11-17-2022

Deep Learning -Powered Computational Intelligence for Cyber-Attacks Detection and Mitigation in 5G-Enabled Electric Vehicle Charging Station

Manoj Basnet

Deep Learning -Powered Computational Intelligence for Cyber-Attacks Detection and Mitigation in 5G-Enabled Electric Vehicle Charging Station

by

Manoj Basnet

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Electrical and Computer Engineering

The University of Memphis

December 2022

DEDICATION

I would love to dedicate this Ph.D. dissertation to my late Mother, Saraswoti Basnet, my caring wife, Kamana Dahal, and the rest of my great family, who compassionately supported me from overseas throughout my studies abroad.

ACKNOWLEDGEMENT

I am sincerely grateful to my supervisor Dr. Mohd. Hasan Ali for his invaluable guidance and continuous support during my Ph.D. study at the University of Memphis. His insightful advice and great experience encouraged me in all aspects of my academic life. My gratitude extends to the Department of Electrical & Computer Engineering for their funding support and Carnegie R1 Doctoral Fellowship.

I would also like to warmly appreciate the committee members Dr. Madhusudhanan Balasubramanian, Dr. John Hochstein, and Dr. Myounggyu Won, who accepted to participate in the committee and for their valuable time and advice.

PREFACE

"***Guided by Learning, Optimization, and Control, to explore Computation, Cosmos, and Consciousness.***"

Most of my youth was driven by learning and maximizing my goals and rewards. The immense pleasure of achieving the goal and the miserable pain of losing were worth experiencing: *The hard way of learning*. However, in my mid-20s, when I had to fulfill multiple competing objectives, i.e., profession, family, health, I realized maximization does not work as it works on one dimension, i.e., one objective. Therefore, I realized life could be more beautiful with stability or balance brought by optimization, not maximization. Control/discipline is the only way to bind us in our desired direction. Like my life, I am always passionate about learning (Computational Intelligence), Optimization (heuristics, Meta-heuristics), and Control algorithms. I am pretty sure that one-day machines will achieve these learning, optimization, and control (LOC) skills to a sublime degree. It could eventually help us unwind the mysteries of Computation, the Cosmos, and Consciousness (3Cs).

The outcome of this research has been published in two journal articles (one under review), four conference papers (one to be submitted), and one book chapter (under publication). The contents of this dissertation include all these publications, which are listed below:

Chapter 2: M. Basnet and M. H. Ali, "Deep Learning-based Intrusion Detection System for Electric Vehicle Charging Station," in *2020 2nd International Conference on Smart Power Internet Energy Systems (SPIES)*, Sep. 2020, pp. 408–413. doi: 10.1109/SPIES48661.2020.9243152.

M. Basnet and M.H. Ali, "WCGAN-Based Cyber-Attacks Detection System in the EV Charging Infrastructure," in *2022 4th International Conference on Smart Power Internet Energy Systems (SPIES)*, Oct. 2022 (Accepted for publication)

Chapter 3: M. Basnet and Mohd. H. Ali, "Exploring cybersecurity issues in 5G enabled electric vehicle charging station with deep learning," *IET Gener. Transm. Distrib.*, p. gtd2.12275, Aug. 2021,

doi: 10.1049/gtd2.12275.

Chapter 4: M. Basnet, S. Poudyal, Mohd. H. Ali, and D. Dasgupta, "Ransomware Detection Using Deep Learning in the SCADA System of Electric Vehicle Charging Station," in *2021 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, Sep. 2021, pp. 1–5. doi: 10.1109/ISGTLatinAmerica52371.2021.9543031.

Chapter 5: M. Basnet and M. H. Ali, “A Deep Learning Perspective on Connected Automated Vehicle (CAV) Cybersecurity and Threat Intelligence,” in *Deep Learning and Its Applications for Vehicle Networks, CRC Press,* Taylor & Francis Group, 2022 (under Publication).

Chapter 6: M. Basnet and Mohd. H. Ali, " Deep Reinforcement Learning-Driven Mitigation of Adverse Effects of Cyber-Attacks on Electric Vehicle Charging Station," *IEEE Systems Journal* (under review).

M. Basnet and Mohd. H. Ali, " Cyber-Attacks Mitigation Strategies in the EV Charging Infrastructure," *2023 IEEE PES General Meeting*, 16-20 July, 2023 (To be submitted).

ABSTRACT

An electric vehicle charging station (EVCS) infrastructure is the backbone of transportation electrification. However, the EVCS has various cyber-attack vulnerabilities in software, hardware, supply chain, and incumbent legacy technologies such as network, communication, and control. Therefore, proactively monitoring, detecting, and defending against these attacks is very important. The state-of-the-art approaches are not agile and intelligent enough to detect, mitigate, and defend against various cyber-physical attacks in the EVCS system. To overcome these limitations, this dissertation primarily designs, develops, implements, and tests the data-driven deep learning-powered computational intelligence to detect and mitigate cyber-physical attacks at the network and physical layers of 5G-enabled EVCS infrastructure. Also, the 5G slicing application to ensure the security and service level agreement (SLA) in the EVCS ecosystem has been studied. Various cyber-attacks such as distributed denial of services (DDoS), False data injection (FDI), advanced persistent threats (APT), and ransomware attacks on the network in a standalone 5G-enabled EVCS environment have been considered. Mathematical models for the mentioned cyber-attacks have been developed. The impact of cyber-attacks on the EVCS operation has been analyzed.

Various deep learning-powered intrusion detection systems have been proposed to detect attacks using local electrical and network fingerprints. Furthermore, a novel detection framework has been designed and developed to deal with ransomware threats in high-speed, high-dimensional, multimodal data and assets from eccentric stakeholders of the connected automated vehicle (CAV) ecosystem. To mitigate the adverse effects of cyber-attacks on EVCS controllers, novel data-driven digital clones based on Twin Delayed Deep Deterministic Policy Gradient (TD3)

Deep Reinforcement Learning (DRL) has been developed. Also, various Bruteforce, Controller clones-based methods have been devised and tested to aid the defense and mitigation of the impact of the attacks of the EVCS operation. The performance of the proposed mitigation method has been compared with that of a benchmark Deep Deterministic Policy Gradient (DDPG)-based digital clones approach. Simulation results obtained from the Python, Matlab/Simulink, and NetSim software demonstrate that the cyber-attacks are disruptive and detrimental to the operation of EVCS. The proposed detection and mitigation methods are effective and perform better than the conventional and benchmark techniques for the 5G-enabled EVCS.

TABLE OF CONTENTS

Content **Page**

List of Tables xiv
List of Figures ..... *xvi*
Abbreviations xix
Chapter 1 Introduction 1
1.1 Background 1
1.2 Motivation 6
1.3 Objectives 8
1.4 Novelty of the Proposed Work 9
1.5 Organization of this Dissertation 10
Chapter 2 Literature Review 11
2.1 Introduction 11
2.2 Intrusion detection and prevention based on deployment location 16
2.2.1 Network Intrusion Detection System (NIDS) 16
2.2.2 Host Intrusion Detection System (HIDS) 17
2.3 Intrusion detection techniques 18
2.3.1 Signature (Knowledge) based approach 18
2.3.2 Anomaly (Behavior) based approach 19
2.3.3 Stateful protocol analysis (Specification) based approach 19
2.4 Anomaly-based ML techniques for IDS 20
2.4.1 Multilayered Perceptron (MLP) 21
2.4.2 Support Vector Machine (SVM) 23
2.4.3 Deep Belief Network (DBN) 23
2.4.4 Convolutional Neural Network (CNN) 24
2.4.5 Recurrent Neural Network (RNN) 26
2.4.6 Hierarchical Temporal Memory (HTM) 27
2.5 Cyberattack impacts mitigation in EVCS 30
2.6 Benchmark Dataset 33
2.6.1 KDD CUP 99 34
2.6.2 NSL KDD 34

2.6.3 UNSW-NB15 34
2.6.4 KYOTO 35
2.6.5 WSN-DS 35
2.6.6 CIC IDS 2017 35
2.7 5G Technology 35
2.8 Chapter Conclusion 37
Chapter 3 Impact Analysis of Cyber-Attacks on 5G-Enabled Electric Vehicle Charging Station 39
3.1 Introduction 39
3.2 Proposed EVCS Architecture 41
3.2.1 Control Circuitry 41
3.2.1.1 PV Control 41
3.2.1.2 BES Control 42
3.2.1.3 EV Control 43
3.2.2 System Formulation and Component Modelling 43
3.2.2.1 PV array 43
3.2.2.2 Boost Converter 44
3.2.2.3 Bidirectional DC-DC converter 45
3.2.3 5G Architecture 46
3.2.3.1 EVCS 46
3.2.3.2 gNB 46
3.2.3.3 *EPC* 47
3.2.3.4 SCADA server 47
3.3 Cybersecurity Issues in EVCS 47
3.3.1 Threat Landscapes of 5G enabled EVCS Cyber-Physical System 48
3.3.1.1 Denial of service 48
3.3.1.2 Malicious code 49
3.3.1.3 Exploitation 49
3.3.1.4 Abuse 49
3.3.1.5 Manipulation 49
3.4 Cyberattack Modeling 50
3.4.1 FDI Attack Modeling 50
3.4.2 DDoS Attack Modeling 51
3.4.3 DDoS Launch through 5G 53

3.5 Simulation Setups 54
3.6 Simulation Results and Discussion 55
3.6.1 Impact analysis of FDI attacks 56
3.6.1.1. FDI Attacks launched on different controllers at different times 56
3.6.1.2 FDI Attacks launched on different controllers simultaneously 58
3.6.2. Impact analysis of DDoS attack 60
3.7. Chapter Conclusion 62
Chapter 4 Cyberattack detection methods in the Electric vehicle charging station 63
4.1 Introduction 63
4.2 Performance Metrics 64
4.2.1 Accuracy 64
4.2.2 Precision 64
4.2.3 True positive rate/Recall 65
4.2.4 F1-Score/Measure 65
4.2.5 False Positive (Alarm) rate 65
4.2.6 Receiving Operating Characteristics(ROC) 65
4.3 Deep Learning-based Network Intrusion Detection System for Electric Vehicle Charging Station 66
4.3.1 Proposed IDS Methodology 66
4.3.1.1 Dataset 67
4.3.1.2 Deep Neural Network 68
4.3.1.3 Long Short-term Memory (LSTM) 68
4.3.2 Results and Discussion 69
4.3.2.1 Plot-based Responses 69
4.3.2.2 Performance Evaluation 75
4.4 Deep Learning-based Host Intrusion Detection System for EVCS 77
4.4.1 Proposed IDS Methodology 77
4.4.1.1 Data Set 78
4.4.1.2 Stacked/Deep LSTM 79
4.4.2 Results and Discussion 81
4.5 Ransomware Detection using Deep Learning in the SCADA System of Electric Vehicle Charging Station 83
4.5.1 SCADA-Controlled EVCS 83

4.5.2 Ransomware Attack Modelling on EVCS 84
4.5.3 Proposed Framework for Ransomware Detection 85
4.5.3.1 Datasets 88
4.5.3.2 Deep Learning Architectures and parameters setting for simulations 88
4.5.4 Simulation results and discussion 89
4.5.4.1 Simulation Setups 89
4.5.4.2 Ransomware-driven DDoS attack 90
4.5.4.3 Ransomware-driven FDI attack 91
4.5.4.4 Deep learning-based analysis 92
4.6 GAN-based Network Intrusion Detection System for Electric Vehicle Charging Station 95
4.6.1 Proposed EC-WCGAN Methodology 95
4.6.1.1 Generative Adversarial Network (GAN) 95
4.6.1.2 WCGAN with External Classifier (EC-WCGAN) 97
4.6.1.3 Data set 99
4.6.2 Results and Discussion 100
4.6.2.1 Plot-based Responses 101
4.6.2.2 Performance comparison with DNN and LSTM Algorithm 103
4.7 Chapter Conclusion 104
Chapter 5 Mitigation of Adverse Effects of Cyber-Attacks on Electric Vehicle Charging Station 106
5.1 Introduction 106
5.2 System Model and Mathematical Formulation 107
5.2.1 Reinforcement Learning 108
5.3 Attack Modeling 111
5.4. Proposed Mitigation Techniques 113
5.4.1. Controller clone-based mitigation employing the DRL TD3 algorithm 113
5.4.1.1 Twin Delayed Deep Deterministic Policy Gradient (TD3) 114
5.4.1.2 Graphical representation of TD3 algorithm 115
5.4.1.3 PV agent 116
5.4.1.4 BES agent 117
5.4.1.5 EV agent 118
5.5 Experimental setups for the TD3-Based method 119
5.5.1 Configurations of TD3 Critic networks 119

5.5.2 Configurations of TD3 Actor networks 120
5.5.3 Training an agent 122
5.6 Benchmark Deep Deterministic Policy Gradient (DDPG) 122
5.7 Computational performance Comparison of DDPG and TD3 123
5.8 Bruteforce mitigation 127
5.9 Controller clone-based mitigation 129
5.10 Simulation Results and Discussion 130
5.10.1 Type I and Type II Attacks 130
5.10.2 Mitigation results and analysis of proposed TD3-based clones vs. DDPG clones 131
5.10.2.1 Type I Attack on Different Times and Mitigation Analysis 131
5. 10.2.2 Type I Attack simultaneously on all controllers and mitigation analysis 134
5.10.2.3 Type II Attack on different times and mitigation analysis 135
5.10.2.4 Type II Attack simultaneously on all controllers and mitigation analysis 136
5.10.2.5 Performance Comparison of Various Proposed Methods 137
5.10.3 Mitigation results and analysis of the Bruteforce and controller clone 139
5.11 Chapter Conclusion 142
Chapter 6 A Deep Learning Perspective on Connected Automated Vehicle Cybersecurity and Threat Intelligence 144
6.1 Introduction 144
6.2 CAV technological enablers: Automation and Connectivity 146
6.3 CAV threat landscape and threat intelligence 148
6.3.1 In-vehicle (low-level sensor) cyber vulnerabilities 149
6.3.2. Vehicle control modules 150
6.3.3. Security analysis of CAV threats 151
6.3.4. Attack surfaces 152
6.3.5 Organizational risks to the CAV ecosystem 153
6.4 CAV threat mitigation: anomaly detection and classification with deep learning 153
6.5 Frontiers in deep learning :Advancement and Future) 155
6.5.1. Meta-learning 157
6.5.2. Federated learning 158
6.6 End-to-end deep CNN-LSTM architecture for CAV cyberattack detection 159
6.6.1 performance analysis 161

6.6.1.1 Dataset 161
6.6.1.2 Evaluation metrics 164
6.7 Results and discussions 165
6.8 Chapter Conclusion 168
Chapter 7 Analysis of 5G Slicing Approach to Electric Vehicle Charging Station 170
7.1 Introduction 170
7.2 Benefits 5G Slicing 170
7.3 Key Performance Index (KPI) requirements for EVCS 171
7.4 5G Slicing model for EV infrastructure 172
7.5 5G network Slicing Architecture 174
7.6 Isolation and Security of NSIs 174
7.7 Proposed 5G slicing architecture for EVCS 176
7.8 Discussions 177
7.9 Chapter Conclusion 178
Chapter 8 Conclusion, Contributions, and Future Work 179
8.1 Conclusion and Contributions 179
8.2 Future Work 181
Bibliography 184

List of Tables

Table 1.1 Cyberattacks disrupting/damaging the physical process 4
Table 3.1 CIA triad of 5G assets..... 48
Table 3.2 Network performance with increasing attack penetration 55
Table 4.1 Confusion matrix. 63
Table 4.2 Datasets overview 67
Table 4.3 Classification metrics of DNN and LSTM for Binary Classification. 74
Table 4.4 Classification metrics of DNN for multi-class Classification 75
Table 4.5 Classification metrics of LSTM for multi-class Classification 75
Table 4.6 stacked LSTM architecture .....*Error! Bookmark not defined.*
Table 4.7 Datasets overview for HIDS .....*Error! Bookmark not defined.*8
Table 4.8 Classification Metrics of Stacked LSTM..... 82
Table 4.9 The area under the curve (AUC) and Accuracy (ACC) of 10-fold stratified cross-validation for RDF 92
Table 4.10 Training Time for RDF..... 92
Table 4.11 Performance Metrics After 10 Fold Cross-Validation for RDF 92
Table 4.12 Classification metrics for DDoS Detection using EC-WCGAN 101
Table 4.13 Classification metrics for multiclass classification using EC-WCGAN 102
Table 5.1 Parameters for attack modeling 111
Table 5.2 Summary of multiple independent agents 118
Table 5.3 Actor-Critic Network parameters 120
Table 5.4 Training parameters setting 120
Table 5.5 PV power statistics in Watt during normal, attack, and mitigation 131
Table 5.6 DC bus voltage statistics in Volts during normal, attack, and mitigation 132
Table 5.7 BES current statistics in Ampere during normal, attack, and mitigation 132
Table 5.8 BES voltage statistics in Volts during normal, attack, and mitigation 132
Table 5.9 EV current statistics in Ampere during normal, attack, and mitigation 133
Table 5.10 EV voltage statistics in Volts during normal, attack, and mitigation 133
Table 5.11 Comparison between the proposed and the STATE-OF-THE-ART algorithms..... 137
Table 5.12 Performance comparison of proposed Bruteforce and Clone Mitigations 140

