho \tag{5.1}$$

and

$$f(x_{adv}) \neq f(x), \tag{5.2}$$

where $||\cdot||$ represents a chosen distance metric, ρ is the maximum imperceptible perturbation according to that metric, and f is the trained classifier that is the target of the attack.

In our work, we consider the worst-case scenario in which the attacker has complete knowledge about the classifier. That is, we evaluate our proposed defense technique against white-box adversarial attacks. Furthermore, we consider that adversarial attacks can be launched directly on wireless receivers, from wireless transmitters, or from separate malicious emitters. When launched on receivers, attackers need to infect the receiver with malware or a malicious piece

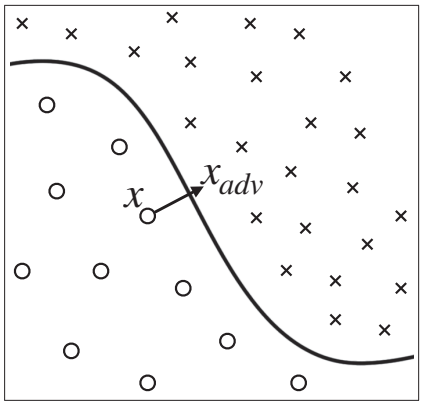


Figure 5.1 Adversarial sample crossing decision boundary

of hardware that tampers with incoming signals to add adversarial perturbations. Similarly, when launched from transmitters, attackers need to infect the transmitter so that it can tamper with outgoing signals and add adversarial perturbations to them. Attackers must also consider channel effects as the adversarial perturbations are transmitted over the air. Finally, when launched from separate malicious emitters, as shown in Figure 5.2, attackers must eavesdrop on the transmitter's signals and consider the perfect synchronization of perturbations and signals, as they are transmitted from different nodes. Since our focus is on defending against adversarial attacks, we assume that attackers are successful in crafting, transmitting, and synchronizing perturbation signals. We therefore consider those tasks to be outside the scope of our work.

5.5 Proposed Wireless Receiver Architecture

In our work, we propose a novel wireless receiver architecture that protects against adversarial attacks on deep learning-based modulation classifiers. Our proposed system has two goals. The first is to remove adversarial perturbations from samples so that they are not forced across decision boundaries. The second is to make the modulation classifier less sensitive to the changes caused by adversarial perturbations so that it is more difficult to force samples across decision boundaries.

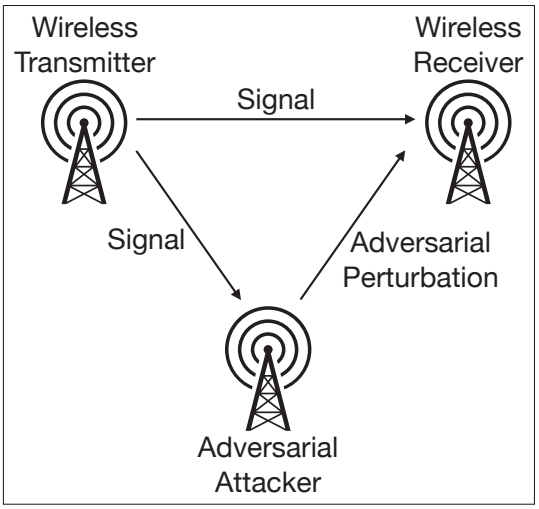


Figure 5.2 Adversary attack model as a perturbation transmitted over the air

To achieve those goals, our proposed system consists of two modules, namely, an adversarial perturbation preprocessor (APP) and an enhanced modulation classifier (EMC). Figure 5.3 shows our proposed architecture. The ADC forwards the received samples to our proposed APP module, which processes and forwards them to the EMC module. Finally, the EMC module classifies the samples and indicates the recognized modulation scheme to the receiver's demodulator.

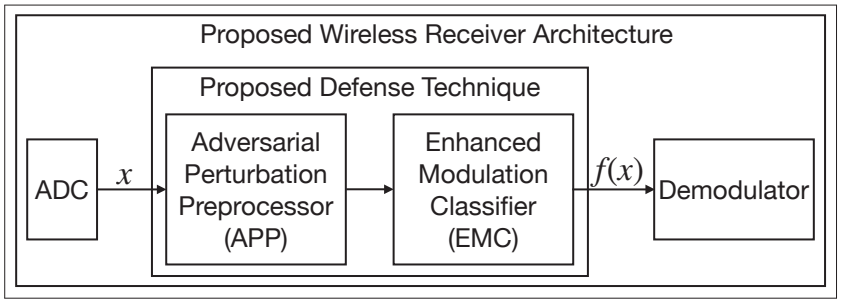


Figure 5.3 Proposed wireless receiver architecture

5.5.1 Adversarial Perturbation Preprocessor

The APP module trains a DAE using Gaussian and adversarial samples so that it learns how to remove noise and adversarial perturbations from samples. During training, the DAE learns how to map samples that have been corrupted with Gaussian noises and adversarial perturbations to clean samples. It is trained to minimize the loss function

$$L_{\text{DAE}} = \frac{1}{N}\sum_{j=1}^{N}(x_{o_j} - x_{i_j})^2, \tag{5.3}$$

where $x_i = x_{\text{clean}} + \eta + \delta$ is the input sample that may or may not have been corrupted by noise η and adversarial perturbations δ, x_o is the DAE's output after noise and adversarial perturbations have been removed, and N is the length of samples x_i and x_o.

The cosine distance between x_i and x_o measures the dissimilarity between them, which represents the correction c that is applied by the DAE to remove noise and adversarial perturbations. Small cosine distances correspond to null or small corrections that happen when input samples have not been tampered with or when they have been altered by small perturbations. Large cosine distances, on the other hand, correspond to large corrections that are applied as a result of large perturbations. Thus, this cosine distance allows us to estimate the amount of perturbation in a sample.

However, since it is impractical to train the DAE (or any other deep learning model) to cover all possible input feature vectors, the DAE may also introduce small errors. Thus, the DAE's output is given by $x_o = x_i + c + e$, where c is the correction that the DAE applies to input samples and e is the error that it introduces. As a result, our proposed defense technique must use the DAE to preprocess data samples only when the perturbations removed are larger than the errors introduced. Otherwise, the DAE may harm classification more than it helps. Therefore, our APP module first estimates the amount of perturbation in a sample by computing the cosine distance between x_i and x_o, and then forwards to the EMC module either x_i when the perturbation is small or x_o when the perturbation is large.

5.5.2 Enhanced Modulation Classifier

The EMC trains deep convolutional neural networks to classify samples of modulated signals by their modulation scheme. Similarly to the DAE, the modulation classifier is trained using samples that have been corrupted with Gaussian noise and adversarial samples in addition to clean samples. Augmenting the training set with Gaussian noise increases the classifier's resistance to multiple directions, i.e., samples that have been slightly dislocated in random directions are still assigned to the same class of x. Similarly, augmenting the training set with adversarial perturbations increases the classifier's resistance to the direction that makes a sample optimally cross the decision boundary according to an adversarial attack technique. As a result, our proposed EMC makes the classifier's prediction of a sample x constant within a small neighborhood around x. Therefore, the decision boundaries become less sensitive, and the classifier becomes more resistant to changes caused by noise and adversarial perturbations. Algorithm 5.1 summarizes how our proposed defense technique works.

Algorithm 5.1 Proposed defense technique

1: Train a DAE with samples tampered with Gaussian noise and adversarial perturbations
2: Train a EMC with samples tampered with Gaussian noise and adversarial perturbations
3: **for** Each incoming sample x_i **do**
4: Compute $x_o = DAE(x_i)$
5: Compute $\beta = CD(x_i, x_o)$
6: **if** $\beta \geq t$ **then**
7: Preprocess data sample $x = x_o$
8: **else**
9: Do not preprocess data sample $x = x_i$
10: **end if**
11: Classify data sample $y = f(x)$
12: **end for**

5.5.3 Adversarial Samples for Training

Our proposed architecture relies on adversarial samples to train both the DAE and the EMC. The DAE leverages adversarial samples to learn how to remove adversarial perturbations. The EMC uses them to enhance its resistance to them. Thus, the choice of adversarial samples considered has a significant impact on the resistance our technique provides. For instance, if our proposed DAE and EMC are trained with adversarial samples crafted using only the FGSM technique, our defense will be effective against only FGSM adversarial samples. Similarly, if we consider adversarial samples crafted using only the FGSM and PGD techniques, our defense technique will protect wireless receivers from only those two specific adversarial attacks. However, it is not feasible to consider many different adversarial attack techniques as doing so would significantly increase our defense technique's computational requirements and training time.

Therefore, a crucial part of our proposed defense technique is to consider an adversarial attack technique that generalizes other types of adversarial attacks. We want our defense technique to protect against different types of adversarial attacks while being trained with adversarial samples crafted using a single attack technique. For this purpose, we leverage our previous work in [Freitas de Araujo-Filho *et al.* (2022)], in which we proposed an input-agnostic adversarial attack technique. This type of attack combines GANs [Goodfellow *et al.* (2014)] and multi-task loss [Kendall *et al.* (2018)] to generate adversarial samples by simultaneously optimizing their ability to cause wrong classifications and not be perceived. Furthermore, it crafts adversarial samples much faster than other adversarial attack techniques. Thus, by using the adversarial attack technique proposed in [Freitas de Araujo-Filho *et al.* (2022)], our proposed defense technique enhances modulation classifiers' resistance to different types of adversarial attacks while also significantly reducing the time it takes to craft the adversarial samples used to train the DAE and EMC.

5.6 Methodology and Experimental Evaluation

In this section, we present the dataset that we used in our experiments to validate our proposed defense technique and then explain the experiments we conducted and the neural network architectures of our proposed DAE and EMC modules.

5.6.1 Dataset

To evaluate our proposed defense architecture, we used DeepSig's publicly available RADIOML 2016.10A dataset [O'Shea *et al.* (2016); O'Shea & West (2016)]. The dataset contains signals that have been modulated using one of eleven modulation schemes (eight digital and three analog): 8PSK, BPSK, QPSK, QAM16, QAM64, CPFSK, GFSK, PAM4, WBFM, AM-DSB, and AM-SSB. The signals are then exposed to an AWGN channel that includes sampling rate offset, random process of center frequency offset, multipath, and fading effects, as described in [O'Shea *et al.* (2016); O'Shea & West (2016)]. Since our goal is to defend against adversarial attacks, we consider an AWGN channel rather than other channel models that could negatively impact the attacks' performance [Flowers *et al.* (2020); Kim *et al.* (2021)]. Moreover, we assume adversarial attacks successfully account for channel and transmission effects without compromising their harmfulness. Finally, the signals are normalized and packaged into 220,000 samples of in-phase and quadrature components of length 128 that are each associated with one of the eleven modulation schemes and a SNR. The SNR indicates the signal's strength. It is the ratio between the power P of the signal and of the background noise, i.e., $SNR_{[dB]} = 10\log(\frac{P_{signal}}{P_{noise}})$. The dataset covers twenty SNRs ranging from -20 dB to 18 dB in steps of 2 dB. Sixty four percent of the samples were used for training our proposed DAE and EMC, 16% were used as a validation set, and 20% were reserved as a testing set to measure and evaluate our proposed architecture's performance.

5.6.2 DAE Experiments

Our proposed DAE relies on fully connected neural networks to encode and decode samples of modulated signals. It encodes clean, noisy, and adversarial samples into a lower-dimensional space, and reconstructs them as clean samples that are free of noise and adversarial perturbations. Noisy samples are produced by adding to clean samples Gaussian noise generated with zero mean and standard deviation σ. Adversarial samples, on the other hand, are produced by crafting and adding to clean samples adversarial perturbations generated using the technique proposed in [Freitas de Araujo-Filho *et al.* (2022)]. Figure 5.4 shows our proposed DAE's architecture.