Table 6.1 Data distribution for CAV .....*Error! Bookmark not defined.*
Table 6.2 Performance metrics of DCNN-LSTM..... 166
Table 6.3 Classwise performance evaluation of the proposed DCNN-LSTM model 166

List of Figures

Figure 2.1 Deployment scenarios of 5G 12
Figure 2.2 5G Security architecture 37
Figure 3.1 Proposed 5G-enabled EVCS Architecture 41
Figure 3.2 BES control a) Outer voltage control b) Inner current control. 42
Figure 3.3 EV controller a) Outer voltage control b) Inner current control 43
Figure 3.4 One diode equivalent circuit of PV module. 44
Figure 3.5 a) Boost converter b) Bidirectional DC-DC converter. 45
Figure 3.6 SYN-Flood Attack 54
Figure 3.7 Snapshot of Network Architecture with three attackers in NetSim.. 55
Figure 3.8 Impacts of FDI attacks launched at PV controller from 2-4 seconds, BES controller from 6-8 seconds, and EV controller at 10 -12 seconds.. 57
Figure 3.9 Impacts of FDI attacks launched at all controllers simultaneously from 2-4 seconds. ... 59
Figure 3.10 Impacts of DDoS attacks launched at PV controller from 2-2.5 seconds, BES controller from 6-6.5 seconds, and EV controller at 10 -10.5 seconds.. 61
Figure 4.1 Steps involved in the Deep learning approach.. 67
Figure 4.2 Variance captured by singular values... 70
Figure 4.3 Two-component PCA plot. 71
Figure 4.4 Model accuracy vs. model loss for the binary classification using DNN.. 72
Figure 4.5 Model accuracy vs. model loss for the binary classification using LSTM.. 72
Figure 4.6 Model accuracy vs. model loss for the multiclass classification using DNN.. 73
Figure 4.7 Model accuracy vs. model loss for the multiclass classification using LSTM 73
Figure 4.8 Binary classification accuracy... 74
Figure 4.9 Multi-class classification accuracy. 74
Figure 4.10 Proposed IDS at EVCS 77
Figure 4.11 Accuracy and loss during Training and Validation progression. 82
Figure 4.12 Confusion Matrix for assessing model performance 82
Figure 4.13 SCADA system connected to EVCS through 5G 84
Figure 4.14 Proposed ransomware detection system for the smart grid architecture.. 87
Figure 4.15 The internal architecture of the proposed RDF monitoring and detection system 87

Figure 4.16 Ransomware-driven DDoS attack impact on charging behavior 91
Figure 4.17 Ransomware-driven FDI attack impact on charging behavior..... 92
Figure 4.18 Model accuracy vs. loss for a single experiment for DNN, 1D CNN, and LSTM top to bottom.. 94
Figure 4.19 EC-WCGAN method for DDoS attack detection..... 97
Figure 4.20 Impacts of DDoS attacks launched at PV controller from 2-2.5 seconds, BES controller from 6-6.5 seconds, and EV controller at 10 -10.5 seconds.. 100
Figure 4.21Generator, Critic, and Classifier loss during the training of binary classification (DDoS attack or Normal operation)..... 102
Figure 4.22 Generator, Critic, and Classifier loss during the training of multiclass classification.. 102
Figure 4.23 Comparative performance of EC-WCGAN with existing DL models..... 104
Figure 5.1 Proposed Detection and Defense based on DRL..... 108
Figure 5.2 The agent environment interaction of RL in a Markov Decision Process... 109
Figure 5.3 Graphical representation of a TD3 agent..... 116
Figure 5.4 Structure of proposed a) Critic-Network b) Actor-Network. 121
Figure 5.5 Training performance of the DDPG and TD3 Agents in terms of average rewards.. 125
Figure 5.6 Training performance of the DDPG and TD3 Agents in terms of episode rewards.. 126
Figure 5.7 Training performance of the DDPG and TD3 Agents in terms of episode Q-values.127
Figure 5.8 Impacts of Type I and Type II attacks on duty cycles of different controllers... 131
Figure 5.9 Impacts of Type I attack launched at PV controller from 5-7 seconds, BES controller from 9-11 seconds, and EV controller from 13-15 seconds and the mitigation performance during the attack. 132
Figure 5.10 Impacts of Type I attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds and the mitigation performance during the attack..... 135
Figure 5.11 Impacts of Type II attack launched at PV controller from 5-7 seconds, BES controller from 9-11 seconds, and EV controller from 13-15 seconds and the mitigation performance during the attack.. 136

Figure 5.12 Impacts of Type II attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds and the mitigation performance during the attack. 137
Figure 5.13 Control actions of Legacy controllers, DDPG clone, and TD3 clone mitigations. 138
Figure 5.14 Corrective duty cycles of Bruteforce and Controller clone mitigations. 140
Figure 5.15 Mitigation performance during the Type I attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds. 141
Figure 5.16 Mitigation performance during the Type II attack launched at the PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds. 142
Figure 6.1 The number of publications on connected automated vehicles over the last five years. 146
Figure 6.2 Connectivity of the CAV. 148
Figure 6.3 Key attack surfaces of CAV. 152
Figure 6.4 Proposed end-to-end deep CNN-LSTM architecture. 160
Figure 6.5 Variance captured by the Singular values. 163
Figure 6.6 Principal Component Analysis. 164
Figure 6.7 Training and Validation progression of deep CNN-LSTM. 166
Figure 6.8 Confusion Matrix of deep CNN-LSTM 168
Figure 7.1 key performance indicator (KPI)requirement for low latency EVCS 172
Figure 7.2 5G-enabled EVCS architecture 173
Figure 7.3 Fig. 5G reference Architecture [161] 174
Figure 7.4 Isolation dimension in Network Slicing [161] 175
Figure 7.5 5G slicing implementation for secure EVCS 176

Abbreviations

AD	Anomaly-based Detection
AI	Artificial Intelligence
AIDS	Anomaly-based IDS
ANN	Artificial Neural Network
APT	Advanced Persistent Threats
AUC	Area Under the Curve
BDC	Bidirectional DC-DC Converter
BES	Battery Energy Storage
CAE	Convolutional Auto Encoder
CAN	Control Area Network
CAV	Connected Automated Vehicles
CIA	Confidentiality Integrity Availability
CNN	Convolutional Neural Networks
CPS	Cyber-Physical System
DDoS	Distributed Denial of Service
DDPG	Deep Deterministic Policy Gradient
DER	Distributed Energy Resource
DL	Deep Learning
DNN	Deep Neural Networks
DoS	Denial of Sevice
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
ECU	Electronic Control Unit
eMBB	extended Mobile Broadband
EPC	Evolved Packet Core
ESS	Energy Storage System

EV	Electric Vehicle
EVCS	Electric Vehicle Charging Station
EVSE	Electric Vehicle Supply Equipment
FAR	False Alarm Rate
FDI	False Data Injection
GAN	Generative Adversarial Network
gNB	next Generation Node B
HIDS	Host Intrusion Detection System
HTM	Hierarchical Temporal Memory
IDS	Intrusion Detection System
IDPS	Intrusion Detection and Prevention System
IIoT	Industrial Internet of Things
IT	Internet Technology
ITS	Intelligent Transportation System
KPI	Key Performance Index
MEC	Multi-Access Edge Computation
MitM	Man-in-the-Middle
ML	Machine Learning
mMTC	massive Machine Type Communication
MPPT	Maximum Power Point Tracking
NFV	Network Function Virtualization
NIDS	Network Intrusion Detection System
NSI	Network Slice Instantiation
OT	Operational Technology
PCA	Principal Component Analysis
PCC	Point of Common Coupling
PV	Photovoltaic

PCC	Point of Common Coupling
SoC	State of Charge
PI	Proportional-Integral
PLC	Programmable Logic Controller
PRN	Pseudo random Number
PWM	Pulse Width Modulation
LSTM	Long-Short Term Memory
RDF	Ransomware Detection Framework
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SD	Signature-based Detection
SDN	Software Defined Network
SLA	Service Level Agreement
SoC	State of Charge
SCADA	Supervisory Control and Data Acquisition
SPA	Stateful Protocol Analysis
SVM	Support Vector Machine
SVD	Singular Value Decomposition
TCP	Transport Control Protocol
TD3	Twin Delayed Deep Deterministic Policy Gradient
TLS	Transport Layer Security
uRLLC	Ultra Reliable Low Latency Communication
V2X	Vehicle to Everything
WCGAN	Wasserstein Conditional GAN
5G	Fifth Generation

Chapter 1 Introduction

1.1 Background

Electric Vehicles (EVs) are anticipated to bring the next generation of electrified mobility into the transportation industry to cut down the global carbon emission. However, the mass adoption of EVs depends on the efficient deployment of EV charging stations (EVCSs), their security, and reliability. According to the second-quarterly (Q2) data of 2021 from the Alternative Fuels Data Center, the United States hosts 128,474 public and private EVCS ports in 50,054 station locations [1]. In 2021 alone, charging stations increased by more than 55% in the United States. This upsurge is anticipated to grow further along with the announcement of the Bipartisan Infrastructure Law (BIL) to build out the nationwide electric vehicle network in April 2021 [2]. In February of 2022, the Whitehouse with the United States Department of Transportation (USDOT) and the US Department of Energy (USDOE), announced $5 billion over five years for the new National Electric Vehicle Infrastructure (NEVI) program under the BIL to create a network of EV charging stations along with designated alternative fuel corridors in the interstate highway [3].

In contrast with the broad interest and investment in transportation electrification and EVCS deployment, the cyber-physical security hygiene of EVCS standalone/network is often slow-paced, poorly defined, and understudied [4]–[7]. The internet-facing elements of EVCS are primarily designed for communications and controls with other Internet of Things (IoTs) and stakeholders such as EV, EV operators, grid, Supervisory Control and Data Acquisition (SCADA), and EVCS owners and push the air-gapped critical physical infrastructures to the

internet [8]. It could potentially open up large attack vectors for the interconnected systems of the EVCS.

The operational failure of critical infrastructures has widespread and devastating impacts. For instance, Texas's February 2021 winter storm left 4.5 million homes and businesses without power, resulting in 57 deaths and $195 billion in property damage [9]. Similarly, cybercrime-induced coordinated attacks can potentially damage and disrupt critical cyber-physical infrastructures. On May 7, 2021, the largest pipeline in the United States, the Colonial Pipeline, was forced to shut down entirely by the ransomware attack that led to gas shortages on the east coast [10]. The attack vector was a Virtual Private Network (VPN) account with compromised passwords that allowed remote access to the company's computer network. The hackers threatened to release 100 GB of customers' data should the ransom was not paid [11]. The colonial pipeline prevented further attacks and damages to its critical physical process by paying cryptocurrency $4.4 million worth of ransom. The cybercriminal group, DarkSide, is believed to be behind the attack. Similarly, the meat Giant JBS paid an $11 million ransom in the late May of the same year that partially halted its operations due to ransomware attacks on its branches in Australia, the US, and Canada [12].

The most lethal attacks are the ones that exploit the vulnerabilities of physical controllers and engineer the attacks with domain expertise. In 2017, one of the deadliest and most sophisticated malware attacks, Triton, targeted the safety instrument system (SIS) designed to save lives (the last line of defense) in the Saudi Petrochemical plant [13]. The SIS used Triconex safety controllers distributed by Schneider Electric and widely used in nuclear, oil, and gas refineries and chemical plants across the globe [14]. While the initial attack vector was unknown, enough traces of Triton attacker mobility across the network were found to disrupt or damage the industrial process. The Triton malware attacker exploited and engineered the legacy Triconex architecture and proprietary

communication protocol TriStation to communicate with safety controllers and remotely manipulate the system memory. However, the attack failed and was discovered after an accidental shutdown. It was concluded that the consequence could have been devastating should the payload have been successful [15]. On December 23, 2015, the hacker group Sandworm compromised the distribution centers in the Ukrainian power grid, leaving 230,000 people in darkness for up to six hours. The adversary initially intruded on the SCADA systems, blinded the dispatchers, wiped the SCADA system servers and workstations, and flooded the call centers with Distributed Denial of Services (DDoS) requests [16]. Even two months after the attack, the control centers were unresponsive to any remote commands from the operator. Cybercriminals left the backdoor open by overwriting the firmware on critical devices at 16 substations [17].

Similarly, in June of 2010, a 500 kB self-replicating worm and spyware, Stuxnet, targeted the Iranian Uranium Enrichment facility to tear down the centrifuges [18]. The attack vector was the infected USB drive. First, it targeted the Microsoft Windows machines and networks, then propagated to Windows-based Siemens step 7 software (to program the industrial control system), and finally compromised the Programmable Logic Controllers (PLCs). These are examples of how cyberattacks can disrupt and damage critical infrastructures. Critical infrastructures are the holy grail of cyber war, and we must prepare to defend them. Due to the interoperability issues, poor security hygiene migrated cyber risks from legacy components and not well-defined protocols and standards. The EVCS can be the next target of cybercriminals. Table 1.1 summarizes some infamous cyber-borne attacks that could disrupt and damage physical systems.

Table 1.1 Cyberattacks disrupting/damaging the physical process

Date	Target	Impact	Attack Class
05/2021	Colonial Oil Pipeline	Entire System shutdown	Ransomware
05//2021	JBS meat	Partial system shutdown	Ransomware
12/2017	Saudi Petrochemical Plant	Took over the plant's safety Instrument systems	Malware/Triton
12/2015	Ukrainian Power Grid	Blackout for 1-6 hours	DDoS/BlackEnergy 3
06/2010	Uranium Enrichment Plant	Tore down the centrifuges	Worm/Stuxnet

The seamless adoption of transportation electrification is bringing endless opportunities for technological disruption, such as next-generation networks (5G/6G), the internet of things (IoT), AI-based prediction, reinforcement learning-based optimal control, and bidirectional grid integration. The EVCS interfaced with such disruptive technologies brings the odd stakeholders to the common ground and acts as a plausible threat envelope for impact propagation. Nonetheless, the incumbent protocols and standards used in the EVCS have myriads of exploitable vulnerabilities. Unlike traditional cyber-physical systems (CPS), EVCS stands out as a networked CPS exposing it to the network.

The EVCS suffers from two blended cyber-physical attacks based on attack origin: cyber-enabled physical attacks and physically enabled cyberattacks [19]. Cyber-enabled physical attacks exploit the vulnerabilities in the cyber layer, mainly communication and network, to disrupt, damage, hijack, and freeze the physical processes. Examples of such attacks are Denial of Service (DoS), Man in the middle (MitM), False data injection (FDI), and Side channel attacks. The physically enabled cyberattacks are the categories of blended attacks that exploit vulnerabilities of

humans, supply chains, and software to launch the cyberattacks. Examples are Trojans, advanced persistent threats (APT), physical damage/disruption of critical controllers, and so on.

It motivates the author to design, implement and test some deep learning-based intelligent systems that could detect cyberattacks at the network level of EVCS by analyzing the network packets. Nonetheless, like other ICS adopting cutting-edge computation, communication, and control, EVCS opens large attack surfaces as it interfaces between the fast pace electric mobility and critical power grids. Also, EVCS hardware, firmware, software, and controllers have vulnerabilities, and humans are the weakest link. An EVCS consists of three essential components: sensing, communication and networking, and computational components [20]. The sensing components deal with the array of wired/wireless sensors to assess the health and safety checks of the various electrical components in the EVCS. Communication and networking components interact with the local grid, SCADA system, internal sensors, and EVs through the internet to ensure energy efficiency and availability. Enabling wireless technology might be Wi-Fi, cellular, Bluetooth, etc. The computational components help perform logical, arithmetic, and control functions. An EV owner has to schedule the charging through an app or the internet so that the maximum number of EVs can be integrated into the grids [21]. An EVCS might ask for authentication before charging so that personal and financial information must be shared through some media such as radio frequency identification (RFID), Bluetooth, and near-field communication (NFC). These wireless communications pose extreme vulnerabilities in EVCS.

The motivations behind the cyberattack on EVCS range from pranks, electricity theft, and identity theft to severe APT such as ransomware and malware, where EVCS might work as an entry point [22]. The denial of service (DoS) attack has been among the most widely seen attacks. The DoS attack causes congestion in the network with fake requests so that all the network

components are busy processing the fake requests and unable to respond to genuine requests [23]. Therefore, DoS attacks must be taken carefully; otherwise, they are costly in terms of the availability of resources.

Based on the above background, as the first layer of defense, this dissertation proposes the Deep Learning powered Network Intrusion Detection System (NIDS) that oversees and monitors the entire EVCS network by using the network packets. The algorithm performance is further improved using the Wasserstein Conditional Generative Adversarial Network (WCGAN) with an external classifier. As a last layer of defense in the physical layer of EVCS infrastructure, this dissertation proposes the air-gapped Host Intrusion Detection System (HIDS) that oversees the electrical and control signals in the operational EVCS. Some Advanced Persistent Threats (APTs) may bypass the network intrusion detection and prevention tools that motivate the author to develop an intelligent system that could detect attacks by analyzing the local electrical fingerprints of the process. The dissertation studies a Bruteforce, controller clones, and data-driven RL-based cyber defense that could detect and mitigate controller targeted APT in the EVCS charging process. The internet of EVs and EVCS may need stringent requirements in terms of latency, bandwidth, and the number of connections. The 5G is the candidate technology capable of providing ultra-reliable low-latency communication (uRLLC), extended mobile broadband (eMBB), and massive machine-type communication (mMTC) that can guarantee the SLAs with added security and isolation. Finally, some of the applications of 5G slicing to the EV infrastructure are surveyed in this work. The following subsections briefly outline the motivation for this research and discuss the objectives pursued by the researcher.