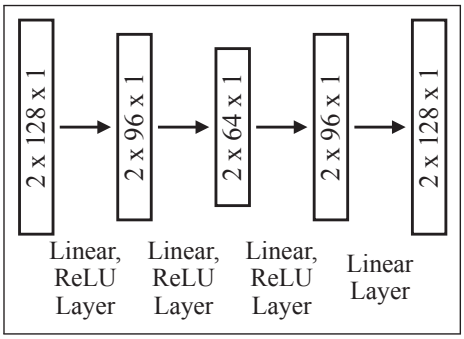


Figure 5.4 DAE neural network architecture

While large standard deviations allow the DAE to remove more considerable noises, they may also induce the DAE to produce more significant errors. Moreover, the more noisy and adversarial samples that are considered in training, the better the DAE gets at removing noise and adversarial perturbations. However, considering too many noisy and adversarial samples may significantly increase the DAE's training time. Thus, we balance these trade-offs by experimenting with several different standard deviations and proportions of noisy and adversarial samples to each clean sample. Furthermore, we optimize the DAE's hyper-parameters, such as learning rate and batch size, using the Optuna framework [Akiba *et al.* (2019)], which automatically searches for the optimal hyper-parameters and the early stopping mechanism. Table 5.1 shows the hyper-parameter values used in the DAE after tuning.

Table 5.1 Hyper-parameter values of the DAE

Hyper-Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	128
Maximum Number of Epochs	100
Early Stopping Patience	5
Standard deviation of Gaussian samples	0.0025
Number of Gaussian samples per clean sample	5
Number of adversarial samples per clean sample	5

Finally, we define the threshold t to which the cosine distance is compared in Algorithm 5.1 as $t = \gamma\tau$, where γ is a hyper-parameter and τ represents the average error introduced by the DAE. Since the DAE is supposed to not change input samples when they do not contain noise or adversarial perturbations, the cosine distance between clean training samples and their reconstructions corresponds to the error introduced by the DAE. Thus, we compute τ by averaging the cosine distances of clean training samples.

5.6.3 EMC Experiments

Rather than improving modulation classifier's accuracy, the main goal of our work is to make them resistant to adversarial attacks, i.e., diminish the degree to which adversarial attacks reduce their accuracy. Thus, instead of proposing a novel neural network architecture, our EMC module relies on deep convolutional neural networks and uses the same architecture as the VT-CNN2 modulation classifier proposed in [O'Shea *et al.* (2016); O'Shea & West (2016)]. This classifier has been largely adopted by most of the works that investigate adversarial attack defense techniques for modulation classifiers. Similarly to the DAE, we optimize the EMC's hyper-parameters using the Optuna framework [Akiba *et al.* (2019)] and the early stopping mechanism. Figure 5.5 shows our proposed EMC's architecture. Table 5.2 shows the hyper-parameter values used in the EMC after tuning. All experiments were conducted using an AMD Ryzen Threadripper 1920X 12-core 2.2GHz processor with 64GB of RAM and an NVIDIA GeForce RTX 2080 in a Pytorch environment.

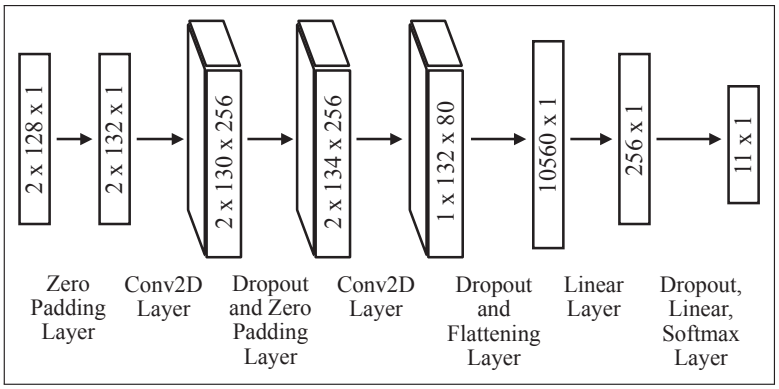


Figure 5.5 EMC neural network architecture

Table 5.2 Hyper-parameter values of the EMC

Hyper-Parameter	**Value**
Optimizer	Adam
Learning Rate	0.001
Batch Size	1024
Dropout Rate	0.25
Maximum Number of Epochs	100
Early Stopping Patience	5
Standard deviation of Gaussian samples	0.0025
Number of Gaussian samples per clean sample	10
Number of adversarial samples per clean sample	10

5.6.4 Adversarial Attacks Considered

As discussed, we evaluated our proposed defense technique against the worst-case scenario of white-box attacks. We selected the FGSM, PGD, and MIM adversarial attacks, which have been shown to significantly compromise the accuracy of deep learning-based modulation classifiers [Lin *et al.* (2021, 2020)].

The FGSM adversarial attack modifies input features by increasing and decreasing them according to the sign of the loss function's gradient. Hence, it is formulated as

$$x_{adv} = x + \varepsilon \text{sign}(\nabla_x J(\theta, x, y)), \tag{5.4}$$

where x_{adv} is the adversarial sample, x is the input sample, $J(\theta, x, y)$ is the classifier's loss function, and ε is a control variable that scales the adversarial perturbation. While the FGSM technique crafts adversarial samples quickly, it may modify all input features so that adversarial samples are more likely to be perceived [Liu, Nogueira, Fernandes & Kantarci (2022); Manoj *et al.* (2021)].

While the FGSM attack technique crafts adversarial samples in a single step, the PGD technique follows an iterative process. It starts with a randomly initialized adversarial sample within the clean sample's L_∞ proximity. Then, it takes gradient steps in the direction of the greatest loss until convergence is achieved. The PGD technique is formulated as

$$\begin{cases} x_{adv}^{(t+1)} = \prod_{x+\Psi}(x_{adv}^{(t)} + \varepsilon \text{sign}(\nabla_x J(\theta, x, y))) \\ x_{adv}^{(0)} = x \end{cases}, \tag{5.5}$$

where $x_{adv}^{(t+1)}$ is the adversarial sample at iteration $t+1$, $J(\theta, x, y)$ is the classifier's loss function, ε is a control variable that scales the adversarial perturbation, and Ψ is the set of allowed perturbations so that x_{adv} remains within the L_∞ neighborhood of the clean sample x. Although the PGD technique's iterations result in a longer training time, it produces adversarial samples that are more harmful than those produced by the FGSM technique [Liu *et al.* (2022); Manoj *et al.* (2021)].

Finally, while the MIM technique also follows an iterative process, it introduces momentum, which adds a fraction of the previous weight update to the current one. Momentum speeds up convergence and helps avoid local minima, better approximating the optimal attack direction. The MIM technique is formulated as

$$\begin{cases} x_{adv}^{(t+1)} = x_{adv}^{(t)} + \frac{\varepsilon}{T_{\max}} \text{sign}(g^{(t+1)}) \\ g^{(t+1)} = \mu g^{(t)} + \frac{\nabla_x J(\theta, x_{adv}^{(t)}, y)}{||\nabla_x J(\theta, x_{adv}^{(t)}, y)||} \\ x_{adv}^{(0)} = x \\ g^{(0)} = \frac{\nabla_x J(\theta, x, y)}{||\nabla_x J(\theta, x, y)||} \end{cases}, \tag{5.6}$$

where $x_{adv}^{(t+1)}$ is the adversarial sample at iteration $t+1$, $J(\theta, x, y)$ is the classifier's loss function, ε is a control variable that scales the adversarial perturbation, $T_{\max}$ is the number of iterations, and μ is a decay factor. The momentum improves stability so that the MIM technique provides stronger generalization and MIM adversarial attacks usually outperforms PGD adversarial attacks [Lin *et al.* (2020, 2021)].

5.7 Results and Discussion

Adversarial attacks on modulation classifiers aim to tamper with signals and reduce a classifier's accuracy while ensuring that perturbations are not perceived. While more significant adversarial perturbations compromise the classifier's accuracy more, they are more distinguishable and likely to be detected. Adversarial perturbations are considered imperceptible when they cannot be distinguished from the noise, i.e., they are in the same order as or below the noise level. Hence, we measure the PNR, i.e., the ratio between the power levels of the perturbation and noise $PNR_{[dB]} = 10\log(\frac{P_{perturbation}}{P_{noise}})$, where $P_{perturbation}$ is the power level of the perturbation power and P_{noise} is the power level of the noise, so that adversarial perturbations are considered imperceptible when PNR < 0 dB.

We first evaluate how much adversarial attacks compromise the VT-CNN2 modulation classifier's accuracy. Figure 5.6 shows the VT-CNN2's accuracy versus PNR for an SNR of 10 dB. While the classifier's accuracy is around 75% without attacks, it is significantly compromised by the FGSM, PGD, and MIM attacks. They reduce the classifier's accuracy by 10 percentage points with adversarial perturbations as low as -16 dB PNR. Moreover, at 0 dB PNR, the FGSM attack reduces the classifier's accuracy to only 30%, and the PGD and MIM attacks reduce it to only 26%. Although the VT-CNN2's accuracy is not very high, we use that classifier because it is largely adopted by most of our related works. Moreover, our goal is not to increase the classifier's accuracy, but to prevent it from being reduced. In addition, our proposed defense architecture does not depend on the classifier's architecture so it can easily be replicated with any other deep learning-based modulation classifier.

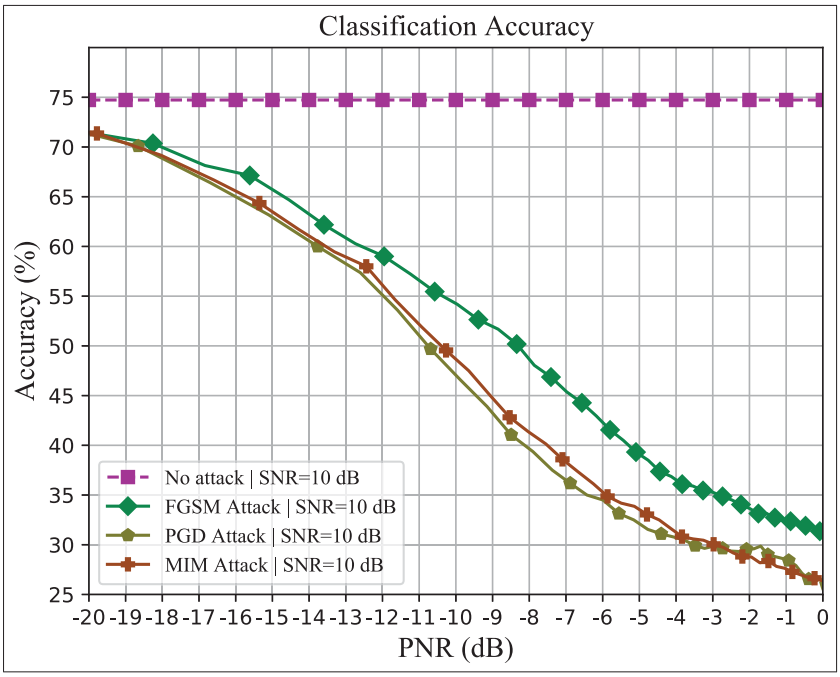


Figure 5.6 VT-CNN2 modulation classifier's accuracy versus PNR

Our proposed APP module estimates the amount of adversarial perturbation in samples by measuring the cosine distance between them and their reconstructed versions obtained from the DAE. Figure 5.7 shows the cosine distance for clean and adversarial samples with PNRs of -20 dB to 0 dB. Small cosine distances indicate that samples are clean or that they have been altered by small perturbations. Thus, the cosine distances of clean samples and adversarial samples with PNRs below -7 dB are small. On the other hand, large cosine distances indicate that samples have been altered by more substantial perturbations. Thus, the cosine distance of adversarial samples with PNRs above -7 dB significantly increases with the PNR.

Furthermore, Figure 5.7 shows that the cosine distance of clean samples is small but not zero because the DAE introduces a small error. As a result, our proposed defense technique preprocesses samples only when the DAE removes more adversarial perturbations than the error it adds, i.e., when the cosine distance measured between incoming sample and its reconstruction is above the threshold set by the cosine distance of clean training samples as described in Algorithm 5.1.

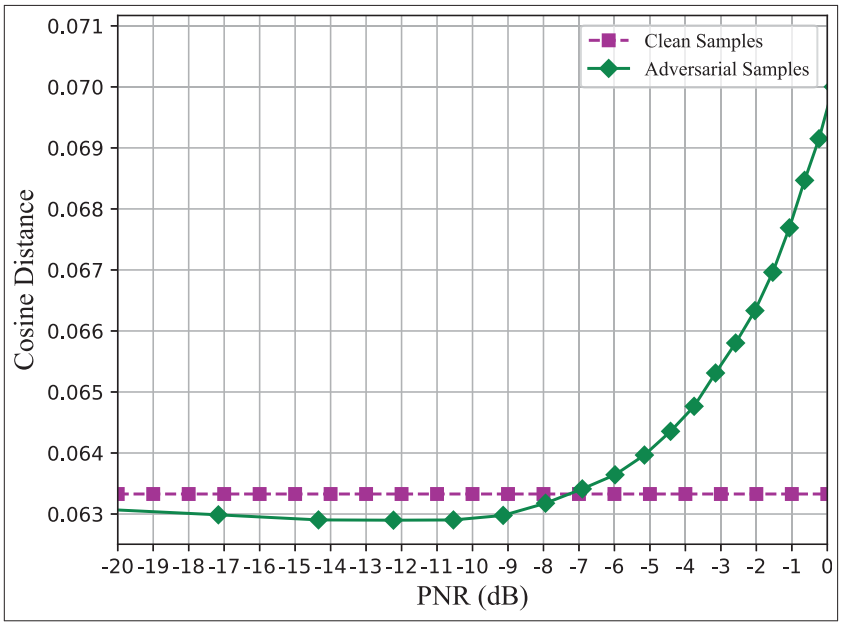


Figure 5.7 Cosine distance between clean and adversarial samples and their reconstructions

We evaluate our proposed defense technique's performance against FGSM, PGD, and MIM adversarial attacks while also assessing our proposed EMC and APP modules' individual contributions to the final defense result. Figures 5.8, 5.9, and 5.10 show the modulation classifier's accuracy against FGSM, PGD, and MIM attacks, respectively, for a SNR of 10 dB without any defense technique, with our proposed defense technique, with only our proposed EMC, and with only our proposed APP. Enhancing the modulation classifier using our EMC diminishes the accuracy reduction that is caused by the three types of adversarial attacks considered because the classifier becomes less sensitive to perturbations. However, the larger the adversarial perturbation, the worse the EMC performs. On the other hand, although our APP cannot improve the classifier's accuracy as much as the EMC does for low PNRs, it ensures less accuracy reduction than the EMC does for higher PNRs because it removes large perturbations. Therefore, by combining the EMC and APP modules, our proposed defense technique significantly diminishes the degree to which the three adversarial attack types considered reduce the classifier's accuracy.

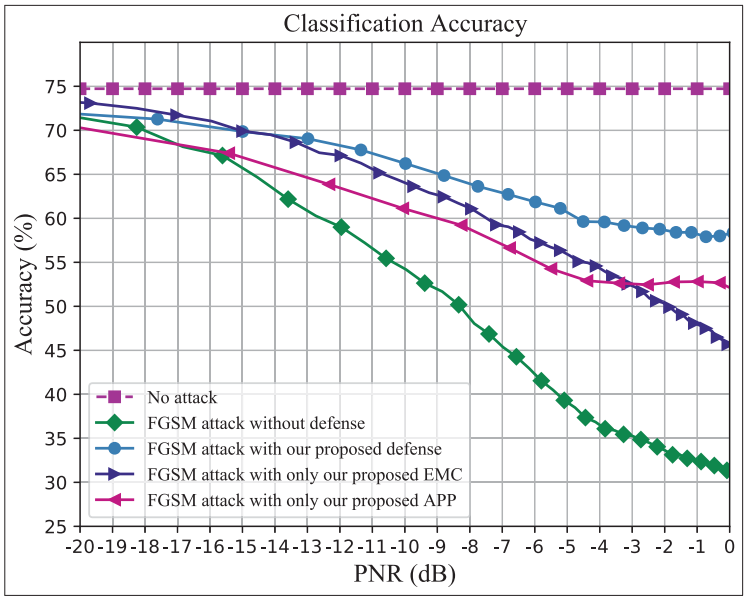


Figure 5.8 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the FGSM adversarial attack for a SNR of 10 dB

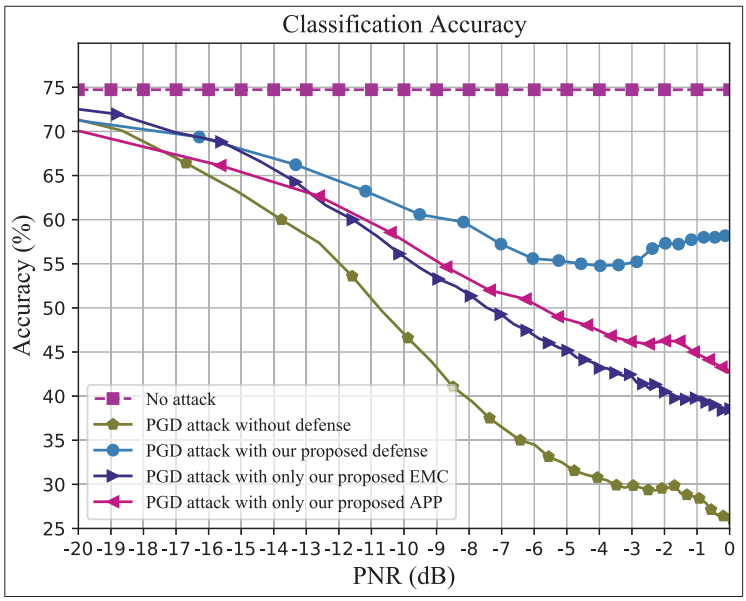


Figure 5.9 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the PGD adversarial attack for a SNR of 10 dB

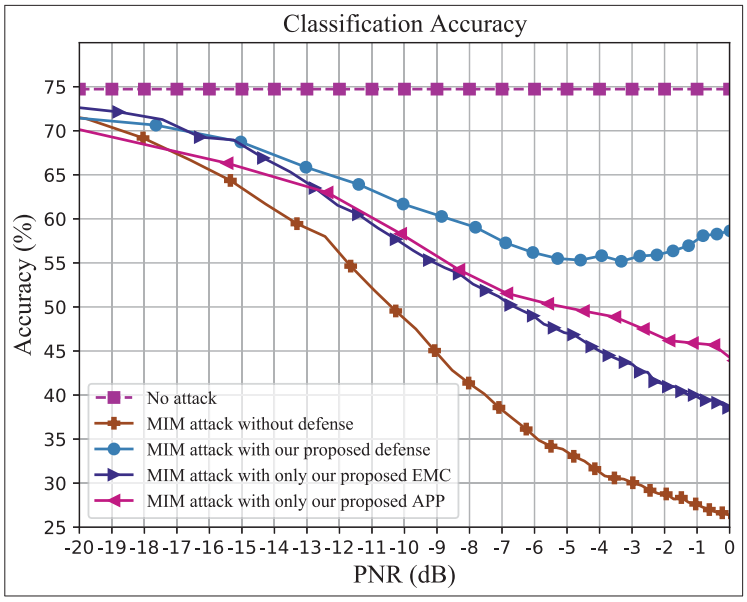


Figure 5.10 Contribution of our proposed APP and EMC to the modulation classifier's accuracy against the MIM adversarial attack for a SNR of 10 dB

Finally, we compare the performance of our proposed defense technique to that of the defense techniques proposed in [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)]. Figures 5.11, 5.12, and 5.13 show the modulation classifier's accuracy against FGSM, PGD, and MIM attacks, respectively, for a SNR of 10 dB without any defense technique, with our proposed defense technique, and with the defense techniques proposed in [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)]. While the techniques proposed in [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)] reduce the classifier's sensitivity to adversarial samples using NR, CAT, GNA, and LS, our solution removes large perturbations using the APP in addition to reducing the classifier's sensitivity to adversarial samples using the EMC.

Although the defense techniques proposed in [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)] diminish the degree to which small adversarial perturbations reduce the classifier's accuracy, they are ineffective against larger adversarial perturbations. For instance, at 0 dB PNR, the classifier's accuracy when the techniques from [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)] are employed is less than 10% greater than when no defense technique is used. While our

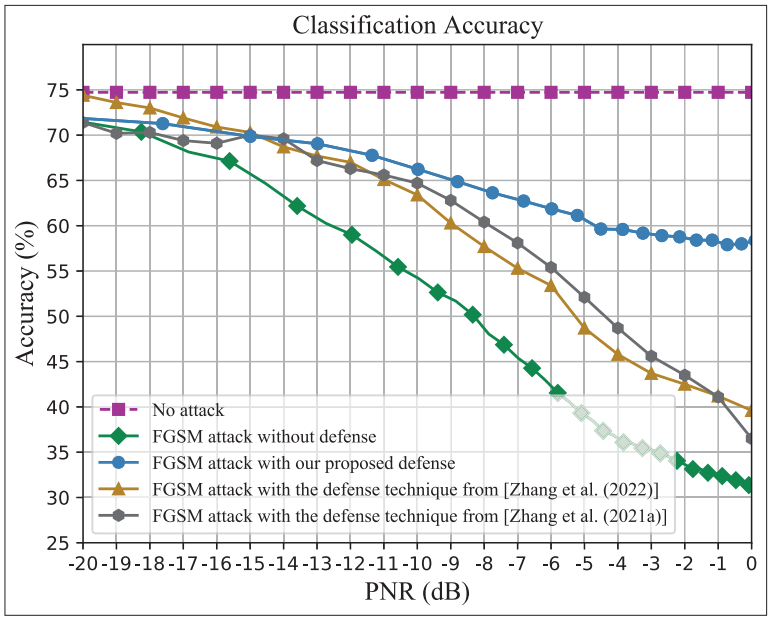


Figure 5.11 Modulation classifier's accuracy versus PNR against the FGSM adversarial attack for a SNR of 10 dB

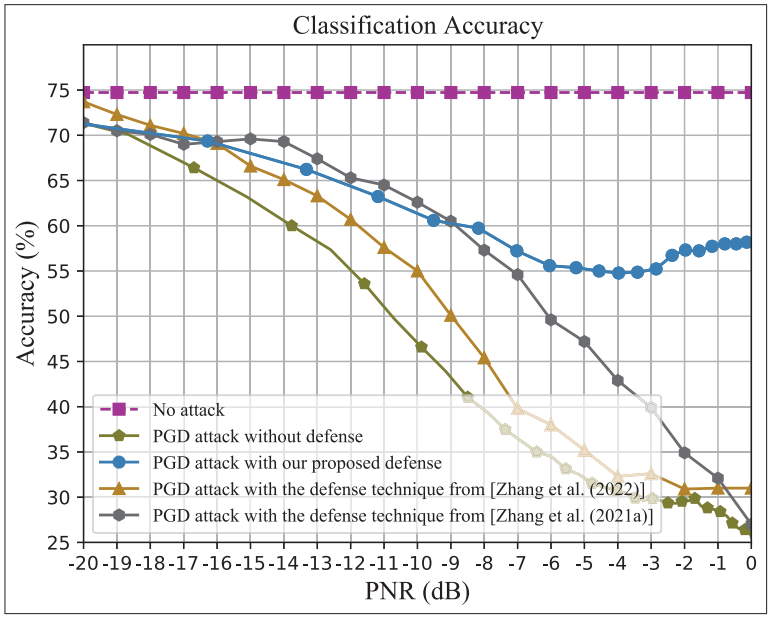


Figure 5.12 Modulation classifier's accuracy versus PNR against the PGD adversarial attack for a SNR of 10 dB