1.2 Motivation

Cyberattacks on critical infrastructures are lethal as it impacts a large population and

compromises the safety of physical and non-physical assets. Therefore, it is imperative to conduct periodic threat assessments and vulnerability analyses and accordingly design and develop agile and stealthy cyberattack detection, mitigation, and defense methodologies. In summary, the history of past attacks taught us vital lessons. Firstly, the attack can be a malware attack on an IT system targeted at the physical system or advanced persistent threats directly targeted at the legacy controllers of the physical system. Secondly, all legacy controllers (software, hardware) lack security by design and are vulnerable to attacks. Thirdly, there is no consensus on designing attack-resilient, agile, intelligent controllers and hardware. Finally, advanced persistent cyber threats engineered with the domain expertise of legacy controllers and safety systems could cause irreversible damage and havoc in the physical system. Above all, there is an imminent need to develop distributed intelligence to defend the critical process controllers proactively and independently under threat incidence. Ideally, one can propose the solution zero to make EVCS off the internet or completely air-gapped. However, it may strip off many functionalities such as EV scheduling, over-the-air updates of EVCS software and firmware, online billing, remote connections to power grids and other EVCS for power management, etc. Also, the air-gapped system still has insider threats that can result in physically enabled cyberattacks. The general points that invoke motivations in this field are:

- EVCS cyber-physical risks, vulnerabilities, and evolving threat landscapes have not been appropriately explored.
- The impact of cyber-borne attacks on EVCS charging at the physical infrastructure layer is not studied in depth.
- The integration of next-generation wireless networks, such as 5G, could revolutionize the experience of EVCS. However, the incumbent technologies' inherent vulnerabilities and

impacts are not fully exploited.

- State-of-the-art algorithms have not been progressing well for attack detection and prediction in EVCS; They can be aided by cutting-edge computational intelligence at both in-network and standalone levels.

- CPS defense (capability of resisting attack) has been ill-defined and often confused with mitigation (reducing the severity). EVCS attack detection, defense, and mitigation are not fully explored.

- The current research lacks the convergence of IT security, OT security, and physical infrastructure security in EVCS, with the slightest attention to OT and physical infrastructure security.

- The current state of the art lacks the proactive vision for developing embedded intelligence that could defend/correct the attacks on the physical assets, mainly controllers of EVCS or any CPS.

1.3 Objectives

This research aims to address the issues and findings outlined in the motivation section as far as possible. The following objectives are sought in this research:

- This research aims to develop the PV-powered standalone EVCS prototype with appropriate power and control circuitry with power generation, energy storage unit, and power delivery unit able to charge an EV.

- The second research goal is to integrate 5G for its speed, connectivity, and bandwidth and assess/exploit its vulnerabilities to disrupt the charging process in EVCS.

- The third research goal is to design, develop, validate, and assess the performance of the

computation intelligence algorithm capable of overseeing, monitoring, and detecting cyberattacks by analyzing the network packets at the network layer of EVCS.

- The fourth goal is to design, develop and validate the computational intelligence that can detect the stealthy cyberattacks targeted at the physical layers of EVCS by using the local electrical fingerprints.

- The fifth goal is to devise and develop defense and mitigation strategies to mitigate and isolate the impacts of cyberattacks on the EVCS.

- The last goal is to explore the potential security and isolation aided by the 5G slicing in the smart grid with EVs and EVCS.

1.4 Novelty of the Proposed Work

The novelties of this work are aligned with the solutions for issues outlined in the research motivation and objective sections. In summary:

- A 5G integrated PV-powered standalone cyber-secured EVCS prototype has been designed and simulated to deliver power to EVs with appropriate control and power circuitries.

- State-of-the-art Deep learning-based models are implemented and validated to detect the attacks at the network level of EVCS, and the performance appraisal is performed.

- Deep learning-based computational intelligence is adopted to detect the bypassed cyberattacks at the physical controllers of EVCS just with the electrical fingerprint.

- As defense and mitigation strategies, Deep reinforcement learning-based digital control clone strategies, the brute force model, and the controller-clone-based model are devised, implemented, and tested for the mitigation of impacts of cyberattacks on EVCS. Also, the performance appraisal of various proposed methods, including TD3 clones and benchmark

algorithms, has been done.

- Cyberattack detection frameworks employing deep learning are proposed to secure EVCS from ransomware. Controller Area Network (CAN) IDS using staked CNN-LSTM algorithm is proposed and tested in CAV environments.

- A novel secure 5G slicing framework has been proposed to aid isolation and security for EVs and EVCS in the smart grid environment.

1.5 Organization of this Dissertation

This dissertation is composed of 8 chapters. Chapter 2 conducts a literature review and provides a detailed background of cyberattacks at EVCS and cyberattack detection techniques. Chapter 3 explores the impact assessment of the cyberattacks on 5G-enabled EVCS. Chapter 4 presents a novel application of the Deep learning model for cyberattack detection in EVCS. Chapter 5 explores the mitigation of adverse impacts of cyberattacks on EVCS. Chapter 6 discusses the deep learning perspective on connected automated vehicle cybersecurity and threat intelligence. Chapter 7 presents the analysis of the 5G slicing approach to EVCS. Finally, chapter 8 concludes the results of this research, presents key findings, and wraps up the dissertation with some recommendations for future research.

Chapter 2 Literature Review

2.1 Introduction

The intelligent and extensive deployment of EVCS coerces heterogeneous stakeholders and customers to coordinate and communicate. The heterogeneous stakeholders mainly refer to i) the intelligent transportation system (ITS)/vehicle to everything (V2X) infrastructures such as roadside sensors, connected automated vehicles (CAV), EVs, ii) Electric grid infrastructures such as utility, generation, transmission, distribution, sensors, protection and relays so on, and iii) financial institution such as credit card companies for the management of transactions [22]. The extent of administrative privilege for coordinating these eccentric stakeholders to/from the EVCS is still a conflict of interest due to the lack of clearly developed standards for proper interoperability and a fully matured, trustworthy environment [24]. Therefore, the EVCS needs robust, secure, and reliable communication with its stakeholders and customers. In such a scenario, the communication between these multiple nodes may need stringent requirements in terms of latency, bandwidth, and the number of connections. 5G must be the ideal communication tech for fulfilling those requirements. As shown in Fig. 2.1, The most updated deployment scenarios of 5G till now are Industrial internet of things (IIoT) and ultra-reliable low-latency communication (uRLLC), extended mobile broadband (eMBB), massive machine-type communication (mMTC), with additions of ITS/V2X, Integrated access and backhaul (IAB) and New Radio based access to unlicensed spectrum (NR-U) [25]–[29].

The general system architecture of EVCS includes the power delivery modules, communication and control modules, sensing and protection modules, and user interfaces [30].

The power delivery module deals with the unidirectional/bidirectional flow of electric power to/from EV/grid, such as a battery, power supply, and power regulator. The communication and control modules extend the capability to communicate with diverse stakeholders such as EV users, operators, utilities, credit card companies, and transportation. The key enablers for wireless technology may be 5G/6G, Wi-fi 6, Bluetooth, and ZigBee [31]. The sensing and protection modules ensure the electrical components' good health and safety in the EVCS. These open communications provide robust, smart, and accurate control and push air-gapped EVCS physical systems to the edge of cyber-physical vulnerabilities. Some vulnerabilities come with communication protocols, such as authentication, authorization, access control, and some inherent component vulnerabilities. Furthermore, there is always a risk of insider threats and socially engineered advanced persistent threats from notorious hackers [32].

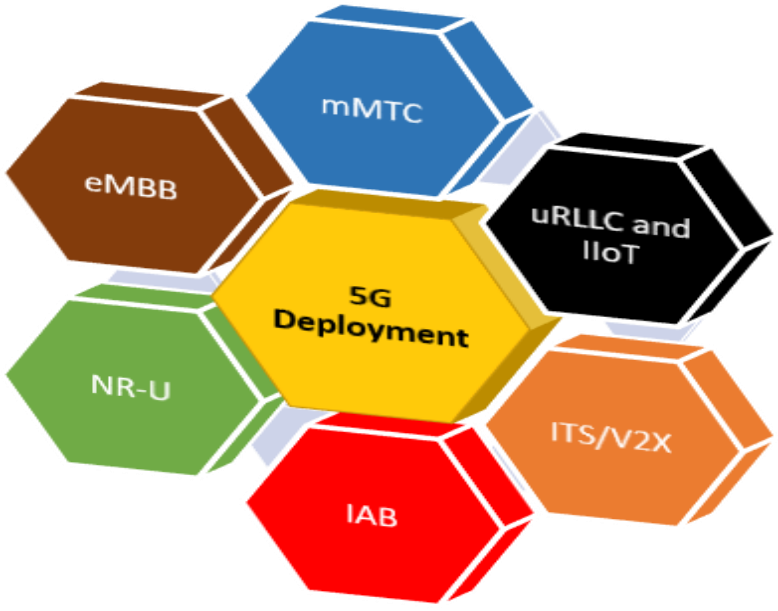


Figure 2.1 Deployment scenarios of 5G

Cyberattacks' motivation in EVCS ranges from a prank, electricity theft, and identity theft to vicious ransomware and malware that could infect the entire EVCS network [33]. The infected

EVCS can spread malware to other EVCS via charging EVs that get infected [34]. The transition and transformation of the attack vector from the communication/cyber layer to the physical infrastructure layer are the intricate metrics that should be analyzed in terms of the aftermath in real physical entities, such as power, current, voltage, and state of charge (SoC). The work in [35] enlists the vulnerability analysis and risk assessment of the EVCS with details of the potential attack scenario, such as a DoS, man in the middle (MitM), and FDI. One of the DoS attacks, e.g., SYN-flood, originates from the inability of transport control protocol (TCP) 's three-way handshakes to correctly identify legitimate requests from the client's nodes and responding each of them. Therefore all of the DoS attacks target the network/source availability by processing illegitimate requests assuming it as the legitimate users causing the congestion [23]. Furthermore, developing a system that efficiently identifies DoS attacks is imperative.

The proactive strategy for accurately detecting and classifying EVCS network attacks on time is known as an intrusion detection system [36]. It helps the operator take protective and preventive measures once the attack is identified. There are three types of IDS derived from their implementation locations, namely: Host-based (HIDS), network-based (NIDS), and hybrid IDS [33]. Three basic intrusion detection techniques have been widely deployed in state-of-the-art applications, namely: Signature-based detection (SD), Anomaly-based detection (AD), and Stateful protocol analysis (SPA) [33] [37].

Researchers listed and characterized exploitable backdoors of the EV charging infrastructure; however, they lack the impact analysis of attacks and detection and mitigation strategies [4]. Authors [20] presented a system approach to list the interactions between various cyber-physical components inside the smart EVCS and a few approaches to improve its cyber-physical security. This research work also lacks an analysis of the impact in EV charging and any proactive detection

techniques. The research work [38] introduced the concept of cyber insurance that transfers the risk of paying a high price from the users to a third party in offline EV charging. Nevertheless, the cyber insurance model lacks security analysis. Likewise, cyber threats targeting different players in an EVCS are presented, lacking impact analysis and mitigation techniques [39]. Most of the research [17]-[20] in EVCS cybersecurity is limited to vulnerability analysis and risk assessment. It cannot explain the exact quantifiable effects on the physical system from the cyber-initiated attack. The literature also lacks proactive attack detection strategies such as IDS [13] and post-attack specifications to deal with the attack in a standalone EVCS or fleet. The impact analysis of cyberattacks in EVCS is studied in [40]–[43]. The three-way handshakes of the TCP protocol at 5G communications are exploited to delay the delivery of critical control signals in EV charging [40]. The control signals are poisoned by injecting the FDI, and the delay-induced and FDI-induced impacts on the EV charging in controllers are studied at the physical layer. Reference [41] simulated the impacts of ransomware-driven DDoS and FDI attacks on SCADA-controlled EVCS capable of damaging and disrupting the battery management system (BMS) operation. On a similar note, Reference [42] investigated the impacts of FDI attacks on the charging interface and hijacking attacks (User mobile app) on EV charging coordination in case of a single point of failure. Reference [43] leveraged reverse engineering and penetration testing techniques to investigate the comprehensive security and vulnerability analysis of the EVCS management system.

EVCS operation and management automation require remote centralized control like SCADA to communicate with numerous field devices with the least possible delay. Since 5G is a proven cellular technology with less than 1 ms latency, capable of millions of machine-type communication [44], it can also be the candidate technology for EVCS communication.

De et al. designed a control-oriented model-based static detector (deviation in battery cell voltage) and a dynamic detector (using system dynamics) algorithms to detect denial of charging attacks and overcharging attacks on PEV battery packs [45]. The threshold-based static and filter-based dynamic detection techniques have the least flexibility toward APT and evolving zero-day attacks.

Girdhar et al. used the STRIDE method for threat modeling and a weighted attack defense tree for vulnerability assessment in the extreme fast charging (XFC) charging station [46]. They proposed a hidden Markov model (HMM)-based detection and prediction system for a multi-step attack scenario. The proposed defense strategy optimizes the objective function to minimize the defense cost added by the cost of reducing the vulnerability index. As a means of defense/mitigation, the authors recommended isolating and taking the compromised EVCS off the interconnections and intercommunication. The traditional isolation-based protection approach miserably fails in the smart grid due to the availability constraints of electricity and few reserved physical backups. On this note, Mousavian et al. implemented mixed-integer linear programming (MILP) that jointly optimizes security risk and equipment availability in grid-connected EVCS systems [34]. Still, their model aims to isolate a subset of compromised and likely compromised EVCS, ensuring the minimal attack propagation risk with a satisfactory level of equipment available for supply-demand. Acharya et al. derived the optimal cyber insurance premium for public EVCS to deal with the financial loss incurred by cyberattacks [47].

The traditional legacy system or embedded devices may not be able to adopt modern cryptographic functions. Intrusion detection and prevention systems can replace the encryption problem in legacy systems [48]. The team of multiple national lab teams recommended the layered network architecture with network segmentation at the lowest hardware/Field devices level for

better zonal management and for blocking the horizontal movement of attacks. Also, they recommended transport layer security (TLS) on the control layer to encrypt the communication between internal devices. The upper layers host the Firewall to control the acceptable traffic and a certificate authority for authentication and attribution capabilities. Finally, the multilayered architecture hosts the external connection in the demilitarized zone, protected data servers, or is enabled via VPN. The business layer featuring the enterprise's IT layer is segmented into it.

2.2 Intrusion detection and prevention based on deployment location

As the first layer of defense, security personnel deploy the IDS in conjunction with intrusion prevention systems (IPS) and collectively called the intrusion detection and prevention system (IDPS). Whenever IDS detects a potential threat, it alerts the IPS to take the appropriate action. Based on the position of its installation in the network, IDS falls under two categories--If the IDS is deployed to monitor the overall network behavior, it is a network IDS (NIDS); on the other side, If the IDS is deployed to monitor the behavior each node (computer) in the network rather than the whole network, It is a host-based IDS (HIDS). However, in practice, both HIDS and NIDS are implemented together as a hybrid IDS to enhance security performance at the cost of increased computational and economical overhead.

2.2.1 Network Intrusion Detection System (NIDS)

NIDS focuses on analyzing the network packets and detecting the anomalous packets in the entire EVCS network. The pros, cons, and monitored packets are as follows.

Pros

- Detects the intrusion by continuously monitoring the network packets
- Deployed in fewer numbers in the network rather than installing on each host computer

- Can detect the threat that bypasses the HIDS

Cons

- Bottleneck due to the congestion created by the network packets destined for the NIDS for the inspection
- Needs more powerful processing capacity (best with parallel processing) and more storage requirements.
- Ruins the whole network security while NIDS got attacked and bypassed by the hackers

Monitored data/packets

- Simple Network Management Protocol (SNMP)
- Network packets (TCP/UDP/ICMP)
- Management Information Base (MIB)
- Router Netflow records

2.2.2 Host Intrusion Detection System (HIDS)

NIDS becomes ineffective against the stealthy attacks originating from the local host or attacks that fool its capability to detect. HIDS focused on local nodes instead of the entire network. The pros, cons, and monitored packets are as follows.

Pros

- Detects the intrusion by continuously monitoring the host files system and system logs.
- Easy installation and no additional hardware requirements

Cons

- Needs to be installed in each host node in the network
- Consumes host resources
- Can only monitor the attack on the machine where it got installed
- Do not care about network invasion
- Delay in reporting attacks

Monitored data/packets

- Log files
- Audit records
- Application program interface (API)
- Rule patterns and system calls

2.3 Intrusion detection techniques

Three basic intrusion detection techniques have been widely deployed in the state of art applications, namely: Signature-based detection (SD), Anomaly-based detection (AD), and Stateful protocol analysis (SPA).

2.3.1 Signature (Knowledge) based approach

Signature-based detection (SD) is based on the pattern or fingerprint of known attacks; that is why it is called knowledge-based or misuse-based detection. SD captures the signature of each incoming data, compares it against the already stored signature, and flags the event as a potential intrusion if it has a different signature.