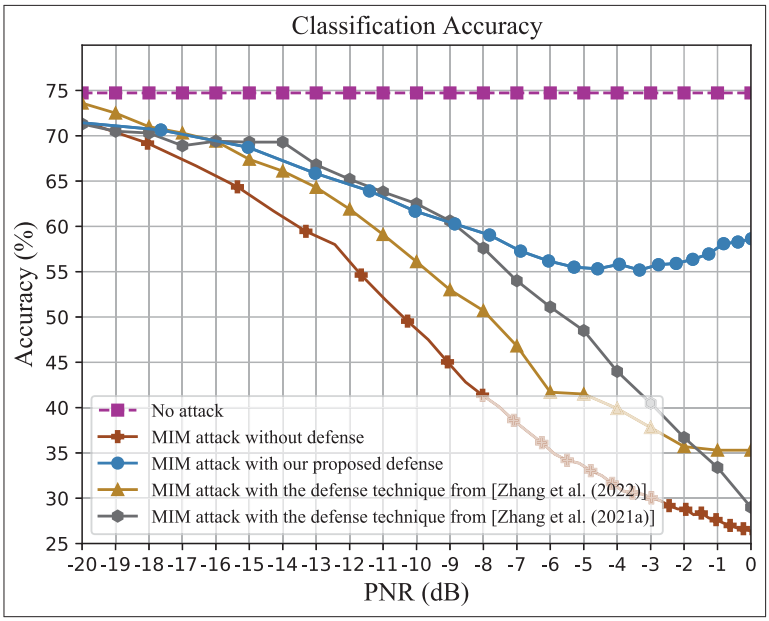


Figure 5.13 Modulation classifier's accuracy versus PNR against the MIM adversarial attack for a SNR of 10 dB

proposed technique performs similarly to the techniques proposed in [Zhang *et al.* (2022)] and [Zhang *et al.* (2021a)] against small perturbations, it significantly outperforms them against larger perturbations. Our technique diminishes the degree to which accuracy is reduced by at least 18 percentage points more than [Zhang *et al.* (2022)] does and 20 percentage points more than [Zhang *et al.* (2021a)] does at 0 dB PNR. This improvement is a result of our proposed APP module, which preprocesses samples that have large adversarial perturbations using our proposed DAE. By removing noise and adversarial perturbations, our DAE makes it much easier for the EMC to classify samples, hence the improvement achieved.

5.8 Conclusion

In this paper, we verified that adversarial attacks significantly compromise deep learning-based modulation classifiers. Then, we proposed a novel wireless receiver architecture that protects modulation classifiers from adversarial attacks by combining two modules: an APP and an EMC. The APP estimates adversarial perturbations in incoming samples and removes them by preprocessing samples with a specially designed DAE. The samples are then forwarded to

be classified in the EMC, which has been specially trained to be less sensitive to adversarial perturbations.

In terms of our proposed EMC, our results show that it successfully diminishes the accuracy reduction that is caused by the three adversarial attack types considered. However, they also show that it degrades when it comes to large adversarial perturbations. In terms of our proposed APP, our results show that it successfully removes large perturbations and therefore ensures less accuracy reduction than the EMC does for higher PNRs. Finally, our results show that, by combining both the EMC and APP modules, our proposed defense technique diminishes the degree to which the three adversarial attacks considered reduce the classifier's accuracy by at least 18% more than existing defense techniques. Therefore, we verified that better defense results are achieved by simultaneously removing adversarial perturbations and making classifiers less sensitive to them.

In future work, we will evaluate our proposed technique against other adversarial attack techniques and investigate how to improve it to make the modulation classifier's accuracy even closer to when there is no adversarial attack. Furthermore, we will also investigate how our proposed defense technique can be adapted to protect other deep learning-dependent wireless communication tasks, such as resource allocation, from adversarial attacks.

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

As the increasing number of connected devices and the use of ML introduce new security challenges, it is necessary to enhance the security of connected things against cyber-attacks and adversarial attacks that can compromise confidentiality, integrity, and availability. Therefore, this thesis studies new strategies and techniques to protect connected things against cyber-attacks and adversarial attacks. We focus on developing novel intrusion detection systems that effectively and efficiently detect cyber-attacks. Moreover, we also investigate the impact that adversarial attacks and the development of defense techniques that mitigate their effects.

Chapter 2 proposed FID-GAN, a GAN-based IDS for detecting cyber-attacks on cyber-physical systems. We combined the discrimination and reconstruction losses of the GAN to compute an anomaly detection score that indicates the probability that the data samples correspond to anomalies. Our experiments verified that combining both losses made it possible to achieve higher AUCROC values, allowing our proposed IDS to achieve lower false positive and negative rates simultaneously. Furthermore, our proposed IDS presented an innovative approach to train an encoder neural network that accelerates the reconstruction loss computation, hence significantly reducing the detection time. Finally, to further minimize the detection time, we proposed a deployment architecture in which the GAN and encoder were trained in the cloud but deployed for inference at a fog-layer closer to the nodes under surveillance.

Chapter 3 considered the detection of known and unknown DDoS attacks that could severely compromise the availability of networks and systems. We evaluated different neural network architectures that consider time dependencies among data by relying on a GAN-based IDS. Our experimental results showed that LSTM networks, which were until recently considered the architecture to go for sequence modeling tasks, were outperformed by other neural network

architectures. Precisely, using self-attention networks granted higher detection rates than LSTM networks, while using TCNs provided shorter detection times. Therefore, this investigation proved that self-attention and TCNs could replace LSTM networks in detecting cyber-attacks.

Chapter 4 investigated how adversarial attacks could compromise the availability of wireless communications. We formulated adversarial attacks as an optimization problem that aims to craft adversarial perturbations that induce ML-based classifiers to make mistakes while not being perceived. Then, we proposed a technique that leverages the multi-task loss for training a GAN that produces adversarial perturbations simultaneously optimizing those two conditions. Moreover, our proposed technique only required access to the target model decisions and was proved to be input-agnostic. Our experiments showed that our technique was able to cause more damage to the accuracy of a modulation classifier that other adversarial attack techniques while being 335 times faster than them. The study in this chapter verified that adversarial attacks could significantly impact the security of systems that rely on ML. Furthermore, it served as the basis for proposing defense techniques against adversarial attacks, as demonstrated in Chapter 5.

Finally, Chapter 5 investigated techniques for enhancing the security of machine-based systems by reducing the extent to which adversarial attacks compromise them. We proposed a defense technique that estimates and removes large adversarial perturbations so that samples of modulated signals received at a wireless receiver are not forced across the decision boundaries of modulation classifiers using a specially trained DAE. Moreover, our proposed technique relies on an EMC that has been specially trained to reduce the sensitivity of its decision boundaries, further reducing the effects of adversarial attacks. The DAE and the enhanced classifier are specially trained using samples tampered with Gaussian noise and adversarial perturbations crafted with the technique we proposed in Chapter 4. Experimental results showed that our proposed technique significantly diminishes the accuracy reduction caused by adversarial attacks on modulation classifiers, and outperforms other protection techniques by at least 18 percentage

points. Therefore, this study outlines an exciting direction for developing effective defense techniques that protect and secure the reliability of ML-based systems.

6.2 Future Work

This section presents future research paths that we consider worth pursuing, drawing from the results obtained in this thesis.

6.2.1 Diffusion-Based Intrusion Detection

In recent years, generative models have been shown to implicitly model systems in various application domains very successfully [Freitas de Araujo-Filho *et al.* (2021)]. For instance, we showed in Chapters 2 and 3 that GANs can successfully model sensor measurements and network flows to detect cyber-attacks. On the other hand, more recently, diffusion models have been gaining interest due to their training stability, which is usually challenging for GANs, and their ability to produce image samples with higher quality than other generative models [Ho, Jonathan and Saharia, Chitwan; Kong, Ping, Huang, Zhao & Catanzaro (2020)]. Diffusion models progressively corrupt training data by adding Gaussian noise, slowly removing data samples' details until there is only noise left. Then, they train a neural network to reverse the corruption process as if it were denoising a pure noise sample until a meaningful data sample is produced [Ho, Jonathan and Saharia, Chitwan; Sohl-Dickstein, Weiss, Maheswaranathan & Ganguli (2015)]. Therefore, since diffusion models have been shown to model systems better and with higher stability than GANs, a future research path is to explore diffusion models for the detection of intrusions. Furthermore, since they are designed to reconstruct samples by removing noise from them, another exciting research path to explore is whether diffusion models can remove adversarial perturbations and thus increase the resistance of ML-based systems against adversarial attacks.

6.2.2 Minimization of the Number of Training Data Required by Attack Classifiers

Since different cyber-attacks might be mitigated in different ways and intrusions detected by anomaly-based IDSs might represent systems malfunctioning rather than attacks, it is necessary to classify intrusions once they are detected. However, due to the lack of labeled data, which is challenging and expensive to obtain, the lack of occurrences of newly identified attacks, and the need to retrain attack classifiers every time a new type of attack is identified, it is necessary to investigate techniques for minimizing the number of required training data. Recent studies propose using transfer learning and few-shot learning to achieve such a goal [Singla, Bertino & Verma (2019); Ren *et al.* (2018)].

Transfer learning can reduce the number of required training data by leveraging previously trained models. Thus, models trained on domains with more data available, e.g., Wi-Fi networks, can be used to minimize the need for data in domains with fewer data available, such as 5G networks [Singla *et al.* (2019)]. Few-shot learning, on the other hand, can recognize new classes given only a few examples from each of those classes. This may significantly reduce the retraining burden when new types of attacks are discovered and must be included in attack classifiers [Ren *et al.* (2018)]. However, such works on transfer learning and few-shot learning are still preliminary studies, and more investigation is required.

6.2.3 Security and Privacy of Digital Twins

Digital twins is an emerging concept based on creating virtual replicas of physical objects, such as jet engines, wind farms, autonomous vehicles, and even whole smart cities. Its goal is to use real-world data to simulate and predict the behavior of systems, thus preventing costly failures in physical objects [Wu, Zhang & Zhang (2021)]. Such technology brings great opportunities in several domains. In 6G, for example, digital twins are being explored to improve spectral and energy efficiency while enabling innovative applications, such as autonomous driving [Wu *et al.*

(2021)]. On the other hand, data transmission between physical objects and their replicas raises severe security and privacy issues, as tampering with data and data leaks might cause significant and undesirable damage and financial losses. Although recent works have been exploring federated learning for securing digital twins while preserving data privacy [Lu, Huang, Zhang, Maharjan & Zhang (2021)], those are still preliminary studies. Therefore, further investigation is necessary to ensure the security and privacy of digital twins.

6.2.4 Adversarial Attacks and Defenses on Regression-Based Applications

While adversarial attacks are being recently exploited in classification-based wireless communication tasks, such as modulation classification [Freitas de Araujo-Filho *et al.* (2022); Sahay *et al.* (2022)], only very few works currently exist on regression-based wireless communication tasks. Thus, the impact that adversarial attacks can cause on applications that rely on regression-based ML models, such as resource allocation, is yet to be further evaluated. Moreover, since adversarial perturbations for regression-based applications may be crafted very differently from those of classification-based applications, the question of whether defense techniques of classification problems can be effectively applied to regression tasks needs to be addressed. Therefore, it is necessary to conduct such an investigation and propose defense techniques specifically designed for regression tasks.

APPENDIX I

APPENDIX OF CHAPTER 2

1. Deployment Architecture

Figure-A I-1 shows the deployment of our proposed IDS on multiple edge servers.

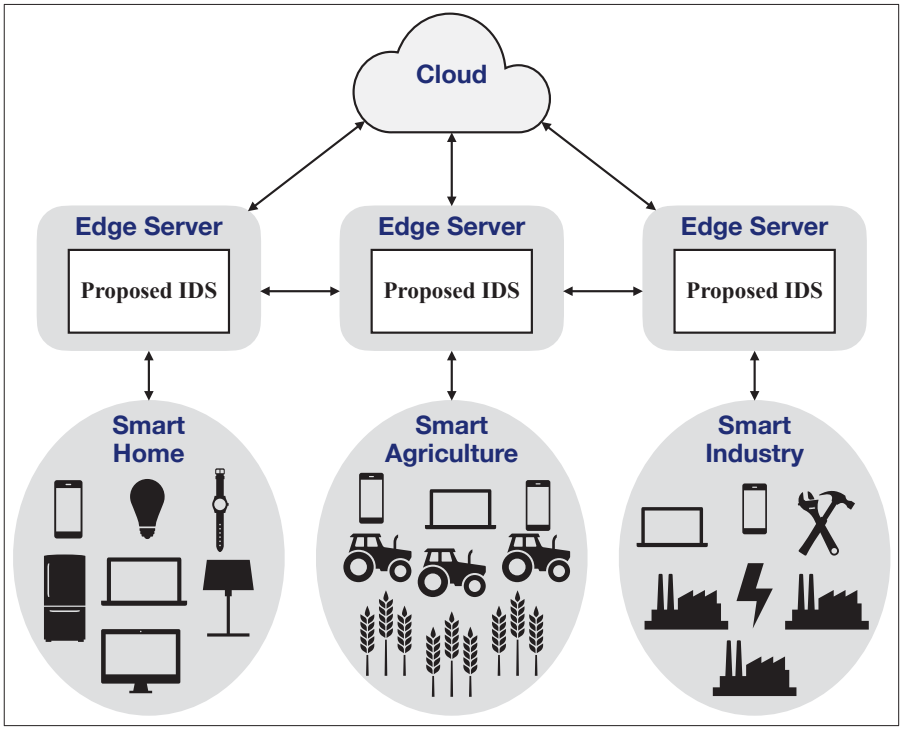


Figure-A I-1 Proposed deployment architecture

2. Features Used

Table-A I-1 lists the features used in our work.

Table-A I-1 Features used

Feature	**Description**
Source IP	Flow source IP
Source Port	Flow source port

Destination IP	Flow destination IP
Destination Port	Flow destination port
Protocol	Flow protocol
Flow duration	Flow duration in microseconds
Total Fwd Packet	Total packets in the forward direction
Total Bwd packets	Total packets in the backward direction
Total Length of Fwd Packet	Total size of packet in the forward direction
Total Length of Bwd Packet	Total size of packet in the backward direction
Fwd Packet Length Max	Maximum size of packet in the forward direction
Fwd Packet Length Min	Minimum size of packet in the forward direction
Fwd Packet Length Mean	Mean size of packet in the forward direction
Fwd Packet Length Std	Standard deviation size of packet in the forward direction
Bwd Packet Length Max	Maximum size of packet in the backward direction
Bwd Packet Length Min	Minimum size of packet in the backward direction
Bwd Packet Length Mean	Mean size of packet in the backward direction
Bwd Packet Length Std	Standard deviation size of packet in the backward direction
Flow Byte/s	Number of flow bytes per second
Flow Packets/s	Number of flow packets per second
Fwd Packets/s	Number of forward packets per second
Bwd Packets/s	Number of backward packets per second
Flow IAT Max	Maximum time between two packets sent in the flow
Flow IAT Min	Minimum time between two packets sent in the flow
Flow IAT Mean	Mean time between two packets sent in the flow
Flow IAT Std	Standard deviation time between two packets sent in the flow
Fwd IAT Max	Maximum time between two packets sent in the forward direction
Fwd IAT Min	Minimum time between two packets sent in the forward direction
Fwd IAT Mean	Mean time between two packets sent in the forward direction

Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
Fwd IAT Total	Total time between two packets sent in the forward direction
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Bwd IAT Min	Minimum time between two packets sent in the backward direction
Bwd IAT Mean	Mean time between two packets sent in the backward direction
Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
Bwd IAT Total	Total time between two packets sent in the backward direction
Fwd PSH flag	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
Bwd PSH Flag	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
Fwd Header Length	Total bytes used for headers in the forward direction
Bwd Header Length	Total bytes used for headers in the backward direction

3. Hyper-Parameter Values

Table-A I-2 lists the hyper-parameter values used in our work.

Table-A I-2 Hyper-parameter values

	LSTM	**TCN (N=1)**	**TCN (N=2)**	**Self-Attention (N=1)**
Maximum number of epochs	50	50	50	50
Early Stopping Patience	15	15	15	15
Optimizer	Adam	Adam	Adam	Adam
Discriminator's Learning Rate	0.00284252	0.00485687	0.00753606	0.01156386
Generator's Learning Rate	0.00004508	0.00015387	0.00000151	0.00003940
Dropout	0.25	0.25	0.25	0.20
Batch Size	64	64	64	64
Latent Dimension	20	10	100	10
LSTM Hidden Dimension	20	-	-	-
TCN Number of Levels	-	1	1	-
TCN Kernel Size	-	2	2	-
Attention Heads	-	-	-	40
N	-	1	2	1

AUTHOR'S PUBLICATIONS

During the course of his Ph.D. research, the author contributed to the following published and submitted research articles.

Main contributions

P. Freitas de Araujo-Filho, G. Kaddoum, M. C. B. Nasr, H. F. Arcoverde, and D. R. Campelo, "Defending Wireless Receivers Against Adversarial Attacks on Modulation Classifiers", submitted to *IEEE Internet of Things Journal*.

P. Freitas de Araujo-Filho, M. Naili, G. Kaddoum, E. T. Fapi and Z. Zhu, "Unsupervised GAN-Based Intrusion Detection System Using Temporal Convolutional Networks and Self-Attention", in *IEEE Transactions on Network and Service Management*, doi: 10.1109/TNSM.2023.3260039.

P. Freitas de Araujo-Filho, G. Kaddoum, M. Naili, E. T. Fapi and Z. Zhu, "Multi-Objective GAN-Based Adversarial Attack Technique for Modulation Classifiers", in *IEEE Communications Letters*, April, 2022, doi: 10.1109/LCOMM.2022.3167368.

P. Freitas De Araujo-Filho, A. J. Pinheiro, G. Kaddoum, D. R. Campelo and F. L. Soares, "An Efficient Intrusion Prevention System for CAN: Hindering Cyber-Attacks With a Low-Cost Platform", in *IEEE Access*, vol. 9, pp. 166855-166869, 2021, doi: 10.1109/ACCESS.2021.3136147.

P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macêdo and C. Zanchettin, "Intrusion Detection for Cyber–Physical Systems Using Generative Adversarial Networks in Fog Environment", in *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247-6256, April 15, 2021, doi: 10.1109/JIOT.2020.3024800.

Collaborations

P. Illy, G. Kaddoum, **P. F. de Araujo-Filho**, K. Kaur and S. Garg, "A Hybrid Multistage DNN-Based Collaborative IDPS for High-Risk Smart Factory Networks," in *IEEE Transactions on Network and Service Management*, 2022, doi: 10.1109/TNSM.2022.3202801.

M. -T. Nguyen, G. Kaddoum, B. Selim, K. V. Srinivas and **P. F. De Araujo-Filho**, "Deep Unfolding Network for PAPR Reduction in Multi-Carrier OFDM Systems," in *IEEE Communications Letters*, 2022, doi: 10.1109/LCOMM.2022.3195042.

A. J. Pinheiro, **P. Freitas de Araujo-Filho**, J. de M. Bezerra and D. R. Campelo, "Adaptive Packet Padding Approach for Smart Home Networks: A Tradeoff Between Privacy and Performance", in *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3930-3938, March 1, 2021, doi: 10.1109/JIOT.2020.3025988.

Luigi F. Marques da Luz, **P. Freitas de Araujo-Filho**, D. R. Campelo, "Multi-Criteria Optimized Deep Learning-based Intrusion Detection System for Detecting Cyberattacks in Automotive Ethernet Networks", accepted in Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Brasília, 2023.

P. do Carmo, **P. Freitas de Araujo-Filho**, D. R. Campelo, E. Freitas, D. Sadok, "Machine Learning-Based Intrusion Detection System for Automotive Ethernet: Detecting Cyber-Attacks with a Low-Cost Platform", in *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Fortaleza, 2022, DOI: https://doi.org/10.5753/sbrc.2022.222153.

L. Prado D'Andrada, **P. Freitas de Araujo-Filho**, and D. R. Campelo, "A Real-time Anomaly-based Intrusion Detection System for Automotive Controller Area Networks", in *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Rio de Janeiro, 2020, pp. 658-671, doi: https://doi.org/10.5753/sbrc.2020.12316.

BIBLIOGRAPHY

Aazam, M., Zeadally, S. & Harras, K. A. (2018). Deploying Fog Computing in Industrial Internet of Things and Industry 4.0. *IEEE Trans. on Ind. Inform.*, 14(10), 4674–4682.

Abeshu, A. & Chilamkurti, N. (Feb, 2018). Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing. *IEEE Commun. Mag.*, 56(2), 169–175.

Ahlashkari. CICFlowMeter. Retrieved from: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter.

Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proce. of the 25rd ACM SIGKDD Int. Conf. on Knowl. Discovery and Data Mining*.

Alguliyev, R., Imamverdiyev, Y. & Sukhostat, L. (2018). Cyber-physical systems and their security issues. *Comput. in Industry*, 100, 212–223.

Ali, S., Saad, W., Rajatheva, N., Chang, K., Steinbach, D., Sliwa, B., Wietfeld, C., Mei, K., Shiri, H., Zepernick, H.-J. et al. (2020). 6G White Paper on Machine Learning in Wireless Communication Networks. *arXiv preprint arXiv:2004.13875*.

Amazon Web Services. What is a DDoS Attack? Retrieved from: https://aws.amazon.com/shield/ddos-attack-protection/.

Amidi, A. & Amidi, S. CS 230 Deep Learning - Convolutional Neural Networks Cheatsheet. Retrieved from: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks.

An, X., Zhou, X., Lü, X., Lin, F. & Yang, L. (2018). Sample Selected Extreme Learning Machine Based Intrusion Detection in Fog Computing and MEC. *Wireless Commun. and Mobile Comput.*, 2018.

Apache Kafka. Apache Kafka. Retrieved from: https://kafka.apache.org/.

Arjovsky, M. & Bottou, L. (2017). Towards Principled Methods for Training Generative Adversarial Networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M., Chintala, S. & Bottou, L. (2017). Wasserstein Generative Adversarial Networks. *Int. Conf. on Machine Learning*, pp. 214–223.

Avatefipour, O., Al-Sumaiti, A. S., El-Sherbeeny, A. M., Awwad, E. M., Elmeligy, M. A., Mohamed, M. A. & Malik, H. (2019). An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning. *IEEE Access*, 7, 127580–127592.

Bach, F. & Chizat, L. (2021). Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization. *arXiv preprint arXiv:2110.08084*.

Bai, S., Kolter, J. Z. & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Baza, M., Sherif, A., Mahmoud, M. M. E. A., Bakiras, S., Alasmary, W., Abdallah, M. & Lin, X. (2021). Privacy-Preserving Blockchain-Based Energy Trading Schemes for Electric Vehicles. *IEEE Trans. on Veh. Technol.*, 70(9), 9369-9384. doi: 10.1109/TVT.2021.3098188.

Brendel, W., Rauber, J. & Bethge, M. (2017). Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. *arXiv preprint arXiv:1712.04248*.

Canadian Institute for Cybersecurity. NSL-KDD Dataset. Retrieved from: https://www.unb.ca/cic/datasets/nsl.html.

Carnero, A., Martín, C., Torres, D. R., Garrido, D., Díaz, M. & Rubio, B. (2021). Managing and Deploying Distributed and Deep Neural Models Through Kafka-ML in the Cloud-to-Things Continuum. *IEEE Access*, 9, 125478-125495. doi: 10.1109/ACCESS.2021.3110291.

Caruana, R., Lawrence, S. & Giles, C. (2000). Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. *Advances in neural information processing systems*, 13.

Casanova, A., Careil, M., Verbeek, J., Drozdzal, M. & Romero Soriano, A. (2021). Instance-Conditioned GAN. *Advances in Neural Inf. Process. Syst.*, 34, 27517–27529.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C. & Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surveys Tut.*, 21(3), 2671-2701. doi: 10.1109/COMST.2019.2896380.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A. & Mukhopadhyay, D. (2018). Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Chang, H., Nguyen, T. D., Murakonda, S. K., Kazemi, E. & Shokri, R. (2020). On Adversarial Bias and the Robustness of Fair Machine Learning. *arXiv preprint arXiv:2006.08669*.