Pros

- Simple and effective in terms of design
- The signature collection is tedious

Cons

- Ineffective detecting unknown attacks and variants of known attacks

2.3.2 Anomaly (Behavior) based approach

If the attacker injects a new kind of intrusion whose signature is unknown, SD fails miserably; thus, the known behavior-based approach, AD, comes to the rescue. AD approach flags the event as an intrusion if it deviates from the usual network behavior and profile. Behavior/profile is derived by monitoring the regular activities, network connections, and users over a specified period. The problem with this approach is the dynamic behavior of the network—various attributes of the network change with time, which in turn, changes the network behavior/profile. If the new incoming attributes are compared against the old profile, it flags the normal traffic as an intrusion. Therefore, an increased false alarm rate (FAR) is the problem with this approach.

2.3.3 Stateful protocol analysis (Specification) based approach

Stateful protocol analysis (SPA),a.k.a. specification-based detection, on the other hand, extracts and crafts the correct behaviors of critical objects as security specifications and compares them against the actual behavior of the network [49]. The difference between SPA and AD is that the former compares the specification against standard security protocols, while the latter compares the behavior against the known network behavior. For optimal performance, hybrid, i.e., a combination of any two, has been used in state-of-the-art [50]. The pros and cons are presented below.

Pros

- Know and trace the protocol states
- Track the unexpected sequence of commands

Cons

- Resource consumption to protocol state tracing and examination
- Failed to inspect benign protocol behaviors
- It might be incompatible with a dedicated OS or APs.

2.4 Anomaly-based ML techniques for IDS

Anomaly-based IDS (AIDS) generally detects the anomaly by analyzing the incoming unknown data (test data) behavior against the known network behavior's fingerprint (train data). Liao et al. listed the various approaches to profile and match network behavior, such as statistics-based, pattern-based, rule-based, state-based, and heuristic-based (ML-based) [51]. The vast majority of literature discussed the implementation of various ML algorithms for anomaly detection, such as K-nearest neighbors, hidden Markov model, Decision tree, Random forest, Fuzzy logic, Genetic algorithm, and SVM [50]–[52]. The classical ML techniques proved inefficient in accuracy, performance, and classification efficiency. The soaring popularity of AIDS is due to its ability to identify the zero-day attack and the growing interest in ML and AI techniques. With the dominating regime of ANN, the adoption of AI and ML in the research and industry is meteoric because AI and ML can automatically learn anything -- once the model is developed and trained enough to get some threshold accuracy -- and can make an intelligent decision without human intervention. The AI approaches are tested and trusted in signal processing, image processing, and computer vision. Along with the development of biologically

inspired AI that tries to mimic the functional mechanism of the human brain's neocortex, two potent tools of AI are born: artificial neural network (ANN) and hierarchical temporal memory(HTM).

Grammatikis et al. [52] compiled the IDPS developed for advanced metering infrastructure (AMI), SCADA systems, substations, and synchrophasors, as well as presented the 37 cases of Intrusion Detection and Protection System (IDPS) for different databases. They briefly present the comparative analysis, limitations, shortcomings, and recommendations of the state of art IDPS in a smart grid domain. IDPS cannot discriminate accidental faults from cyberattacks; therefore, one should adopt the software-defined network (SDN) technology for global visibility and virtualization. However, SDN-based IDPS adds strengths, which might not be stand-alone against sophisticated human-driven targeted attacks such as coordinated attacks, APT, DDoS, and botnets. The security information and event management system (SIEM) is a multi-agent hierarchical system that aggregates and normalizes information from various information communication technology (ICT) devices to tackle threats. The deep neural net (DNN) based IDPS integrated with SDN and SIEM is the motivation for future works in the smart grid.

2.4.1 Multilayered Perceptron (MLP)

Kasango et al. in [53] incorporated the feature engineering technique in supervised learning to propose a new model for Intrusion Detection. They implemented the min-max normalization algorithm preceded by log transformation and Information Gain (IG) based filter to extract the high-ranked 21 features from the 41 features NSL-KDD dataset before feeding into the four-layered multilevel perceptions (MLP). They achieved the best accuracy of 99.54% on validation data and 86.19% on test data to classify five different datasets: Normal, R2L, U2R, Probe, and DoS. The architecture was composed of three hidden layers with 60 nodes for each, with SoftMax

function for output and ReLU function for the hidden units. However, 21 ranked features generated by the IG filter reduced the validation accuracy and classification accuracy of Support Vector Machine (SVM), k-Nearest Neighbors (kNN), and Random Forests (RF) compared to the one with all features. Apart from that, it is apparent that the Information Gain (IG) filter improved Naïve Bayes (NB's) validation and classification accuracy. Although they got improved validation accuracy using the IG filter, their classification accuracy was reduced by .44%. Also, the algorithm could not improve R2L and U2R anomaly classification accuracy. Machine learning has been ubiquitously implemented for Intrusion Detection and Classification of publicly available datasets. However, no existing works have shown a detailed algorithm performance analysis on those datasets.

Vinay Kumar et al. tried to scrutinize the technical details of the deep neural network and tried to exploit the effects of change in neural network architecture, hyperparameter tuning, and so on on the algorithm performance of six different public datasets, namely: KDDCup 99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS 2017 [54]. The good thing about this paper is its attempt to combine the NIDS and HIDS to develop a scalable real-time IDS. It is evident from the literature that features engineering is the one that is thriving the performance of the DNN algorithm on the various data sets. However, the paper does not aid in improving the performance; instead, it works on the analysis of the comparative performance of Net on six different data sets, as mentioned earlier.

Yao et al. [55] proposed a novel idea of incorporating pure class extraction, pattern discovery, and fine-grained classification in semi-supervised ML to classify the KDD Cup 99 data set. The work attempted to address the two common problems of ML, namely: 1) traffic imbalance in the train set: Unequal number of samples for different categories of Intrusion leads ML not to learn

the non-linear relationships among the features efficiently, especially while there is few data sample in that category. That is why U2R and R2L attacks are poorly classified in publicly available intrusion datasets. 2) The non-identical distribution between the train and test set in the feature space: In contrast to the one, enough samples sometimes do not guarantee classification accuracy if the statistical distribution of the trained and tested features has a vast discrepancy. The claimed accuracy is 96.6 %, with a good F1 score; however, most of their work focused on data engineering rather than the ML model. The hyper-parameters are not flexible for tuning as the modern DNNs. The paper lacks an analysis of the performance of their algorithm based on training time, testing time, and computational cost.

2.4.2 Support Vector Machine (SVM)

The comparative study of the PCA-based and AE-based SVM is done to classify the attacks in the UNSW-NB15 dataset [56]. This dataset is newly introduced and includes nine kinds of recent attacks with 42 features. Artificial Bee Colony (ABC) is used to optimize the parameters of the SVM. The PCA-SVM-ABC has less than 0.1 % accuracy and slightly more FAR (0.1 %) compared to AE-SVM-ABC, with the latter having an accuracy of around 89% and FAR 7 %, respectively. The algorithm's computational cost and running time are unknown, with no significant difference in PCA and AE-based accuracy and FAR. A room for improvement is there to implement advanced algorithms such as CNN with a combination of feature engineering techniques like IG, PCA, and AE.

2.4.3 Deep Belief Network (DBN)

Yang et al. devised the Deep Belief Network (DBN) with Multi Restricted Boltzmann Machine (RBM) to extract the features; the backpropagation (BP) layer to fine-tune the weights of the RBM; and finally fed the features from the aforementioned unsupervised layers to the

supervised SVM to classify the attacks in NSL-KDD dataset [57]. The accuracy of 97.45 %, recall rate of 97.48 %, and precision rate of 97.78 % are achieved. However, time and spatial complexity increases as the number and dimensionality of data increase for the SVM. The robustness of the algorithm to classify different types of attacks is not discussed in the paper. In the context of saturation of the architectural advancement of the neural network, optimization of the architecture is plummeting to improve the accuracy and detection rate of the classification.

Wei et al. [58] implemented particle swarm optimization (PSO) to devise the optimized parameters for the Deep Belief Network (DBN). They claimed the improvement (with respect to other PSO algorithms to optimize DBN, Naïve Bayes, Random Forest, and SVM) of classification accuracy from at least 1.3 % to 24.69 % at the cost of increased average training time by 6.9 % on the benchmark KDD-NSL dataset. Starting with the Artificial Fish Swarm Algorithm (AFSA) to obtain the initial particles by optimizing the PSO algorithm, they applied the initial optimized particles as the initial particle swarm of the genetic algorithm PSO to get the optimal global parameters for the DBN. The longer training time constraints real-time IDS, and they lack the performance comparison with some advanced ANN, such as MLP, CNN, and LSTM-RNN.

2.4.4 Convolutional Neural Network (CNN)

Xiao et al. in [59] proposed the feature reduction technique employed in convolution neural networks to detect the Intrusion in KDD CUP 99 dataset claiming this method is more time-efficient and suitable for real-time IDS along with its greater accuracy as compared to other machine learning algorithms such as Logistic Regression, Decision tree, Random Forest, SVM, AdaBoost, and Naïve Bayes. The layout of the method goes like this: 1) Data Preprocessing: digitize the non-numeric format into the unifying digital format, followed by min-max normalization to get all the data in the range of [0,1] and one-hot encoding to convert the

categorical data into a digital one. 2) Dimensionality Reduction (Principal Component Analysis (PCA) followed by Auto Encoders (AE)): PCA, the linear dimensionality reduction technique, reduces the highly correlated variables into an uncorrelated or independent variable in the lower dimension, and Auto-Encoder (AE) for non-linear dimensionality reduction. 3) Convert the 1D data into 2D to create the image-like data, and 4) Convolution Neural Network (CNN) along with Batch Normalization (BN) for robust fine-tuning. The author claimed accuracy of 94%, detection rate (DR) of 93 %, False Alarm Rate (FAR) of 0.5 %, and training time and testing time of the 20s per epoch and 1 s, respectively. Although the performance seems satiating in terms of time and accuracy, like [50], it can still not address the low accuracy and low detection rate for U2R and R2L attacks raised from the limited data sets of those categories. The proposed future work to resolve the issue is the generative adversarial network to identify more features of these categories.

Park et al. [60] proposed a CNN-based anomaly detection technique for the HTTP attacks employing a convolutional autoencoder (CAE)-decoder inspired by Inception-ResNet-v2, which achieved the highest accuracy so far in state of the art [22]. The HTTP messages are mapped into the image data using character-level binary image transformation. The Encoder decoder is the fully connected symmetrical structure that reconstructs the normal image with lesser Binary Cross-Entropy (BCE) error and anomaly image with more BCE error making it easier to classify. After training the encoder-decoder module, the reconstruction error for the anomaly image varies. Therefore, the Author is the first to incorporate these variations in the error function as binary cross-variable entropy (BCV). The CAE with the BCV achieved higher accuracy than that of BCE. CNN is famous for using the least computational complexity and producing very high accuracy, thus very popular among image and signal processing researchers.

On top of that, CAE removes the additional burden of feature engineering for data cleansing,

while BCV adds more accuracy to the classification. However, it lacks to address the tradeoff between the speed vs. accuracy of the proposed net. The novelty in the paper [61] is to give more cost function weights to the class, having a smaller number of feature vectors to alleviate the effect of traffic imbalance on the DNN algorithm. CNN can learn better features and classify the data in cost and time-effective ways because it shares the same convolution kernels reducing the number of weights stored at the nodes. The author compared the performance of CNN and RNN, successful algorithms in state of the art. On testing the performance on the NSL-KDD test dataset, RNN achieves 2.4 % higher accuracy, 1.6 % higher detection rate, and 3.75 % lesser false alarm rate than CNN [56], [57]. The paper claimed the number of parameters in RNN is 20 times that of CNN, which needs twice the calculation time of CNN, meaning CNN has twice the operational efficiency of RNN [62]. We can infer that CNN is more suitable for real-time intrusion detection systems. However, the comparison does not seem more relevant because both models are not run in the same environment. Their method of converting 1D data into the 2D image using the square root method tends to lose some features since the number of features in the form of a perfect square of some number is highly unlikely. The paper focused on losing the feature with the least coefficient of variation rather than retaining all the features. The proper method for image mapping from 1D data is still open for exploration.

2.4.5 Recurrent Neural Network (RNN)

Sheikhan et al. proposed the partially connected reduced-size RNN with the cluster of features to classify the KDD data set [63]. The misuse-based approach with only two hidden layers achieved an accuracy of 94.1 %, FAR of 0.38, and cost per example (CPE) of 0.1666, which the author claimed was superior to the other classical ML algorithms. However, they could not solve the intriguing problem of the ML algorithm: FAR, which is the lower FAR achieved from MLP

(0.28). The training time for the proposed model is almost 600 seconds faster than MLP; however, it is twice slower compared to Elman-based Neural Networks.

Yin et al. implemented the RNN for the binary classification and multiclass classification of the KDD data set and achieved a binary classification accuracy of 99.81% and multiclass accuracy of 99.53 % [62]. RNN stands out in terms of accuracy compared to traditional ML algorithms, such as Naïve Bayes, J48, NB Tree, Random Forest, Random Tree, MLPs, and SVM. The fully connected RNN has a stronger modeling ability and a higher detection rate than the partially connected one [64]. However, due to the loaded parameters, it is nearly 400 seconds slower than the one with a reduced size RNN.

Kim et al. [65] proposed the long short-term memory (LSTM) RNN for intrusion detection in the KDD Cup99 dataset. They attempted to optimize the architecture by tuning the learning rate and the number of hidden units. With a time step size of 100, batch size of 50, epoch size of 500, a learning rate of 0.01, and hidden units of 80, the average detection ratio achieved is 98.88 % with 10.04 % of FAR. The best part of the paper is an attempt to formulate the efficiency of IDS in terms of detection rate and FAR ratio. However, computational cost and training time is not a part of the evaluation. The proposed model cannot detect the U2R since there are only 30 U2R instances in training. GAN might be one of the good alternatives for traffic imbalance in the data set. Also, a FAR of 10% might not be suited for a sensitive cyber-security scenario.

2.4.6 Hierarchical Temporal Memory (HTM)

HTM is a biologically inspired machine intelligence that mimics the architectures and processes of the neocortex [66]. Unlike deep learning, HTM is a continuous online unsupervised learning that does not need training data or separate training models because automatic model building and learning remove the need for manual maintenance and updating data and model. The

supervised neural network cannot deal with high-speed dynamic data, which requires real-time intrusion detection. HTM can efficiently solve the problem of a different distribution of data and a sparse sample. HTM automatically builds the model for each metric that is being monitored. HTM assigns the time stamp to each metric value, and the encoder converts the temporal data to the sparse distributed representation (SDR). The SDR enables essential attributes such as generalizability across the data stream and strong resistance to noise in the data points. The sequence of SDR is fed to the HTM algorithm. HTM algorithm learns the temporal pattern of the metric data stream under consideration online, like memorizing the melody pattern with newly learned patterns replacing the older pattern. HTM-based anomaly detection is the best candidate for real-time intrusion detection since it is a memory-based online learning system representing any data stream as an SDR to detect the temporal variance. The successful implementation of HTM for anomaly detection in modification attack and replay attack scenarios in In-Vehicle Networks has demonstrated higher accuracy, precision, and recall than the RNN and Hidden Markov Model(HMM) [67]. Although the performance seems superior, there is still room for improvement in FAR and performance improvement.

The paper [68] starts with the ideal requirements for real-world anomaly detection: a) prediction must be made online that is x_t should be classified as usual or anomaly before receiving x_{t+1}. b) the algorithm must run continuously without the requirement for the stream to be stored. c) the algorithm must be automated and unsupervised that does not need data labels and tweaking parameters. d) the algorithm must be adaptive to the dynamic environment and concept drift since the statistics of the data stream are highly non-stationary. e) algorithm should be robust and minimize the false-positive and false-negative rates. The HTM algorithm can detect spatial and temporal anomalies in the noisy and predicted environment. Errors are not always correlated in

HTM and all other algorithms, so that an ensemble-based approach would enhance the accuracy. Since HTM is relatively new, there is plenty of room to work on it.

The works at [40]–[42], [69] assessed the impacts of cyber-enabled physical attacks on EVCS infrastructures ranging from disruption, damage, hijack, and so on. On that note, researchers have worked on numerous detection methods implementing different computational intelligence algorithms, including machine learning and deep learning [7], [9], [13], [14]. The reference [7] designed and engineered a Deep learning-powered (DNN, LSTM) network intrusion detection system that could detect DDoS attacks for the EVCS network based on network fingerprint with nearly 99% accuracy. Similarly, [40] developed a stacked LSTM-based host intrusion detection system solely based on a local electrical fingerprint that could detect stealthy 5G-borne DDoS and FDI attacks targeting the legacy controllers of EVCS with nearly 100% accuracy. Furthermore, several Deep Learning-based ransomware detection engines have been proposed, tested, and evaluated that can share the information in a cloud-based or distributed ransomware detection framework for EVCS [41].