Cheng, K., Bai, Y., Zhou, Y., Tang, Y., Sanan, D. & Liu, Y. (2020). CANeleon: Protecting CAN Bus with Frame ID Chameleon. *IEEE Trans. on Veh. Technol.*, 1-1.

Choi, H., Kim, M., Lee, G. & Kim, W. (Sep, 2019). Unsupervised learning approach for network intrusion detection system using autoencoders. *Springer J. of Supercomput.*, 75(9), 5597–5621.

Chowdhury, M. M. U., Hammond, F., Konowicz, G., Xin, C., Wu, H. & Li, J. (2017). A few-shot deep learning approach for improved intrusion detection. *2017 IEEE 8th Annual Ubiquitous Comput., Electronics and Mobile Commun. Conf. (UEMCON)*, pp. 456–462.

Cloudfare. Famous DDoS attacks | The largest DDoS attacks of all time. Retrieved from: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/.

Creswell, A. & Bharath, A. A. (Jul, 2018). Inverting the Generator of a Generative Adversarial Network. *IEEE Trans. on Neural Netw. and Learn. Syst.*, 30(7), 1967–1974.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018). Generative Adversarial Networks: An overview. *IEEE Signal Process. Mag.*, 35(1), 53–65.

da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R. & de Albuquerque, V. H. C. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. *Comput. Netw.*, 151, 147–157.

de Almeida, I. B. F., Mendes, L. L., Rodrigues, J. J. P. C. & da Cruz, M. A. A. (2019). 5G Waveforms for IoT Appl. *IEEE Commun. Surv. Tut.*, 21(3), 2554-2567. doi: 10.1109/COMST.2019.2910817.

Developers, G. Loss Functions. Retrieved from: https://developers.google.com/machine-learning/gan/loss.

Ding, D., Han, Q.-L., Xiang, Y., Ge, X. & Zhang, X.-M. (2018). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275, 1674–1683.

Diro, A. A. & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.*, 82, 761–768.

Duc, T. N., Minh, C. T., Xuan, T. P. & Kamioka, E. (2020). Convolutional Neural Networks for Continuous QoE Prediction in Video Streaming Services. *IEEE Access*, 8, 116268-116278. doi: 10.1109/ACCESS.2020.3004125.

Feng, C., Li, T. & Chana, D. (2017). Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. *2017 47th Annual IEEE/IFIP Int. Conf. on Dependable Syst. and Netw. (DSN)*, pp. 261–272.

Ferdowsi, A. & Saad, W. (2019). Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things. *2019 IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-6. doi: 10.1109/GLOBECOM38437.2019.9014102.

Fernández Maimó, L., Perales Gómez, L., García Clemente, F. J., Gil Pérez, M. & Martínez Pérez, G. (2018). A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. *IEEE Access*, 6, 7700-7712. doi: 10.1109/ACCESS.2018.2803446.

Finney, M. N. K. (2014). Cybersecurity and Cyberwar: What Everyone Needs to Know. *Parameters*, 44(3), 149.

Flores, S. Variational Autoencoders are Beautiful. Retrieved from: https://www.compthree.com/blog/autoencoder/.

Flowers, B., Buehrer, R. M. & Headley, W. C. (2020). Evaluating Adversarial Evasion Attacks in the Context of Wireless Communications. *IEEE Trans. on Inf. Forensics and Secur.*, 15, 1102-1113. doi: 10.1109/TIFS.2019.2934069.

for Cybersecurity, C. I. DDoS Evaluation Dataset (CIC-DDoS2019). Retrieved from: https://www.unb.ca/cic/datasets/ddos-2019.html.

Freitas de Araujo-Filho, P., Kaddoum, G., Campelo, D. R., Gondim Santos, A., Macêdo, D. & Zanchettin, C. (2021). Intrusion Detection for Cyber–Physical Systems Using Generative Adversarial Networks in Fog Environment. *IEEE Internet of Things J.*, 8(8), 6247-6256. doi: 10.1109/JIOT.2020.3024800.

Freitas de Araujo-Filho, P., Kaddoum, G., Naili, M., Fapi, E. T. & Zhu, Z. (2022). Multi-Objective GAN-Based Adversarial Attack Technique for Modulation Classifiers. *IEEE Commun. Lett.*, 1-1. doi: 10.1109/LCOMM.2022.3167368.

Freitas de Araujo-Filho, P., Naili, M., Kaddoum, G., Fapi, E. T. & Zhu, Z. (2023). Unsupervised GAN-Based Intrusion Detection System Using Temporal Convolutional Networks and Self-Attention. *IEEE Trans. on Netw. and Service Manage.*, 1-1. doi: 10.1109/TNSM.2023.3260039.

Gallenmüller, S., Naab, J., Adam, I. & Carle, G. (2020). 5G URLLC: A Case Study on Low-Latency Intrusion Prevention. *IEEE Commun. Mag.*, 58(10), 35-41. doi: 10.1109/M-COM.001.2000467.

Garcia, S., Grill, M., Stiborek, J. & Zunino, A. (2014). An empirical comparison of botnet detection methods. *Comp. & Secur.*, 45, 100–123.

Garg, S., Kaur, K., Batra, S., Kaddoum, G., Kumar, N. & Boukerche, A. (2020). A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications. *Future Gener. Comput. Syst.*, 104, 105–118.

Ghafir, I., Kyriakopoulos, K. G., Aparicio-Navarro, F. J., Lambotharan, S., Assadhan, B. & Binsalleeh, H. (2018). A Basic Probability Assignment Methodology for Unsupervised Wireless Intrusion Detection. *IEEE Access*, 6, 40008-40023. doi: 10.1109/ACCESS.2018.2855078.

Gingras, B., Pourranjbar, A. & Kaddoum, G. (2020). Collaborative Spectrum Sensing in Tactical Wireless Networks. *ICC 2020 - 2020 IEEE Int. Conf. on Commun. (ICC)*, pp. 1-6. doi: 10.1109/ICC40277.2020.9149223.

Goh, J., Adepu, S., Tan, M. & Lee, Z. S. (2017). Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. *IEEE 18th Int. Symp. on High Assurance Syst. Eng. (HASE)*, pp. 140–145.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (pp. 2672–2680). Curran Associates, Inc. Retrieved from: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. Cambridge, MA, USA: MIT press.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. (2017). Improved Training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*.

Hachimi, M., Kaddoum, G., Gagnon, G. & Illy, P. (2020). Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks. *2020 Int. Symp. on Netw., Comput. and Commun. (ISNCC)*, pp. 1-5. doi: 10.1109/ISNCC49221.2020.9297290.

Han, S., Xie, M., Chen, H. & Ling, Y. (2014). Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges. *IEEE Syst. J.*, 8(4), 1052-1062. doi: 10.1109/JSYST.2013.2257594.

Harrou, F., Nounou, M. N., Nounou, H. N. & Madakyaru, M. (Jul, 2015). PLS-based EWMA fault detection strategy for process monitoring. *Elsevier J. of Loss Prevention in the Process Ind.*, 36, 108–119.

He, Y. & Zhao, J. (2019). Temporal convolutional networks for anomaly detection in time series. *J. of Physics: Conf. Series*, 1213(4), 042050.

Ho, Jonathan and Saharia, Chitwan. High Fidelity Image Generation Using Diffusion Models. Retrieved from: https://ai.googleblog.com/2021/07/high-fidelity-image-generation-using.html.

Hollis, T., Viscardi, A. & Yi, S. E. (2018). A comparison of LSTMs and attention mechanisms for forecasting financial time series. *arXiv preprint arXiv:1812.07699*.

Hu, P., Ning, H., Qiu, T., Song, H., Wang, Y. & Yao, X. (2017). Security and Privacy Preservation Scheme of Face Identification and Resolution Framework Using Fog Computing in Internet of Things. *IEEE Internet of Things J.*, 4(5), 1143–1155.

Huang, S., Liu, Y., Fung, C., He, R., Zhao, Y., Yang, H. & Luan, Z. (2020). HitAnomaly: Hierarchical Transformers for Anomaly Detection in System Log. *IEEE Trans. on Netw. and Service Manage.*, 17(4), 2064-2076. doi: 10.1109/TNSM.2020.3034647.

Ibitoye, O., Abou-Khamis, R., Matrawy, A. & Shafiq, M. O. (2019). The Threat of Adversarial Attacks on Machine Learning in Network Security - A Survey. *arXiv preprint arXiv:1911.02621*.

Illy, P., Kaddoum, G., Miranda Moreira, C., Kaur, K. & Garg, S. (2019). Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning. *2019 IEEE Wireless Commun. and Netwo. Conf. (WCNC)*, pp. 1-7. doi: 10.1109/WCNC.2019.8885534.

Illy, P., Kaddoum, G., Freitas de Araujo-Filho, P., Kaur, K. & Garg, S. (2022). A Hybrid Multistage DNN-Based Collaborative IDPS for High-Risk Smart Factory Networks. *IEEE Trans. on Netw. and Service Manage.*, 19(4), 4273-4283. doi: 10.1109/TNSM.2022.3202801.

Ilyas, A., Engstrom, L., Athalye, A. & Lin, J. (2018). Black-box adversarial attacks with limited queries and information. *Int. Conf. on Mach. Learn.*, pp. 2137–2146.

Injadat, M., Moubayed, A. & Shami, A. (2020). Detecting Botnet Attacks in IoT Environments: An Optimized Machine Learning Approach. *2020 32nd Int. Conf. on Microelectronics (ICM)*, pp. 1-4. doi: 10.1109/ICM50269.2020.9331794.

iTrust Singapore University of Technology and Design (SUTD). The Secure Water Treatment (SWaT). Retrieved from: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/.

iTrust Singapore University of Technology and Design (SUTD). The Water Distribution (WADI). Retrieved from: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/.

Jia, Y., Zhong, F., Alrawais, A., Gong, B. & Cheng, X. (2020). FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks. *IEEE Internet of Things J.*, 7(10), 9552-9562. doi: 10.1109/JIOT.2020.2993782.

Kaiser, L. (2017). Tensor2Tensor Transformers New Deep Models for NLP.

Kekki, S., Featherstone, W., Fang, Y., Kuure, P., Li, A., Ranjan, A., Purkayastha, D., Jiangping, F., Frydman, D., Verin, G. et al. (2018). MEC in 5G networks. *ETSI White Paper*, 28, 1–28.

Kendall, A., Gal, Y. & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit.*, pp. 7482–7491.

Kim, B., Sagduyu, Y. E., Davaslioglu, K., Erpek, T. & Ulukus, S. (2021). Channel-Aware Adversarial Attacks Against Deep Learning-Based Wireless Signal Classifiers. *IEEE Trans. on Wireless Commun.*, 1-1. doi: 10.1109/TWC.2021.3124855.

Kong, Z., Ping, W., Huang, J., Zhao, K. & Catanzaro, B. (2020). DiffWave: A Versatile Diffusion Model for Audio Synthesis. *arXiv preprint arXiv:2009.09761*.

Kravchik, M. & Shabtai, A. (2021). Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA. *IEEE Trans. on Dependable and Secure Comput.*, 1-1. doi: 10.1109/TDSC.2021.3050101.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H. & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. *Int. Conf. on Machine Learning*, pp. 1558–1566.

Li, D., Chen, D., Goh, J. & Ng, S.-k. (2018a). Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. *arXiv:1809.04758*.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J. & Ng, S.-K. (2019). MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. *Springer Int. Conf. on Artif. Neural Netw.*, pp. 703–716.

Li, J., Jin, J., Yuan, D. & Zhang, H. (2017). Virtual Fog: A Virtualization Enabled Fog Computing Framework for Internet of Things. *IEEE Internet of Things J.*, 5(1), 121–131.