The sophisticated cyberattack detection techniques for EVCS have rapidly evolved on the network and physical levels [2], [3], [6], [7], [8]. Reference [40] proposed and tested stacked-LSTM-based detection capable of detecting cyber-enabled physical attacks on different electronic controllers with nearly 100% performance metrics solely based on the local electrical fingerprint. Moreover, reference [3] tested the efficacy of different deep learning algorithms (DNN, CNN, LSTM) to detect ransomware attacks on networked EVCS. Also, reference [33] proposed deep learning (DNN and LSTM) powered DDoS detection engine for the networked EVCS solely based on the network fingerprint. Similarly, reference [45] tested the effectiveness of the static and dynamic detection algorithm under a denial of charging and overcharging attacks at PEV. The

support vector machine (SVM) based detection algorithm against the FDI attack under the scenario of a P2P energy transaction based on blockchain has been proposed for connected EVs in the parking lots [70].

2.5 Cyberattack impacts mitigation in EVCS

As far as the author's knowledge, there has been minimal or no work toward mitigating cyberattacks in EVCS. Girdhar et al. proposed a defense strategy that optimizes the objective function to minimize the defense cost added by the cost of reducing the vulnerability index. As a means of defense/mitigation, the authors recommended isolating and taking the compromised EVCS off the interconnections and intercommunication [46]. The traditional isolation-based protection approach miserably fails in the smart grid due to the availability constraints of electricity and few reserved physical backups. On this note, Mousavian et al. implemented mixed-integer linear programming (MILP) that jointly optimizes security risk and equipment availability in grid-connected EVCS systems [34]. Still, their model aims to isolate a subset of compromised and likely compromised EVCS, ensuring the minimal attack propagation risk with a satisfactory level of equipment available for supply-demand. Acharya et al. derived the optimal cyber insurance premium for public EVCS to deal with the financial loss incurred by cyberattacks [47].

Reference [71] proposed the PI controller-based mitigation approach for the FDI attack on microgrids. This method is based on the reference tracking application. A feed-forward neural network produces the reference voltage required for the PI controller, and the PI controller injects the signal to nullify the FDI. The problem with the method is that the neural networks optimized under the microgrid's normal operating conditions may produce unreliable reference signals under adversarial conditions such as manipulated inputs. Recurrent neural networks better deal with reference tracking problems than regular feed-forward networks. The proposed model imposes

additional hardware requirements and is not efficient enough to deal with non-linear and periodic FDI attacks.

Reference [72] implemented the DRL-based approach for mitigating oscillations of unstable smart inverters in DER caused by cyberattacks. The adversary who gained system access can reconfigure the control settings of the smart grid to disrupt the distribution grid operations. To mitigate the impact, the authors trained the actor-critic-based proximal policy optimization (PPO) DRL to develop the optimal control policy to reconfigure the control settings of uncompromised DERs. However, this article has not presented the DRL efficacy of mitigation methods.

The reference [73] proposed the concept of an Autonomous Response Controller that uses the hierarchical risk correlation tree to model the paths of an attacker and measures the financial risk at CPS assets. In addition, the competitive Markov Decision Process was used to model the reciprocal security interaction between the protection system and adversary as a multi-step, sequential, two-player stochastic game. The proposed method is tested against the Aurora attack that can create cascading failure and voltage collapse by opening the generator breaker in the testbed.

Based on the above discussions, there are several significant findings: Firstly, state-of-the-art algorithms have progressed well for attack detection and prediction in EVCS, aided by cutting-edge computational intelligence at in-network and standalone levels. Secondly, CPS defense (capability of resisting attack) has been ill-defined and often confused with mitigation (reducing the severity). As a result, CPS research is jumping towards mitigation that optimizes the cost function to protect remaining assets from further invasions by implementing predefined strategies, such as isolation of compromised assets, optimal insurance premium design, and mobilizing reserve resources. The obvious questions are, are we even trying to defend against any attacks in

our critical CPS? Are we correcting the intruder/intruded signals? Thirdly, the current research lacks the convergence of IT security, OT security, and physical infrastructure security, with the slightest attention to OT and physical infrastructure security. The most devastating attacks in history have exploited the vulnerabilities in legacy controllers in OT or physical environments, as evident in Trident and Stuxnet attacks. The current state-of-the-art lacks a proactive vision for developing embedded intelligence that could defend/correct the attacks on the physical assets, mainly EVCS controllers.

Along with the progress of AI, detection engines have been evolving rapidly for EVCS like other CPS. However, cyberattack mitigation, correction, and defense have a vast void to fill in for EVCS, like other CPS. References [71]–[73] devised the cyberattack mitigation techniques for CPS, ranging from PI-based control, RL-based optimal control, and game theoretic defense. The PI controller-based mitigation strategy for the FDI attack on microgrids was suggested in [71]. The reference tracking application is the foundation for this approach. A feed-forward neural network generates the reference voltage needed by the PI controller, and the PI controller injects the signal to cancel the FDI. The issue with the technology is that under adversarial conditions, such as manipulated inputs, neural networks that have been optimized for the microgrid's typical operating settings may give incorrect reference signals. Compared to traditional feed-forward networks, recurrent neural networks are more adept at handling reference tracking issues. The suggested model imposes more hardware requirements and is ineffective against non-linear and recurrent FDI attacks.

The DRL-based method was utilized in Reference [72] to mitigate oscillations of unstable smart inverters in DER brought on by cyberattacks. The attacker who entered the system might change the smart grid's control settings to interfere with the distribution grid's operations. The

authors trained the actor-critic-based proximal policy optimization (PPO) DRL to create the best control policy to modify the control settings of uncompromised DERs to lessen the impact. The effectiveness of mitigation strategies for DRL hasn't been discussed in this article, though. The reference [73] proposed an autonomous response controller that models an attacker's paths using a hierarchical risk correlation tree and assesses the financial risk to CPS assets. Additionally, the adversary and protection system's reciprocal security interaction was modeled using the competitive Markov Decision Process as a multi-step, sequential, two-player stochastic game. The suggested technique is tested against the Aurora attack, which can open the generator breaker and cause cascade failure and voltage collapse in the testbed.

2.6 Benchmark Dataset

For the best performance of ML algorithms, the training data set must represent the real-world attack scenario, perfectly balanced traffic data (enough and a proportional number of samples for each attack category), and similar distribution between train and test data. However, academia is still struggling to get the best data set due to security and privacy issues. On the other hand, data collection, labeling, and attack categorization to build the train data set are tedious and costly for a researcher. Therefore, the simulated data set might be a good option if it could represent the real attack scenario, but it is hard to achieve. At the dawn of rapid advancement in computer processing, storage, cloud computing, sensor resources, wireless communication, and AI algorithms, it is obvious to expect powerful and unforeseen cyber threats in the coming days. Therefore, the ideal dataset should update all attacks daily for the best intrusion detection. Until now, no publicly available data sets can achieve all the requirements above, enabling the author to foray into the best data set for IDS in the smart grid paradigm. The mostly used datasets available online are as follows.

2.6.1 KDD CUP 99

It has been the most widely used dataset since 1999 after Stolfo et al. [74] prepared it based on data captured(binary TCP dump data) in the DARPA'98 IDS evaluation program [75] with 494,021 connections in train data and 311,029 connections in test data set each sample with 41 features. Samples are labeled as either normal or as an attack with four different types of attack, namely denial of service (DoS), the user-to-root attack (U2R), remote to-local attack (R2L), and probing attack [76]. Some problems with the data set are i) data are synthesized and fabricated to preserve privacy and security, which threatens the validity of fabricated data. ii) TCPdump used to collect the traffic data are highly likely to be overloaded and drop the packet, which loses some essential features. iii) no exact definition of attacks.

2.6.2 NSL KDD

Tavallaee et al. [76] proposed the NSL-KDD data set, the refined version of KDD Cup 99, to remove the data redundancies, duplication, and traffic imbalance after a thorough statistical analysis of KDDCup 99. They applied the filter to remove connection records numbered 136,489 and 136,497 from the test data so that the ML algorithms are not biased. The train and test sizes are reduced to 125,973 and 22,544, with 41 features. Still, this data set cannot address the traffic imbalance, Spatio-temporal variation of the network behavior, and the exact representation of the attack scenario.

2.6.3 UNSW-NB15

The two existing data sets mentioned above cannot represent the contemporary orientation of network traffic and attacks. Moustafa et al. [77] created the dataset using the IXIA tool to generate real modern normal and synthetic abnormal network traffic in a synthetic environment. Unlike the previous two data sets, it represents nine major families of modern cyber-attacks

(Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms) with 49 features.

2.6.4 KYOTO

Song et al. [78] created the honeypot system with only 24 features, among which 14 are the most used features of KDDCup 99, and 10 features were extracted from the traffic log of the university of Kyoto in the year 2015.

2.6.5 WSN-DS

In 2016, Almomani et al. [79] came up with a brand new data set to better detect and classify the DoS attacks(in wireless sensor networks (WSN)) into four subcategories, namely, Blackhole, Grayhole, Flooding, and Scheduling attacks. They considered one of the most popular routing protocols in WSN, LEACH protocol, to extract 23 features from NS2 and applied the ANN to classify them.

2.6.6 CIC IDS 2017

CIC IDS 2017 is the most recent and complete public dataset developed by the Canadian Institute for Cybersecurity in 2017. It has focused on collecting real-time traffic (by using B-profile) while building the dataset [80]. It has 84 features designed explicitly for feature extraction using dimensionality reduction techniques such as PCA and AE. and seven recent attacks, namely SSH-Patator, FTP-Patator, DoS, Web, Bot, DDos, and Port Scan. The CIDS 2017 has been updated to CIDS 2018.

2.7 5G Technology

5G is one of the next-generation communication technologies famous for ultra-reliable low latency, extended mobile broadband, massive machine-type communications, and industrial IoT.

The key enablers for 5G are network function virtualization (NFV), software-defined networks (SDN), multi-access edge computation (MEC), and so on. The virtualization of the core network (CN) in 5G, unlike its predecessor, adds higher flexibility and portability for the network resource management that delineates the control and user plane separation (CUPS) with the help of SDN and NFV [81]. SDN and NFV complement each other for simpler network control and management, better elasticity, and eliminating vendor-specific solutions [82]. Besides CN, SDN, and NFV, management and network orchestrator (MANO) and multi-access edge computing (MEC) are key enablers of 5G. The physical infrastructures layer embodies storage, computing, and networking infrastructures. The virtualized infrastructure has a 5G Radio access network (RAN), IAB, 5G core network functions (5G CN NF), MEC, and data network (DN) with MANO for additional network slicing. The ENISA has added processes to the security architecture on its December release because these are the prime stakeholders towards 5G security. The added processes are mobile network operators (MNO), assurance, and vendors. Fig. 2.2 represents the functional diagram of security architecture, and the in-depth information on individual components can be found here [82].

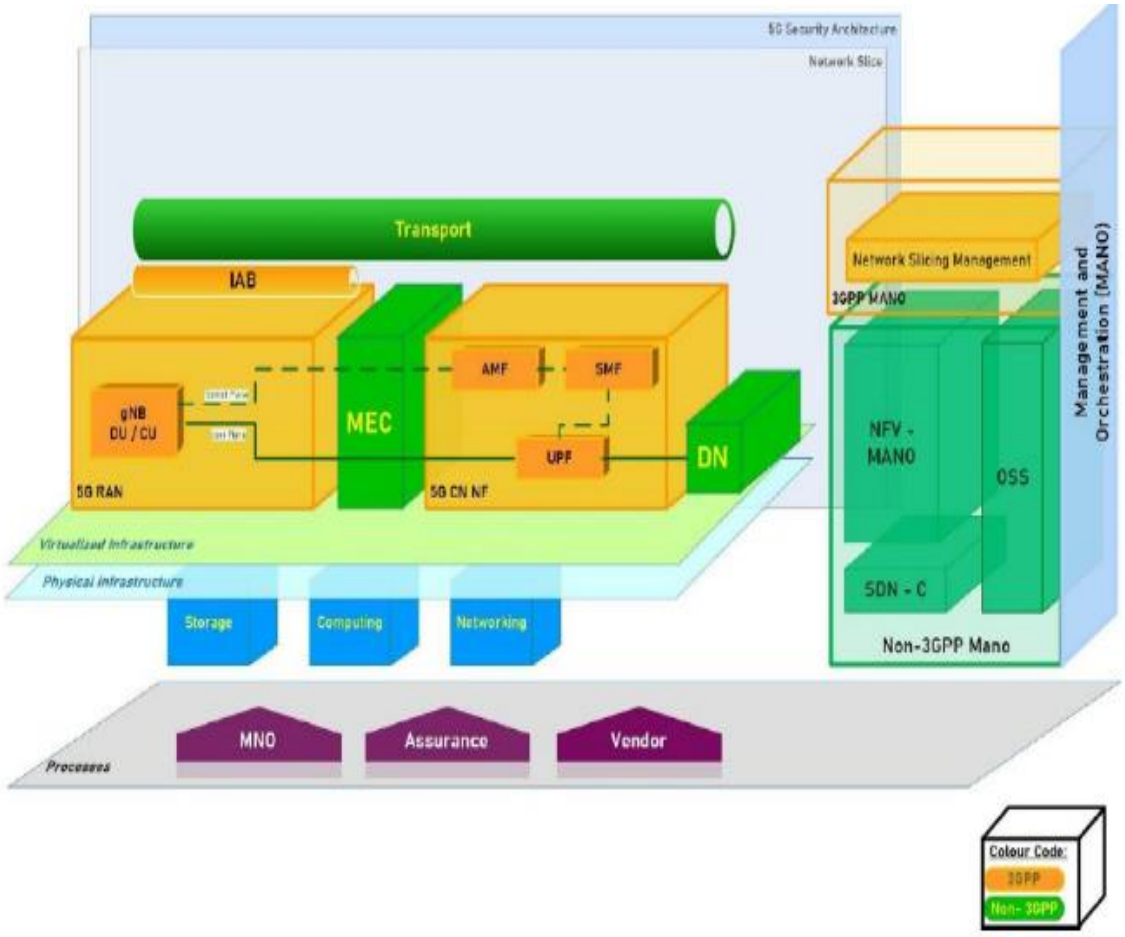


Figure 2.2 5G Security architecture

Threat actors first identify the vulnerabilities of the 5G assets and then exploit them by assessing the attack surfaces.

2.8 Chapter Conclusion

EV charging infrastructures in the 5G paradigm integrate the asynchronous interconnections among heterogeneous ICTs, machine-to-machine, and machine-to-human interconnectedness through IoTs, integrated communications between various wireless sensors and network components to add the robustness, automation, remote control, and self-healing capability in the existing electrical grids. However, cyber-physical threats inherently come up with this boon. IDS acts as the first layer of security by timely detection of the intrusion. Various ML

algorithms discussed above have their unique strengths and weaknesses. DBN, MLP, CNN, RNN, and HTM have an increased order of accuracy from the former to the latter. However, all ML algorithms always have a trade-off between accuracy and convergence speed. Moreover, the author attempted to present the overview of various public datasets and performance metrics that need to be benchmarked to better compare different ML algorithms for intrusion detection. The supervised Deep learning method fails when there is a traffic imbalance between different classes. Also, feature engineering such as PCA, AE, and IG filters was applied to get the best out of the data. High-speed dynamic data classification is still the daydream for supervised deep learning. Therefore, HTM and RL-based approaches are becoming popular day by day. IDS integrated with ML-based SDN offers virtualization and slicing of the network layer, which can flag the anomaly from faults that seemed impossible with all ML algorithms. The research focus should shift towards integrating SDN-based IDS with SIEM, which could tackle the advanced persistent threats by offering real-time visualization. This chapter provided a thorough literature review on cybersecurity issues in EVCS, AI-based detection, and mitigation. In the next chapters, the main parts of the research are presented, where a more specific and detailed explanation of each study will be elaborated. Chapter 3 will study the impact analysis of cyberattacks in EVCS.

Chapter 3 Impact Analysis of Cyber-Attacks on 5G-Enabled Electric Vehicle Charging Station

3.1 Introduction

The surging usage of EVs demands the robust deployment of a trustworthy EVCS with millisecond range latency and massive machine-to-machine communications where 5G could act. However, 5G suffers inherent protocols, hardware, and software vulnerabilities that threaten the communicating entities' cyber-physical security. To overcome these limitations in the EVCS system, this dissertation analyses the impact of FDI and DDoS attacks on the operation of EVCS. This chapter simulates the FDI attack and the SYN flood DDoS attacks on the 5G-enabled remote SCADA system that controls the solar photovoltaics (PV) controller, Battery Energy Storage (BES) controller, and EV controller of the EVCS. The delay has been increased to more than 500 milliseconds with the severe DDoS attack via 5G. The attacks make the EVCS system oscillate or shift the DC operating point. The frequency of oscillation, damping, and the system's resiliency is related to the attacks' intensity and target controller. Finally, we propose the novel stacked LSTM-based IDS solely based on the electrical fingerprint.