Li, K., Yu, X., Zhang, H., Wu, L., Du, X., Ratazzi, P. & Guizani, M. (2018b). Security Mechanisms to Defend against New Attacks on Software-Defined Radio. *2018 Int. Conf. on Comput., Netw. and Commun. (ICNC)*, pp. 537-541. doi: 10.1109/ICCNC.2018.8390381.

Li, S., Da Xu, L. & Zhao, S. (2018c). 5G Internet of Things: A survey. *J. of Ind. Inf. Integration*, 10, 1–9.

Li, S. & Wen, J. (Jan, 2014). A model-based fault detection and diagnostic methodology based on PCA method and wavelet transform. *Elsevier Energy and Buildings*, 68, Part A, 63–71.

Li, Y., Zhang, L., Lv, Z. & Wang, W. (2021). Detecting Anomalies in Intelligent Vehicle Charging and Station Power Supply Systems With Multi-Head Attention Models. *IEEE Trans. on Intell. Transp. Syst.*, 22(1), 555-564. doi: 10.1109/TITS.2020.3018259.

Liang, F., Shen, C. & Wu, F. (2018). An Iterative BP-CNN Architecture for Channel Decoding. *IEEE J. of Sel. Topics in Signal Process.*, 12(1), 144-159. doi: 10.1109/JSTSP.2018.2794062.

Lin, Y., Zhao, H., Tu, Y., Mao, S. & Dou, Z. (2020). Threats of Adversarial Attacks in DNN-Based Modulation Recognition. *IEEE INFOCOM 2020 - IEEE Conf. on Comput. Commun.*, pp. 2469-2478. doi: 10.1109/INFOCOM41043.2020.9155389.

Lin, Y., Zhao, H., Ma, X., Tu, Y. & Wang, M. (2021). Adversarial Attacks in Modulation Recognition With Convolutional Neural Networks. *IEEE Trans. on Rel.*, 70(1), 389-401. doi: 10.1109/TR.2020.3032744.

Liu, J., Nogueira, M., Fernandes, J. & Kantarci, B. (2022). Adversarial Machine Learning: A Multi-Layer Review of the State-of-the-Art and Challenges for Wireless and Mobile Systems. *IEEE Commun. Surveys Tut.*, 24(1), 123-159. doi: 10.1109/COMST.2021.3136132.

Lu, H., Li, Y., Chen, M., Kim, H. & Serikawa, S. (2018). Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Netw. Appl.*, 23(2), 368–375.

Lu, H., Zhang, M., Xu, X., Li, Y. & Shen, H. T. (2020). Deep Fuzzy Hashing Network for Efficient Image Retrieval. *IEEE Trans. on Fuzzy Syst.*

Lu, Y., Huang, X., Zhang, K., Maharjan, S. & Zhang, Y. (2021). Communication-Efficient Federated Learning and Permissioned Blockchain for Digital Twin Edge Networks. *IEEE Internet of Things J.*, 8(4), 2276-2288. doi: 10.1109/JIOT.2020.3015772.

Lucic, M., Kurach, K., Michalski, M., Gelly, S. & Bousquet, O. (2017). Are GANs Created Equal? A Large-Scale Study. *arXiv preprint arXiv:1711.10337*.

Manoj, B. R., Santos, P. M., Sadeghi, M. & Larsson, E. G. (2022). Toward Robust Networks Against Adversarial Attacks for Radio Signal Modulation Classification. *2022 IEEE 23rd Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, pp. 1-5. doi: 10.1109/SPAWC51304.2022.9833926.

Manoj, B., Sadeghi, M. & Larsson, E. G. (2021). Adversarial Attacks on Deep Learning Based Power Allocation in a Massive MIMO Network. *arXiv preprint arXiv:2101.12090*.

Martín, C., Langendoerfer, P., Zarrin, P. S., Díaz, M. & Rubio, B. (2022). Kafka-ML: Connecting the data stream with ML/AI frameworks. *Future Gener. Comput. Syst.*, 126, 15–33.

Meftah, A., Kaddoum, G., Do, T. N. & Talhi, C. (2022). Federated Learning-Based Jamming Detection for Distributed Tactical Wireless Networks. *MILCOM 2022 - 2022 IEEE Mil. Commun. Conf. (MILCOM)*, pp. 629-634. doi: 10.1109/MILCOM55135.2022.10017755.

Midi, D., Rullo, A., Mudgerikar, A. & Bertino, E. (2017). Kalis — A System for Knowledge-Driven Adaptable Intrusion Detection for the Internet of Things. *2017 IEEE 37th Int. Conf. on Distrib. Comput. Syst. (ICDCS)*, pp. 656-666. doi: 10.1109/ICDCS.2017.104.

Miranda, C., Kaddoum, G., Boukhtouta, A., Madi, T. & Alameddine, H. A. (2022). Intrusion Prevention Scheme Against Rank Attacks for Software-Defined Low Power IoT Networks. *IEEE Access*, 10, 129970-129984. doi: 10.1109/ACCESS.2022.3228170.

Mitchell, R. & Chen, I.-R. (Apr, 2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv. (CSUR)*, 46(4), 1–29.

Mittal, A. Understanding RNN and LSTM. Retrieved from: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

Moati, N., Otrok, H., Mourad, A. & Robert, J.-M. (2014). Reputation-Based Cooperative Detection Model of Selfish Nodes in Cluster-Based QoS-OLSR Protocol. *Wireless Pers. Commun.*, 75(3), 1747–1768.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O. & Frossard, P. (2017, July). Universal Adversarial Perturbations. *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit. (CVPR)*.

Moubayed, A., Injadat, M. & Shami, A. (2020). Optimized Random Forest Model for Botnet Detection Based on DNS Queries. *2020 32nd Int. Conf. on Microelectronics (ICM)*, pp. 1-4. doi: 10.1109/ICM50269.2020.9331819.

Mourad, A., Laverdiere, M.-A. & Debbabi, M. (2007). Towards an Aspect Oriented Approach for the Security Hardening of Code. *21st Int. Conf. on Adv. Inf. Netw. and Appl. Workshops (AINAW'07)*, 1, 595–600.

Mourad, A., Laverdière, M.-A. & Debbabi, M. (2008). A High-level Aspect-oriented-based Framework for Software Security Hardening. *Inf. Secur. J.: A Global Perspective*, 17(2), 56–74.

Naeem, F., Ali, M. & Kaddoum, G. (2023). Federated-Learning-Empowered Semi-Supervised Active Learning Framework for Intrusion Detection in ZSM. *IEEE Commun. Mag.*, 61(2), 88-94. doi: 10.1109/MCOM.001.2200533.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G. & Ghani, N. (2019). Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Commun. Surveys & Tut.*, 21(3), 2702–2733.

Nguyen, M.-T., Kaddoum, G., Selim, B., Srinivas, K. V. & Freitas de Araujo-Filho, P. (2022). Deep Unfolding Network for PAPR Reduction in Multicarrier OFDM Systems. *IEEE Commun. Lett.*, 26(11), 2616-2620. doi: 10.1109/LCOMM.2022.3195042.

Nicholson, P. Five Most Famous DDoS Attacks and Then Some. Retrieved from: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/.

Nisioti, A., Mylonas, A., Yoo, P. D. & Katos, V. (2018). From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods. *IEEE Commun. Surveys & Tut.*, 20(4), 3369-3388. doi: 10.1109/COMST.2018.2854724.

Olufowobi, H., Young, C., Zambreno, J. & Bloom, G. (2019). Saiducant: Specification-based automotive intrusion detection using controller area network (can) timing. *IEEE Trans. on Veh. Technol.*, 69(2), 1484–1494.

O'Shea, T. J. & West, N. (2016). Radio Machine Learning Dataset Generation with GNU Radio. *Proc. of the 6th GNU Radio Conf.*

Osseiran, A., Monserrat, J. F. & Marsch, P. (2016). *5G mobile and wireless communications technology*. Cambridge University Press.

Ozgumus, S. Y. (2019). *Adversarially learned anomaly detection using generative adversarial networks*. (Ph.D. thesis, Politecnico di Milano, Milan, Italy). Retrieved from: http://hdl.handle.net/10589/149395.

O'Shea, T. J., Corgan, J. & Clancy, T. C. (2016). Convolutional radio modulation recognition networks. *Int. Conf. on Eng. Appl. of Neural Networks*, pp. 213–226.

O'Shea, T. J., Roy, T. & Clancy, T. C. (2018). Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE J. of Sel. Topics in Signal Process.*, 12(1), 168-179. doi: 10.1109/JSTSP.2018.2797022.

Papamartzivanos, D., Gómez Mármol, F. & Kambourakis, G. (2019). Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. *IEEE Access*, 7, 13546-13560. doi: 10.1109/ACCESS.2019.2893871.

Peng, L., Yang, Y., Zhang, X., Ji, Y., Lu, H. & Shen, H. T. (2020). Answer Again: Improving VQA with Cascaded-Answering Model. *IEEE Trans. on Knowl. and Data Eng*.

Pham, Q., Fang, F., Ha, V. N., Piran, M. J., Le, M., Le, L. B., Hwang, W. & Ding, Z. (2020). A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art. *IEEE Access*, 8, 116974-117017. doi: 10.1109/ACCESS.2020.3001277.

Philipp, A., Cowen, D. & Davis, C. (2009). *Hacking exposed computer forensics*. New York, NY, USA: McGraw-Hill.

Pourranjbar, A., Kaddoum, G. & Aghababaiyan, K. (2022a). Deceiving-Based Anti-Jamming Against Single-Tone and Multitone Reactive Jammers. *IEEE Trans. on Commun.*, 70(9), 6133-6148. doi: 10.1109/TCOMM.2022.3192385.

Pourranjbar, A., Kaddoum, G. & Saad, W. (2022b). Recurrent Neural Network-based Anti-jamming Framework for Defense Against Multiple Jamming Policies. *IEEE Internet of Things J.*, 1-1. doi: 10.1109/JIOT.2022.3233454.

Pourranjbar, A., Elleuch, I., Landry-pellerin, S. & Kaddoum, G. (2023). Defense and Offence Strategies for Tactical Wireless Networks Using Recurrent Neural Networks. *IEEE Trans. on Veh. Technol.*, 1-6. doi: 10.1109/TVT.2023.3243127.

Prabavathy, S., Sundarakantham, K. & Shalinie, S. M. (2018). Design of cognitive fog computing for intrusion detection in Internet of Things. *J. of Commun. and Netw.*, 20(3), 291-298. doi: 10.1109/JCN.2018.000041.

Praseed, A. & Thilagam, P. S. (2022). HTTP request pattern based signatures for early application layer DDoS detection: A firewall agnostic approach. *J. of Inf. Secur. and Appl.*, 65, 103090.

Prechelt, L. (1998). Early Stopping-but when? In *Neural Netw.: Tricks of the trade* (pp. 55–69). Springer.

Puzanov, A. & Cohen, K. (2018). Deep reinforcement one-shot learning for artificially intelligent classification systems. *arXiv preprint arXiv:1808.01527*.

R. Dobbins and S. Bjarnason. Fancy Lazarus DDoS Extortion Campaign. Retrieved from: https://www.netscout.com/blog/asert/fancy-lazarus-ddos-extortion-campaign.

Redana, S., Bulakci, Ö., Zafeiropoulos, A., Gavras, A., Tzanakaki, A., Albanese, A., Kousaridas, A., Weit, A., Sayadi, B., Jou, B. T. et al. (2019). 5G PPP Architecture Working Group: View on 5G Architecture.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H. & Zemel, R. S. (2018). Meta-Learning for Semi-Supervised Few-Shot Classification. *arXiv preprint arXiv:1803.00676*.

Rodriguez, J. (2015). *Fundamentals of 5G mobile networks*. John Wiley & Sons.

Roland Dobbins and Steinthor Bjarnason. High-Profile DDoS Extortion Attacks Against SIP/RTP VoIP Providers. Retrieved from: https://www.netscout.com/blog/asert/high-profile-ddos-extortion-attacks-against-siprtp-voip.