3.2 Proposed EVCS Architecture

The proposed EVCS is a standalone, PV-powered, and off-the-grid system. It has three electrical units: PV Generation Unit (PGU), Energy Storage Unit (ESU), and Power Delivery Unit (PDU), as shown in Fig. 3.1. The PGU consists of a PV array, a boost converter, and control circuitry. The PV arrays deliver 1.065 kW at maximum power point (MPP) with the corresponding voltage of 36.75 V and current of 29 A at the constant irradiance of 1000 $\mathrm{W/m^2}$ and constant temperature of 25 ℃. The boost converter boosts the PV voltage (V_{PV}) to the DC link bus bar voltage of V_{ref_bus} . The ESU consists of BES, a DC-DC converter, and control circuitry. The BES

has a nominal voltage of 48 V and a nominal discharge current of 43.47 A with a 100 Ah rated capacity. The DC-DC converter charges the BES in buck mode while there is a surplus generation and discharges to the bus bar in boost mode while the PV can't meet the EV demand. The control circuitry continuously senses the battery current (i_B) and V_{ref_bus}. This generates the pulses to drive the bidirectional DC-DC converter (BDC). PDU could be an offboard in the EVCS or onboard integrated with the EV. Either way, the functionalities remain the same. PDU has a buck converter and an EV controller. The buck converter steps down the voltage from the point of common coupling (PCC) as per the requirement of EV, i.e., V_{ref_ev}. The EV controller continuously monitors the status of the EV battery voltage (V_{ev}) and battery current (I_{ev}) and generates the pulse to adjust the switching of the Buck Converter. Here, all three control units, namely, PV control, BES control, and EV control, can be assessed/overridden by remote operators at SCADA, EVCS, or EV owner through apps or the web via robust 5G communication infrastructures. Though in this dissertation we have considered the EVCS prototype with more focus on cyber-physical security as shown in Fig. 3.1, the EVCS architectures, structures, capacity, charging time, charging type, and interconnections may vary in real practice. But the basic functionalities remain the same with the prototype.

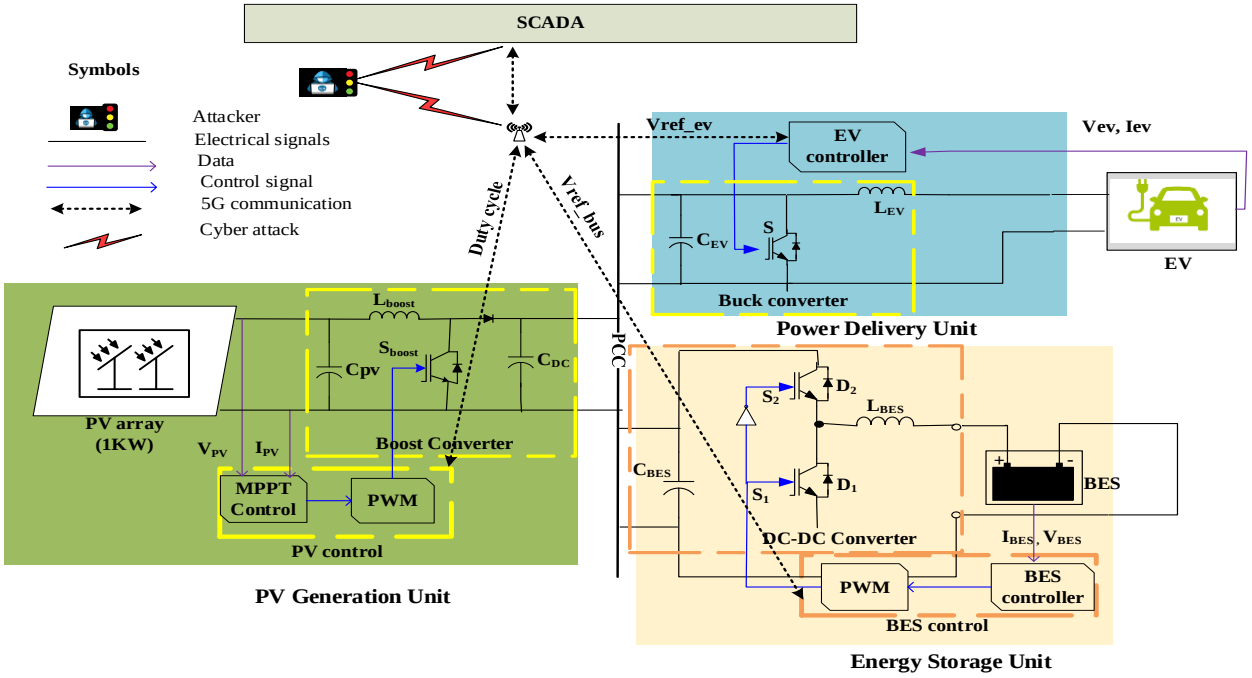


Figure 3.1 Proposed 5G-enabled EVCS Architecture

3.2.1 Control Circuitry

As shown in Fig. 3.1, there are three controllers: the PV control, BES control, and EV control. These are explained in detail below.

3.2.1.1 PV Control

The MPPT control continuously reads the V_{PV} and I_{PV} signals from the PV output. The Perturb and observe Maximum Power Point Tracking (P&O MPPT) algorithm tracks the maximum power points and corresponding V_{PV} and I_{PV} and adjust the duty cycle accordingly. The pulse width modulation (PWM) circuitry block will generate the S_{boost} signal from the duty cycle. The details of the P&O MPPT algorithm can be found in [83]. The duty cycle of MPPT can be adjusted and reinitialized manually by a human operator in the case of malfunction, disaster, and emergency. This operator at SCADA can remotely monitor and control the PV controller via 5G communication.

3.2.1.2 BES Control

The BES controller continuously monitors the I_{BES} ,V_{bus}, and V_{ref_bus} from the system. The slower outer loop in Fig. 3.2a. controls the bus voltage with the help of a Proportional-Integral (PI) controller driven by the error $V_{ref\_bus} - V_{bus}$. It generates the reference signal I_{bes_ref} for the inner current control loop, which is ten times faster than the outer loop, as in Fig. 3.2b.

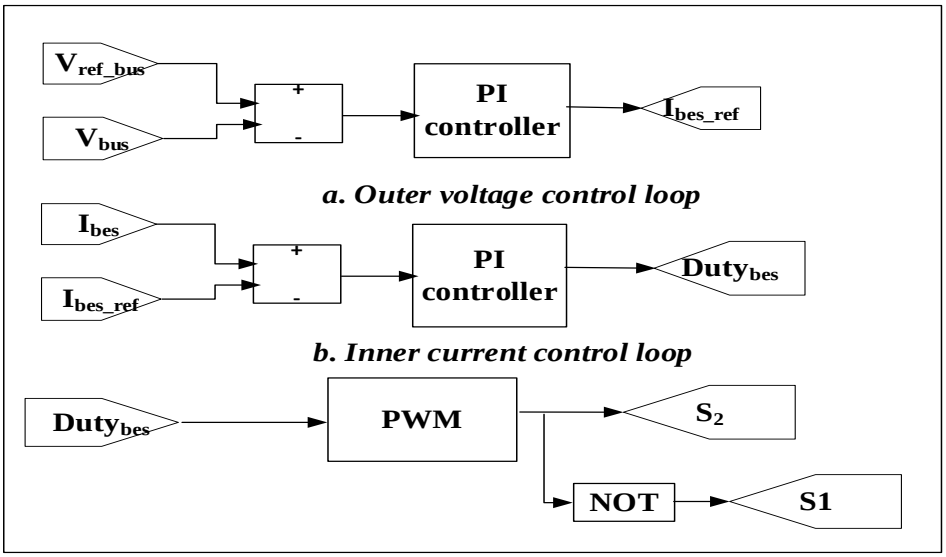


Figure 3.2 BES control a) Outer voltage control b) Inner current control.

The PI controller tries to track the I_{bes_ref} by minimizing the error between the reference current and measured current and generating the duty signal $Duty_{bes}$. This $Duty_{bes}$ drives the PWM to create complementary pulses S_1 and S_2 that trigger the switching of a boost and a buck converter, respectively, in the bidirectional DC-DC converter. The details of this cascaded PI control strategy can be found in [84].

Through the 5G, the SCADA operator at the remote station could wirelessly monitor and control the BES controller at EVCS and can set the V_{ref_bus} as well as other PI controller settings.

3.2.1.3 EV Control

The EV controller continuously monitors the battery's voltage (V_{Bev}) and current (I_{Bev}) of the PEV. This control block might be off-board or onboard the EV. The reference battery voltage (V_{ref_ev}) can be set by the EVCS owner or SCADA operator if it is off-board or can be set by the EV owner for dynamic charging or hardcoded by the original equipment manufacturer (OEM) in the CAN bus if it is onboard. These communications may take place through 5G. As in BES control, the same cascaded outer voltage and inner current control strategy are implemented to generate the Dutybev, which controls the buck converter to regulate the EV charging, as shown in Fig. 3.3 below.

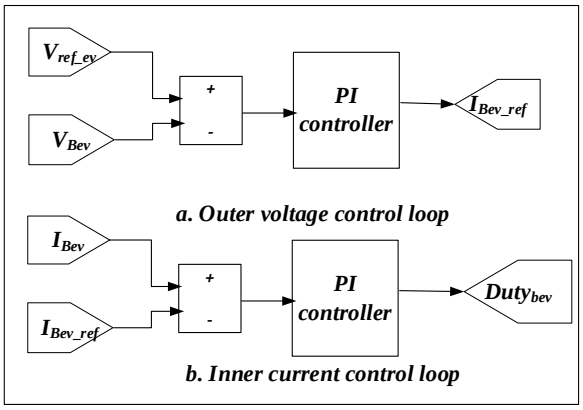


Figure 3.3 EV controller a) Outer voltage control b) Inner current control

3.2.2 System Formulation and Component Modelling

The formulation and modeling of different system components are presented as follows.

3.2.2.1 PV array

The mathematical representation of the one-diode equivalent circuit model of a PV system as of Fig. 3.4 is given by the transcendental equation [85] as in (3.1).

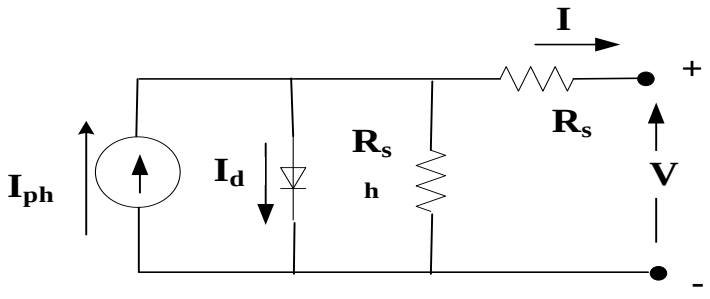


Figure 3.4 One diode equivalent circuit of PV module.

$$I = I_{ph} - I_0 . \left\{ e^{\frac{q}{N_{cs} . k . T . \gamma} . (V + I . R_S)} - 1 \right\} - \frac{V + I . R_S}{R_{Sh}} \quad (3.1)$$

Where I and V are the current and voltage of the PV module, respectively. N_{cs}, k, T, and q denote the number of cells in series, Boltzmann constant, cell temperature, and elementary charge, respectively. The free model parameters are photocurrent I_{ph}, diode saturation current I_0, series resistance R_S, shunt resistance R_{Sh}, and the diode ideality factor γ. The photocurrent depends on irradiation G and temperature T as in (3.2) [86].

$$I_{ph}(G, T) = \frac{G}{G_{ref}} . [I_{phref} + \mu_{I_{SC}} . (T - T_{ref})] \quad (3.2)$$

Where G_{ref}, I_{phref}, and T_{ref} are the irradiance, photocurrent, and temperature at some arbitrarily chosen reference conditions with $\mu_{I_{SC}}$ representing the temperature coefficient of I_{SC}.

3.2.2.2 Boost Converter

The boosting of voltage in the boost converter depends on the duty ratio D_b as in (3.3) [87]. Also, the boosting parameters L and C can be further calculated as (3.4) and (3.5), respectively.

$$V_{DC} = \frac{1}{1 - D_b} V_{PV} \tag{3. 3}$$

$$L_{boost} = \frac{V_{DC}.D_b}{\Delta I_L f} \tag{3. 4}$$

$$C_{DC} = \frac{V_{PV} D_b}{R_0 \Delta V_{PV} f} \tag{3. 5}$$

Where V_{PV}, V_{DC}, ΔI_L, V_{PV}, R_0 and f is the input voltage from PV, the output voltage of the converter, inductor ripple current, capacitor ripple voltage, the output impedance of boost converter, and switching frequency, respectively, in Fig. 3.5a.

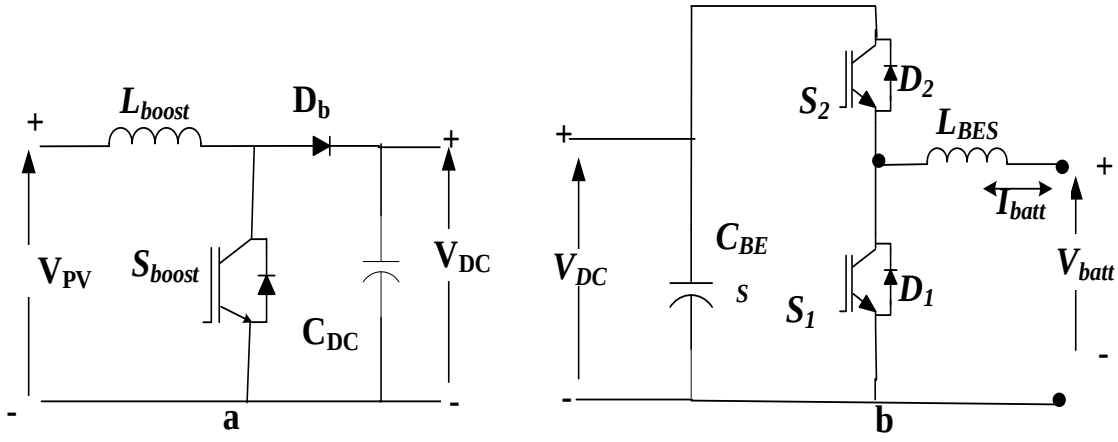


Figure 3.5 a) Boost converter b) Bidirectional DC-DC converter.

3.2.2.3 Bidirectional DC-DC converter

The inductance and capacitance for buck mode of BDC, as in Fig. 3.5 b, while charging is given by 3.6 and 3.7.

$$L_{buck} = \frac{(V_{DC} - V_{batt}).D_{buck}}{\Delta I_L f} \tag{3. 6}$$

$$C_{buck} = \frac{(1 - D_{buck})V_{batt}}{8L_{buck}\Delta V_{batt} f^2} \quad (3.7)$$

Similarly, for discharging mode, i.e., the BES charging EV controller should act as a Boost converter and follow the (3.4) and (3.5).

3.2.3 5G Architecture

The detailed security architecture is already presented in section 2.7. The 5G communication has the following components.

3.2.3.1 EVCS

EVCSs are equipped with long-term evolution new radio user equipment (LTE-NR-UE) device type with a configurable mobility model set to be static. The application layer implements an open-flow protocol and SDN controller, while the transport layer has a configurable TCP protocol. The network layer uses IPV4 protocol with a user-defined IP address, subnet mask, and default gateway. The physical layer of UE is equipped with configurable height, transmission power, and beamforming gain, respectively, set to 1.50 meters, 23 dBm, and 0 dB. The same configuration goes UE with rogue EVCS with spoofed IP.

3.2.3.2 gNB

Next-generation node B (gNB) is the 5G wireless base station that communicates with UE and the 5G core network. The gNB height is 10 meters with a transmission power of 40 dBm. The wireless communication between UE and gNB is time division duplexing (TDD) with 15 kHz subcarrier spacing, the outdoor scenario of rural macro, and channel characteristics with no path loss. The gNB is set to have a round-robin scheduling type with a UE measurement report of 120 milliseconds for experimentation.

3.2.3.3 EPC

Evolved packet core (EPC) was introduced in 4G LTE and had a core network functionality.

3.2.3.4 SCADA server

This wireless node is the SCADA system that continuously gets logs of electrical signals from EVCS and sends control commands to manage the controllers at EVCS. This setup is designed to communicate between the EVCS and the remote control station at SCADA. The SCADA continuously monitors the operation of EVCS and issues the control commands through the 5G network. These experimental setups start with no attack scenario, i.e., a normal operating condition of a 5G communication link, and the number of attackers increases from 0 to 15. The NetSim simulation time is set to run for 200 seconds to observe the delay.

3.3 Cybersecurity Issues in EVCS

The communication system is the nerve of the EVCS that facilitates various operations such as EV scheduling, slot allotment, authentication and authorization, charging session control, grid integration, and on and on [33]. Therefore, the entire EVCS risks disruption and dismantling once the communication is compromised. The future grid is envisioned to handle bidirectional power flow, blockchain-assisted peer-to-peer energy transactions, and vehicle-to-everything (V2X) communication [88]. The proper communication technology like 5G should moderate this odd marriage of evolving technology and the traditional grid infrastructure. Once the critical infrastructure is exposed to the open cyber layer through communication links, It is no more secure [38]. The communication vulnerabilities can be exploited to access the SCADA or EVCS system.

An attacker may use social engineering, such as phishing and/or reverse engineering, to get legitimate SCADA's or EV's credentials. Then, the attacker can impersonate the legitimate SCADA operator or EV owner to breach the system's security [33].

3.3.1 Threat Landscapes of 5G enabled EVCS Cyber-Physical System

Threat actors first identify the 5G assets' vulnerabilities and exploit them by assessing the attack surfaces. Table 3.1 depicts the CIA triad of 5G assets indicating the assets' risk [82].

Based on the works [20], [38], [39], [50], any cyber-physical security threats can be classified into four categories: Nefarious act/Abuse, Eavesdropping/ Hijacking/Interception, Intentional and/or Accidental damages, and Outages. The first two are ill-willed malicious actions generally targeted in cyberspace, while the latter two are threats to physical security. Nefarious activity/ Abuse targets the ICT infrastructures to steal, alter, or destroy the target. Eavesdropping/Hijacking/Interception targets unauthorized communication links to listen, seize, or interrupt the services. Intentional/ unintentional damage is intentional/unintentional action that causes damages/harm to the physical infrastructures and persons. Outages are the category that disrupts the availability and quality of service. Threats from nefarious activity/abuse are the most prominent and damaging threats for 5G and EVCS infrastructures. Some of them are listed below:

Table 3.1. CIA triad of 5G assets

Asset	**C**	**I**	**A**
MANO			
Network products			
Interconnections			
Services			
Organizations			
protocols			N/A
Data			
Processes			

Red= Very high, yellow= high, green= medium

3.3.1.1 Denial of service

The prime target of DoS is to disrupt 5G/EVCS service availability. DoS can be triggered in

many ways, such as botnet/DDoS, flooding network components/base stations, jamming/interfering with the radio frequency, replay, amplification attacks, etc.

3.3.1.2 Malicious code

The injection of malicious code into the software environment detriments and affects the system's processes, control actions, and operating conditions. Some examples are viruses, malware, rootkits, worms, trojans, rogueware, ransomware, and SQL and XSS injection attacks.

3.3.1.3 Exploitation

Most hardware and software systems have glitches or weaknesses. The attacker can exploit vulnerabilities in the architecture, design, and configuration of the network and software/hardware, such as zero-day exploits, open API, and edge API exploits.

3.3.1.4 Abuse

Since 5G-based EVCS is a highly complex, heterogeneous cyber-physical system with poorly developed administrative coordination and control, there is immense potential for abuse of remote access to the network, authentication/authorization, information leakage, virtualization, and even lawful interception.

3.3.1.5 Manipulation

An insider/outsider attacker can compromise/manipulate hardware equipment, control settings, data, and network resources. They might attempt MAC spoofing, memory scraping, side-channel attacks, fake nodes, rouge MEC gateway, and UICC format exploitation. Besides, there are always imminent threats from compromised vendors, spectrum sensing, data breach, unauthorized activities, identity theft/spoofing, and signaling storms/frauds.

3.4 Cyberattack Modeling

The research has the following assumptions: to initiate the DoS attacks, hackers used the spoofed IP of legitimate EVCS. The channel loss of the 5G network is set to zero.

3.4.1 FDI Attack Modeling

A N_m-dimensional measurement vector y of any nonlinear EVCS system function H depends on N_n -dimensional system state variable vector x with normally distributed N_m-dimension measurements error vector e as in (3.8) below [89].

$$\mathrm{y} = \mathrm{Hx} + \mathrm{e}; \quad \mathrm{e} \sim \mathrm{N}(0,1) \tag{3. 8}$$

$$\hat{\mathrm{x}} = \arg\min_{\mathrm{x}} \mathrm{J(x)} = (\mathrm{y} - \mathrm{Hx})^{\mathrm{T}}\mathrm{W}(\mathrm{y} - \mathrm{Hx}) \tag{3. 9}$$

The overdetermined system (where $N_m > N_n$) may not have an exact solution. Therefore, the attacker estimates the system variable $\hat{x}$ by using lightweight optimization algorithms such as mean square error, least square error, or log-likelihood. This work uses weighted least square error over the residual function $J(x)$ as in (3.9). Where W is the weight matrix and defined as $diag\{\sigma_1^{-2}, \sigma_2^{-2}, \ldots, \sigma_{N_m}^{-2}\}$ and σ_i^2 is the variance of i^{th} measurement. The y is identified as FDI if it exceeds the predetermined residual threshold (Euclidean norm) [90] τ as in (3.10).

$$\mathrm{J}(\hat{\mathrm{x}}) = (\mathrm{y} - \mathrm{H}\hat{\mathrm{x}})^{\mathrm{T}}\mathrm{W}(\mathrm{y} - \mathrm{H}\hat{\mathrm{x}}) > \tau \tag{3. 10}$$

$$\mathrm{y_a} = \mathrm{y} + \mathrm{a} = \mathrm{Hx} + \mathrm{a} \tag{3. 11}$$

Let y_a, in (3.11), be the measurement vector under the FDI attack vector $\boldsymbol{a}$ having the same dimension as y_a. The attacker can access the logged data y and limited state variable x during the reconnaissance phase of the attack.

The attacker can choose the distribution of the attack vector $\boldsymbol{a}$ randomly or based on some heuristics. The more sophisticated and stealthy attack can be launched without being caught, but

the impact may not be enough to disrupt normal operations. The stealthy false data $\hat{x}_a$ can be estimated using some nonlinear functions g as (3.12) with the help of (3.11). The stealthy attack's objective is to get the attack vector $\boldsymbol{a}$ that maximizes the error injected into the system without exceeding the detection threshold of τ, which is the constrained optimization problem as in (3.13).

$$\hat{\mathrm{x}}_{\mathrm{a}} = \mathrm{g}(\mathrm{y}_{\mathrm{a}}) = \mathrm{g}(\mathrm{y} + \mathrm{a}) \tag{3. 12}$$

$$\max_{\mathrm{a}} \left|\left|\hat{\mathrm{x}}_{\mathrm{a}} - \hat{\mathrm{x}}\right|\right| \text{ subject to } (\mathrm{y} - \mathrm{H}\hat{\mathrm{x}})^{\mathrm{T}}\mathrm{W}(\mathrm{y} - \mathrm{H}\hat{\mathrm{x}}) < \tau \tag{3. 13}$$

Based on the above background, the attacker can launch an FDI attack at three different controllers: PV controller on duty cycle, BES controller on V_{ref_bus}, and EV controller on V_{ref_ev}. The attacker can solely control the duration of the attack and the distribution of false data. The eqs. (3.14)-(3.16) represent the FDI attack vector for PV control, BES control, and EV control, respectively. The PRN (0,1,10) stands for a pseudorandom number that fluctuates ten times between the lower bound of 0 and the upper bound of 1. The reason for choosing PRN is completely heuristic-based, as the duty cycle ranges within this limit. Similarly, the attack injection at BES and EV follows the Gaussian distribution (G) with respective mean and variance as in eqs. (3.15) and (3.16).

$$\widehat{\mathrm{D}}_{\mathrm{a}} = \mathrm{D} + \Delta\mathrm{D}\ ;\ \Delta\mathrm{D} \sim \mathrm{PRN}\ (0,1,10) \tag{3. 14}$$

$$\widehat{\mathrm{V}}_{\mathrm{ref\_bus_a}} = \mathrm{V}_{\mathrm{ref\_bus}} + \Delta\mathrm{V}_{\mathrm{ref\_bus}}\ ;\ \Delta\mathrm{V}_{\mathrm{ref\_bus}} \sim \mathrm{G}(48,10) \tag{3. 15}$$

$$\widehat{\mathrm{V}}_{\mathrm{ref\_ev_a}} = \mathrm{V}_{\mathrm{ref\_ev}} + \Delta\mathrm{V}_{\mathrm{ref\_ev}}\ ;\ \Delta\mathrm{V}_{\mathrm{ref\_ev}} \sim \mathrm{G}(24,10) \tag{3. 16}$$

3.4.2 DDoS Attack Modeling

It is considered that the remote SCADA station continuously monitors all three control stations through the 5G. Once the DDoS launched through the 5G core network, no signal would

reach the EVCS. The duration of signal loss depends on the communication delay of the 5G network. The SCADA issues a control signal $\{x_i\}_{i=0}^{N}$ at any timestamp i, with N being the total number of samples. If the communication delay caused by a DDoS attack in a 5G network is N_0, then the original control signal and delayed signal are presented in eq. (3.17) and (3.18), respectively.

$$\mathrm{x_{orig}(n) = x(n) = \{x_i\}_{i=0}^{N}} \tag{3. 17}$$

$$\mathrm{x_{del}(n) = x(n - N_0) = \{x_i\}_{i=-N_0}^{N-N_0}} \tag{3. 18}$$

$$x_{del}(n + N_0) = x_{orig}(n) \tag{3. 19}$$

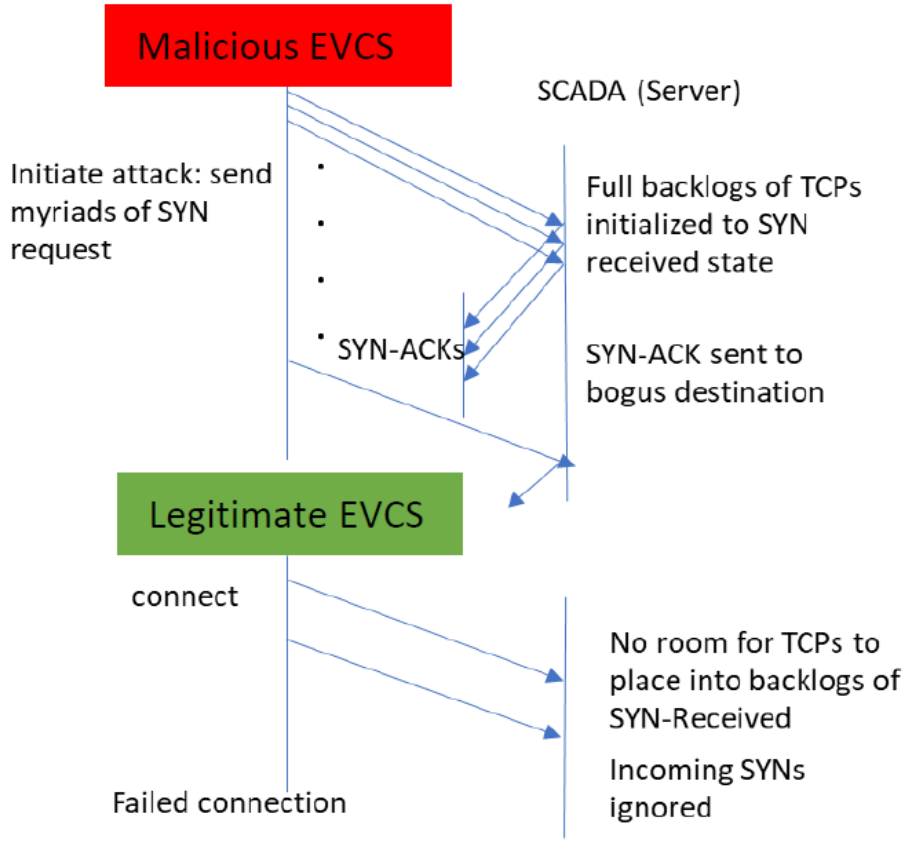
In other words, to get the same sample from the original control signal x_{orig}, we should add N_0 sample time to the current timestep to x_{del} as in (3.19).

The attack sequence goes like this: i) EVCS working normally, i.e., communicating normally with remote SCADA through 5G preattack as in (3.20a). ii) Suddenly, the DDoS attack starts at the sample time of $n_1 \geq 0$ and lasts up to $n_2 = n_1 + N_0 \leq N$ where N_0 is the variable delay that depends on the severity of the DDoS attack and comes from NetSim 5G simulation. At this time, the signal is completely lost, i.e., zero, as in (3.20b). iii) After the attack is gone, the signal should retain the $n_1 + 1$ signal sample as in post-attack of eq (3.20c) because DDoS should not compromise the signal integrity. The composite control signal reaching to the EVCS controller x_{EVSE} can be expressed as eq. (3.20).

$$\mathrm{x_{EVCS}(n)} = \begin{cases} \mathrm{x_{orig}(n)} \ \text{if } 0 \leq n < n_1, \text{pre attack (a)} \\ 0 \qquad \text{if } n_1 < n \leq n_2\text{, attack} \qquad \text{(b)} \\ \mathrm{x_{del}(n)} \quad \text{if } n_2 < n \leq N, \text{post attack (c)} \end{cases} \tag{3. 20}$$

3.4.3 DDoS Launch through 5G

TCP-SYN flood attack is the type of DDoS attack that exploits the vulnerability of a three-way handshake in the TCP protocol of machine-to-machine communication. The reasons to choose TCP protocol are: 1) it is the most ubiquitously used protocol along with IP to specify how data are exchanged over the internet by providing end-to-end communications. 2) It has known vulnerability in its three-way handshakes. 3) It is adopted by 5G communication. The attacker sends the TCP connection requests faster than the targeted machine can process, culminating in network saturation [91]. As shown in Fig. 3.6, The Attack can be summarized as follow: i) Client EVCS sends a TCP packet with an SYN flag on using a 5G network, ii) SCADA server, on receiving the SYN packet sends back an SYN-ACK packet to the client EVCS leaving half-open port for up to TCP connection timeout period. iii) EVCS acknowledges the SYN-ACK packet by sending an ACK to the SCADA server, and the communication starts. Before the half-open connections expire, malicious EVCS, either impersonating the legitimate EVCS or spoofing the IP, sends myriads of SYN requests to create many more half-open connections [92]. The malicious EVCS never receives SYN-ACK in spoofed IP and never sends ACK, coercing the SCADA server



to wait forever.

Figure 3.6 SYN-Flood Attack

3.5 Simulation Setups

This research work uses licensed NetSim Standard version 12.2 as a discrete event network simulator to simulate a 5G network through which an EVCS communicates with SCADA for command and control.

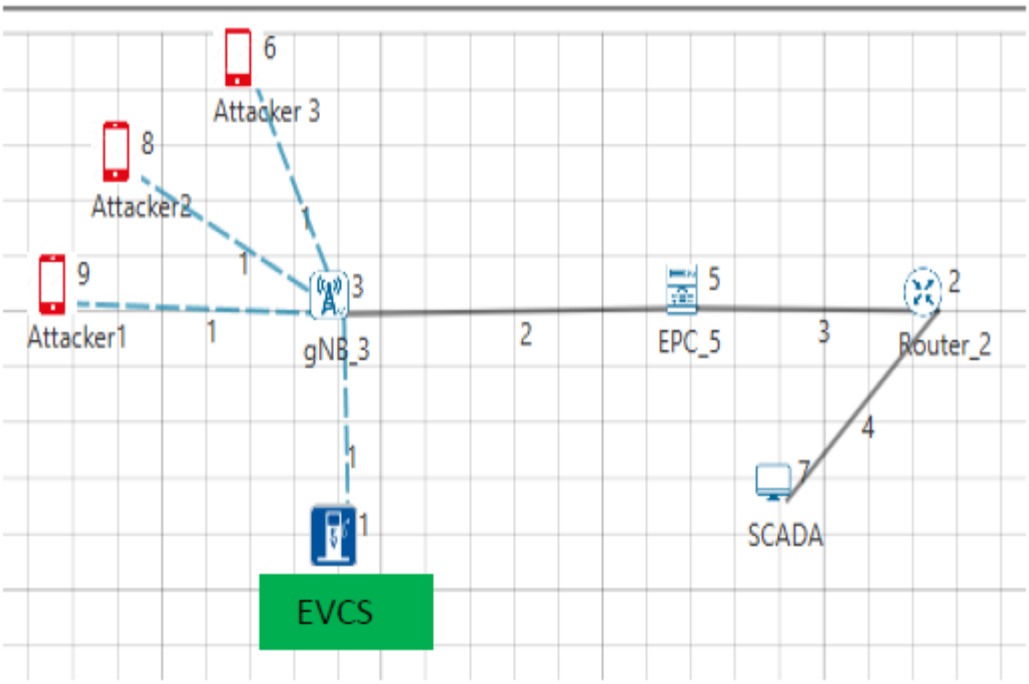


Figure 3.7 Snapshot of Network Architecture with three attackers in NetSim.

Each network component in a NetSim complies with Release 15/3GPP 38. xxx series and is flexible enough to specify user-defined longitude and latitude. The network components are summarized in section 3.2.3. Each smaller square box in Fig. 3.7 represents the area with 10-meter longitude and 10-meter latitude.

3.6 Simulation Results and Discussion

In this work, the Syn flood attacks are triggered in the 5G-enabled communication link between the SCADA node and the EVCS node. As evident from Table 3.2, The incremental change in delay with an increase in the number of attacks is more vehement. The throughput has been more consistent than latency, and the throughput drops to 20.94% compared to the base throughput. The latency has been increased significantly as the number of attackers increases from 0 to 15. The worst scenario of 509.476 ms (0.5 seconds) delay has been used to visualize the impact on the EVCS system.

Table 3.2. Network performance with increasing attack penetration

Malicious nodes	delay(ms)	Throughput (Mbps)
0	2.957	23.138
1	23.262	22.944
2	23.261	22.944
3	92.687	22.283
4	127.318	21.928
5	162.019	21.617
6	196.850	21.285
7	231.308	20.954
8	266.424	20.629
9	300.954	20.274
10	335.503	19.956
11	370.730	19.621
12	405.044	19.290
13	439.928	18.973
14	474.631	18.627
15	509.476	18.293

3.6.1 Impact analysis of FDI attacks

In this case, the simulation runs continuously for 15 seconds in Simulink, and attacks are launched at different controllers within the simulation time. The FDI attack analysis has been done in two scenarios to quantify the severity. i) Attacks are launched on different controllers at different times, and ii) Attacks are launched on different controllers simultaneously. The black plot in Fig. 3.8 represents the various electrical parameters during normal operation, while the red represents those parameters under the FDI attack.