Ronen, B., Jacobs, D., Kasten, Y. & Kritchman, S. (2019). The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies. *Advances in Neural Inf. Process. Syst.*, 32.

Saad, W., Bennis, M. & Chen, M. (2020). A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems. *IEEE Netw.*, 34(3), 134-142. doi: 10.1109/MNET.001.1900287.

Sadeghi, M. & Larsson, E. G. (2019). Adversarial Attacks on Deep-Learning Based Radio Signal Classification. *IEEE Wireless Commun. Lett.*, 8(1), 213-216. doi: 10.1109/LWC.2018.2867459.

Sahay, R., Love, D. J. & Brinton, C. G. (2021). Robust Automatic Modulation Classification in the Presence of Adversarial Attacks. *2021 55th Annu. Conf. on Inf. Sciences and Syst. (CISS)*, pp. 1-6. doi: 10.1109/CISS50987.2021.9400326.

Sahay, R., Brinton, C. G. & Love, D. J. (2022). A Deep Ensemble-Based Wireless Receiver Architecture for Mitigating Adversarial Attacks in Automatic Modulation Classification. *IEEE Trans. on Cognitive Commun. and Netw.*, 8(1), 71-85. doi: 10.1109/TCCN.2021.3114154.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. (2016). Improved Techniques for Training GANs. *Advances in Neural Inf. Process. Syst.*, pp. 2234–2242.

Samangouei, P., Kabkab, M. & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*.

Sanguinetti, L., Zappone, A. & Debbah, M. (2018). Deep Learning Power Allocation in Massive MIMO. *2018 52nd Asilomar Conf. on Signals, Syst., and Comput.*, pp. 1257-1261. doi: 10.1109/ACSSC.2018.8645343.

Santos, J., Leroux, P., Wauters, T., Volckaert, B. & De Turck, F. (2018). Anomaly detection for Smart City applications over 5G low power wide area networks. *NOMS 2018 - 2018 IEEE/IFIP Netw. Operations and Manage. Symp.*, pp. 1-9. doi: 10.1109/NOMS.2018.8406257.

Sayad Haghighi, M., Farivar, F. & Jolfaei, A. (2020). A Machine Learning-based Approach to Build Zero False-Positive IPSs for Industrial IoT and CPS with a Case Study on Power Grids Security. *IEEE Trans. on Industry Appl.*, 1-1. doi: 10.1109/TIA.2020.3011397.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U. & Langs, G. (2017). Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. *Springer Int. Conf. on Inf. Process. in Med. Imag. (IPMI)*, pp. 146–157.

Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G. & Schmidt-Erfurth, U. (May, 2019). f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Elsevier Med. Img. Anal.*, 54, 30–44.

Schulz, P., Matthe, M., Klessig, H., Simsek, M., Fettweis, G., Ansari, J., Ashraf, S. A., Almeroth, B., Voigt, J., Riedel, I., Puschmann, A., Mitschele-Thiel, A., Muller, M., Elste, T. & Windisch, M. (2017). Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. *IEEE Commun. Mag.*, 55(2), 70-78. doi: 10.1109/MCOM.2017.1600435CM.

Sharafaldin, I., Lashkari, A. H., Hakak, S. & Ghorbani, A. A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. *2019 Int. Carnahan Conf. on Secur. Technol. (ICCST)*, pp. 1–8.

Sharma, A., Kalbarczyk, Z., Barlow, J. & Iyer, R. (2011). Analysis of security data from a large computing organization. *2011 IEEE/IFIP 41st Int. Conf. on Dependable Syst. Netw. (DSN)*, pp. 506-517. doi: 10.1109/DSN.2011.5958263.

Shi, W., Pallis, G. & Xu, Z. (2019). Edge Computing [Scanning the Issue]. *Proc. of the IEEE*, 107(8), 1474-1481. doi: 10.1109/JPROC.2019.2928287.

Shi, Y., Davaslioglu, K. & Sagduyu, Y. E. (2019). Generative adversarial network for wireless signal spoofing. *Proc. of the ACM Workshop on Wireless Secur. and Mach. Learn.*, pp. 55–60.

Shone, N., Ngoc, T. N., Phai, V. D. & Shi, Q. (Feb, 2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Trans. on Emerg. Topics in Comput. Intell.*, 2(1), 41–50.

Shtaiwi, E., Ouadrhiri, A. E., Moradikia, M., Sultana, S., Abdelhadi, A. & Han, Z. (2022). Mixture GAN For Modulation Classification Resiliency Against Adversarial Attacks. *arXiv preprint arXiv:2205.15743*.

Silva, T. An intuitive introduction to Generative Adversarial Networks (GANs). Retrieved from: https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/.

Singla, A., Bertino, E. & Verma, D. (2019). Overcoming the Lack of Labeled Data: Training Intrusion Detection Models Using Transfer Learning. *2019 IEEE Int. Conf. on Smart Comput. (SMARTCOMP)*, pp. 69–74.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *International Conference on Machine Learning*, pp. 2256–2265.

Sriram, S., Vinayakumar, R., Alazab, M. & KP, S. (2020). Network Flow based IoT Botnet Attack Detection using Deep Learning. *IEEE INFOCOM 2020 - IEEE Conf. on Comp. Commun. Workshops (INFOCOM WKSHPS)*, pp. 189-194. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U. & Sutton, C. (2017). VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. *Advances in Neural Inf. Process. Syst.*, pp. 3308–3318.

Stanford. CS231n Convolutional Neural Networks for Visual Recognition. Retrieved from: https://cs231n.github.io/convolutional-networks/.

Strom, B. E., Battaglia, J. A., Kemmerer, M. S., Kupersanin, W., Miller, D. P., Wampler, C., Whitley, S. M. & Wolf, R. D. (2017). Finding cyber threats with ATT&CK-based analytics. *The MITRE Corporation, Tech. Rep.*

Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X. & Sidiropoulos, N. D. (2017). Learning to optimize: Training deep neural networks for wireless resource management. *2017 IEEE 18th Int. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, pp. 1-6. doi: 10.1109/SPAWC.2017.8227766.

Tan, M., Iacovazzi, A., Cheung, N. M. & Elovici, Y. (2019). A Neural Attention Model for Real-Time Network Intrusion Detection. *2019 IEEE 44th Conf. on Local Comput. Netw. (LCN)*, pp. 291-299. doi: 10.1109/LCN44214.2019.8990890.

Torres, D. R., Martín, C., Rubio, B. & Díaz, M. (2021). An open source framework based on Kafka-ML for Distributed DNN inference over the Cloud-to-Things continuum. *J. of Syst. Architecture*, 118, 102214.

Umer, M. F., Sher, M. & Bi, Y. (2017). Flow-based intrusion detection: Techniques and challenges. *Comput. & Secur.*, 70, 238–254.

Usama, M., Asim, M., Latif, S., Qadir, J. et al. (2019). Generative Adversarial Networks for Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems. *2019 15th Int. Wireless Commun. & Mobile Comput. Conf. (IWCMC)*, pp. 78–83.

Vakhter, V., Soysal, B., Schaumont, P. & Guler, U. (2022). Threat Modeling and Risk Analysis for Miniaturized Wireless Biomedical Devices. *IEEE Internet of Things J.*, 9(15), 13338-13352. doi: 10.1109/JIOT.2022.3144130.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Vigneswaran, K. R., Vinayakumar, R., Soman, K. & Poornachandran, P. (2018). Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. *IEEE 9th Int. Conf. on Comput., Commun. and Netw. Technologies (ICCCNT)*, pp. 1–6.

Wahab, O. A., Bentahar, J., Otrok, H. & Mourad, A. (2020). Optimal Load Distribution for the Detection of VM-Based DDoS Attacks in the Cloud. *IEEE Trans. on Services Comput.*, 13(1), 114-129.

Wei, X., Gong, B., Liu, Z., Lu, W. & Wang, L. (2018). Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect. *arXiv preprint arXiv:1803.01541*.

Wu, P., Guo, H. & Buckland, R. (2019). A Transfer Learning Approach for Network Intrusion Detection. *2019 IEEE 4th Int. Conf. on Big Data Analytics (ICBDA)*, pp. 281–285.

Wu, Y., Zhang, K. & Zhang, Y. (2021). Digital Twin Networks: A Survey. *IEEE Internet of Things J.*, 8(18), 13789-13804. doi: 10.1109/JIOT.2021.3079510.

Xu, X., Lin, K., Lu, H., Gao, L. & Shen, H. T. (2020). Correlated Features Synthesis and Alignment for Zero-shot Cross-modal Retrieval. *Proc. of the 43rd Int. ACM SIGIR Conf. on Res. and Develop. in Inf. Retrieval*, pp. 1419–1428.

Xu, X., Li, J., Yang, Y. & Shen, F. (2021). Toward Effective Intrusion Detection Using Log-Cosh Conditional Variational Autoencoder. *IEEE Internet of Things J.*, 8(8), 6187-6196. doi: 10.1109/JIOT.2020.3034621.

Yang, Y., Wu, L., Yin, G., Li, L. & Zhao, H. (2017). A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things J.*, 4(5), 1250-1258. doi: 10.1109/JIOT.2017.2694844.

Yousefpour, A., Ishigaki, G., Gour, R. & Jue, J. P. (2018). On Reducing IoT Service Delay via Fog Offloading. *IEEE Internet of Things J.*, 5(2), 998–1010.

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J. & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. of Syst. Architecture*, 98, 289–330.

Yuan, X., He, P., Zhu, Q. & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. on Neural Netw. and Learn. Syst.*, 30(9), 2805–2824.

Zarpelao, B. B., Miani, R. S., Kawakani, C. T. & de Alvarenga, S. C. (Apr, 2017). A survey of intrusion detection in Internet of Things. *Elsevier J. of Netw. and Comput. Appl.*, 84, 25–37.

Zavrak, S. & İskefiyeli, M. (2020). Anomaly-Based Intrusion Detection From Network Flow Features Using Variational Autoencoder. *IEEE Access*, 8, 108346-108358. doi: 10.1109/ACCESS.2020.3001350.

Zenati, H., Foo, C. S., Lecouat, B., Manek, G. & Chandrasekhar, V. R. (2018a). Efficient GAN-based anomaly detection. *arXiv:1802.06222*.

Zenati, H., Romain, M., Foo, C.-S., Lecouat, B. & Chandrasekhar, V. (2018b). Adversarially Learned Anomaly Detection. *IEEE Int. Conf. on Data Mining (ICDM)*, pp. 727–736.

Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. (2019a). Self-attention generative adversarial networks. *Int. Conf. on Mach. Learn.*, pp. 7354–7363.

Zhang, H., Qin, B., Tu, T., Guo, Z., Gao, F. & Wen, Q. (2019b). An Adaptive Encryption-as-a-Service Architecture Based on Fog Computing for Real-time Substation Communications. *IEEE Trans. on Ind. Inform.*

Zhang, L., Lambotharan, S., Zheng, G., AsSadhan, B. & Roli, F. (2021a). Countermeasures Against Adversarial Examples in Radio Signal Classification. *IEEE Wireless Commun. Lett.*, 10(8), 1830-1834. doi: 10.1109/LWC.2021.3083099.

Zhang, L., Lambotharan, S., Zheng, G. & Roli, F. (2021b). A Neural Rejection System Against Universal Adversarial Perturbations in Radio Signal Classification. *2021 IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-5. doi: 10.1109/GLOBECOM46510.2021.9685697.

Zhang, L., Lambotharan, S., Zheng, G., Liao, G., Demontis, A. & Roli, F. (2022). A Hybrid Training-Time and Run-Time Defense Against Adversarial Attacks in Modulation Classification. *IEEE Wireless Commun. Lett.*, 11(6), 1161-1165. doi: 10.1109/LWC.2022.3159659.

Zhao, H., Lin, Y., Gao, S. & Yu, S. (2020). Evaluating and Improving Adversarial Attacks on DNN-Based Modulation Recognition. *GLOBECOM 2020 - 2020 IEEE Global Commun. Conf.*, pp. 1-5. doi: 10.1109/GLOBECOM42002.2020.9322088.