3.6.1.1. FDI Attacks launched on different controllers at different times

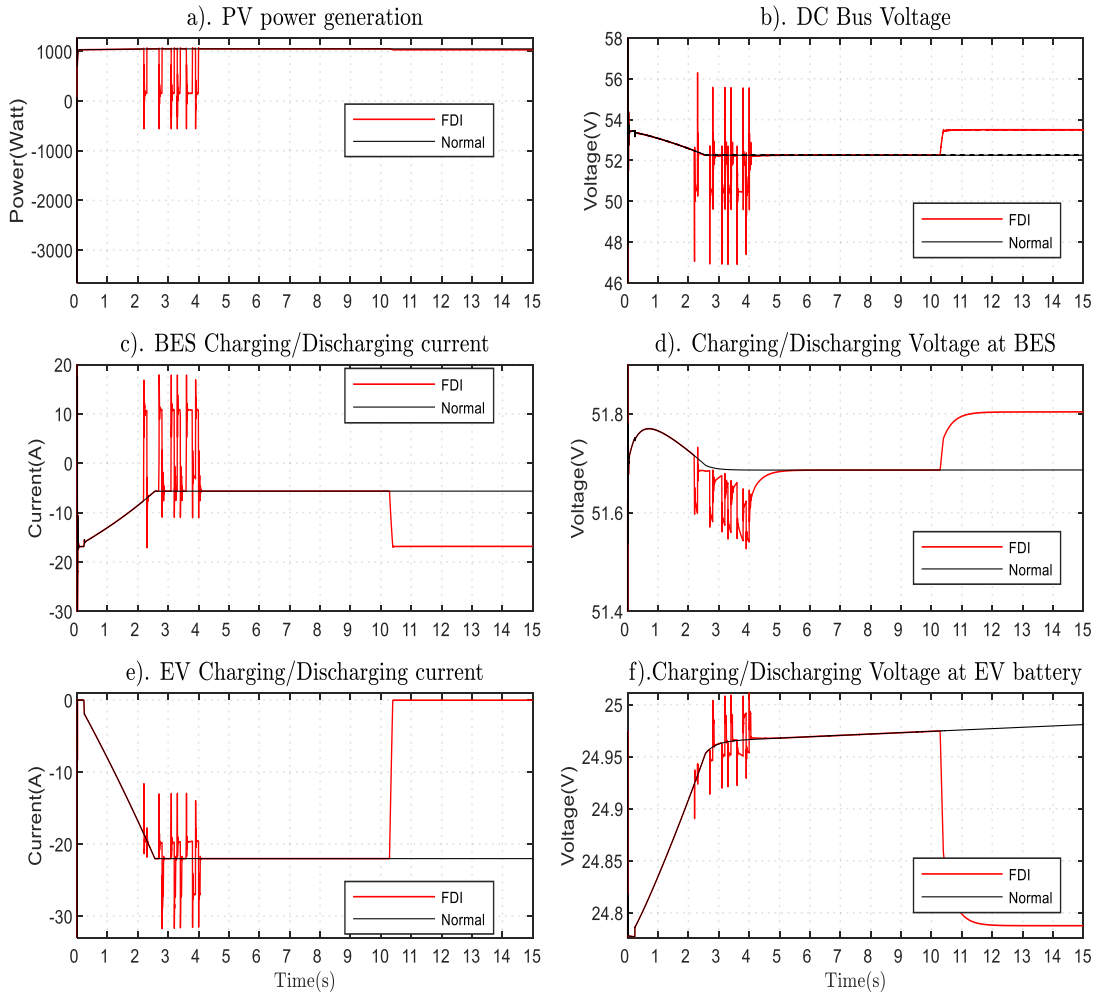


Figure 3.8 Impacts of FDI attacks launched at PV controller from 2-4 seconds, BES controller from 6-8 seconds, and EV controller at 10 -12 seconds.

A. PV controller attack

The duty cycle attack launched from 2 to 4 seconds at the PV controller has caused severe ripples in PV power generation, excursing the power level to -500 Watts, as shown in Fig. 3.8a. The ripples at the PV controller contribute to the oscillating voltage at the DC bus bar by $\pm$ 5 V, as in Fig. 3.8b. Further ahead, the ripples pass on to the BES and oscillate the current through the range of [-17, +17] A from the normal operating current of -5 A as in Fig. 3.8 c. At the same time, the BES voltage goes down and oscillates, as in Fig.3.8d. Similarly, these low-frequency oscillations severely impact the EV battery as the charging current has steep spikes and dips

ranging from -31 A to -13 A within 105 ms, as in Fig.3.8 e. Fig. 3.8 f shows the slight oscillations in voltage during the attack. Therefore, the FDI attack at the PV controller can destabilize the entire EVCS ecosystem, i.e., PGU, ESU, and a plugged-in EV. However, the system gains its normal operating conditions as the attack goes off at t=4 seconds.

B. BES controller attack

The FDI attack at the BES controller does not seem to have any significant impact as the finely tuned PI controller saturates the incoming fluctuation from t=6-8 seconds in Fig.3.8e.

C. EV controller attack

The EV controller attack starts at t=10 seconds and lasts for 2 seconds, as in Fig.3.8. This impact is irreversible and does not affect PGU, except it increases the bus voltage by 2V (Fig.3.8b). The attack has risen sharply the constant charging current at BES by 11 A (Fig.3.8c) with a slight DC shift of 117 mv (Fig.3.8 d). The attack on EV has shifted the constant charging battery current at -21 A to 0.268 µA (Fig.3.8e), assisted by a slight voltage drop (Fig.10f), i.e., forces the charging EV to stop charging completely.

3.6.1.2 FDI Attacks launched on different controllers simultaneously

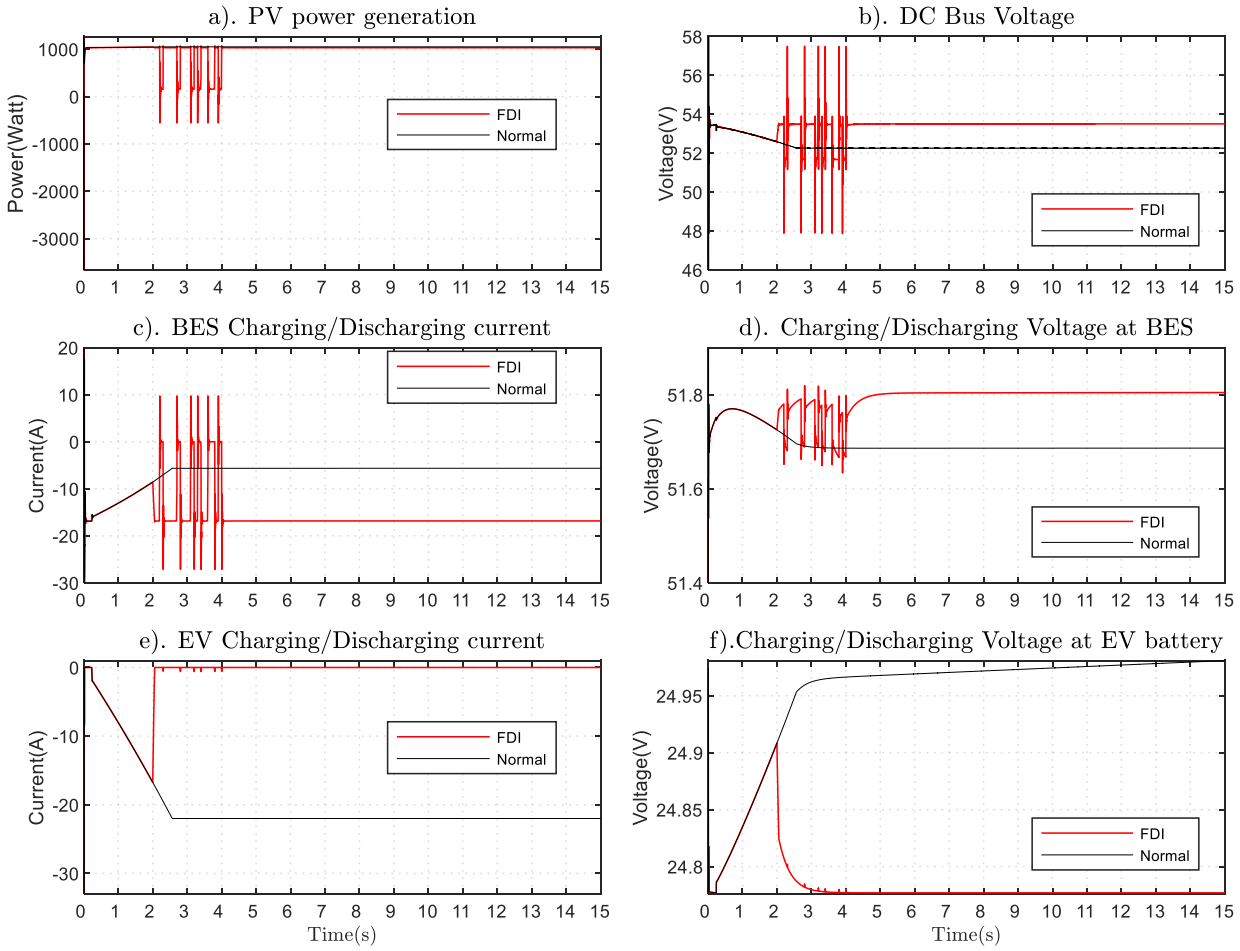


Figure 3.9 Impacts of FDI attacks launched at all controllers simultaneously from 2-4 seconds.

In this case, all the controllers are attacked simultaneously from 2 to 4 seconds, as in Fig.3.9. The integrated attacks resulted in reversible low-frequency oscillations throughout the attack and irreversible DC shift. The magnitude and frequency of oscillation at power generation remain the same Fig.3.9a. The DC bus voltage has spikes of equal magnitude and shifted up by 2V, as shown in Fig.3.9b, as opposed to the standalone attack at PV control.

The variation of BES current ranges from [-25 A 8.5 A] from normal constant charging of -5.6 A. It signifies BES's frequent charging and discharging within a short period. The peaks are even, and the constant charging current has been increased by 300% even after the attack that never returns to normal mode, as shown in Fig.3.9c. The BES voltage oscillations are the same as the

standalone attack at PV except for an irreversible DC shift of 117 mV, as shown in Fig.3.9d.

The EV charging current is dropped to zero permanently from -22 A with slight oscillation around 0-2 seconds, as shown in Fig.3.9e. That means the EV has stopped charging. The minimal temporary oscillation has been observed in EV battery voltage with a slight irreversible drop shown in Fig.3.9f.

3.6.2. Impact analysis of DDoS attack

Under similar simulation setups as the FDI attack, different delays resulting from a cyberattack on 5G in NetSim have been tested in our EVCS system. Likewise, the DDoS attack is carried out in all three control components simultaneously and at different times. Fig. 3.10 presents the impact of DDoS attacks on the physical system caused by the 500 ms delay of the 5G communication system.

The attacker launches the DDoS attack on the PV controller at t=2 seconds and lasts for 500 ms. This attack causes high-frequency oscillations at the power signal that fluctuates between -3.6 kW to 1.277 kW, while the PV generation system was designed to deliver 1.065 kW, as shown in Fig.3.10a. The negative power means the PV is drawing the power from the EVCS. Similarly, these high-frequency oscillation causes momentary voltage swing and drop of 2V at the DC bus, as shown in Fig.3.10b. At BES, the current surges by 17 A, as shown in Fig.3.10c, and voltage drops by 200 mV, as shown in Fig.3.10d.

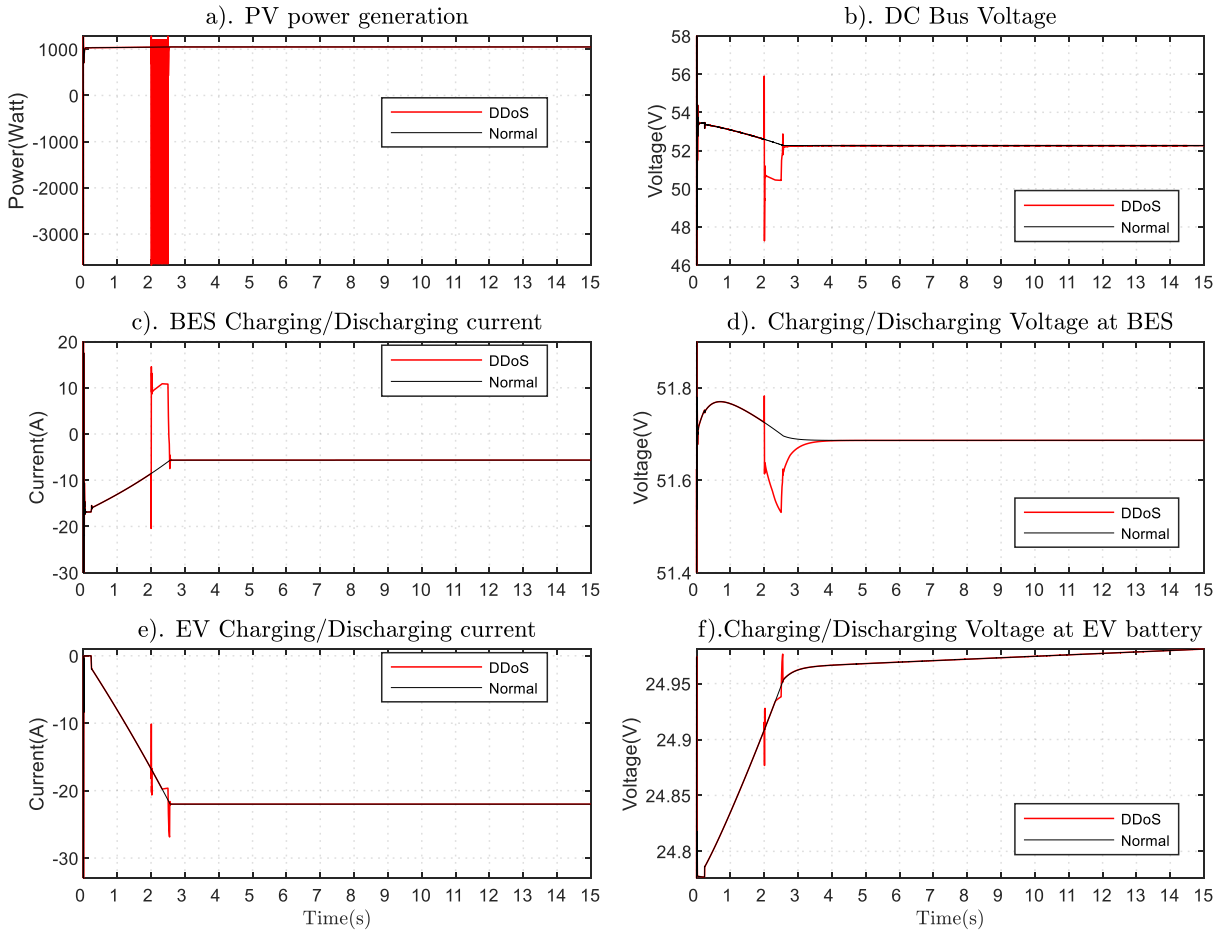


Figure 3.10 Impacts of DDoS attacks launched at PV controller from 2-2.5 seconds, BES controller from 6-6.5 seconds, and EV controller at 10 -10.5 seconds.

Similarly, spikes of -10 A and -26 A were observed at the beginning and end of the attack, respectively, in the charging current, as shown in Fig.3.10e, and are accompanied by small complementary voltage surges in the EV battery, as shown in Fig.3.10f. As soon as the attack stops, the system completely comes back.

The attacks at BES control and EV controller have no impact on system response. This is because the PI controller has fixed upper and lower saturation thresholds that do not let the manipulated V_{ref} signal to produce a zero-control signal, though the signal is completely lost throughout the attack. Our experiment suggests that the improperly tuned PI controller with no saturation thresholds is found to be exploited by the DDoS attack.

3.7. Chapter Conclusion

This work analyzes the cybersecurity issues in the 5G-enabled EVCS system with a deep learning-based attack detection method. The 5G enabled standalone off-the-grid PV-powered EVCS architecture has been built and simulated to charge the PEV. The FDI and DDoS attacks have been successfully launched and simulated in the 5G-powered EVCS communicating with remote SCADA, and consequent impact analysis has been presented. The following conclusions can be drawn from the study:

- The low-frequency FDI attack on the PV controller's duty cycle produces ripples and impacts all the subsequent components throughout the attack.
- The FDI attack at BES has no visible impact since the PI controller operates around the saturation region to cope with the attack.
- The FDI attack at the EV controller has resulted in an irreversible DC shift in operating current and voltage.
- The simultaneous FDI attacks at all controllers have an integrated impact of points previous three bullets.
- The DDoS attack at the PV controller has caused high-frequency oscillations at PV power generation and high-magnitude spikes and dips to subsequent Bus, BES, and EV controllers throughout the attack.
- The DDoS attack at EV and BES has no impact due to PI controller saturation.

In the following chapters, the study will focus on developing and designing a deep learning-based algorithm for Cyberattack detection and mitigation in EVCS.

Chapter 4 Cyberattack detection methods in the Electric vehicle charging station

4.1 Introduction

Timely detection of the attacks and their most accurate classification helps the EVCS operator take the appropriate prevention strategy against them, known as an IDS. Host-based (HIDS), network-based (NIDS), and hybrid IDS are mostly used IDS and classified accordingly in terms of their implementation location. Three basic intrusion detection techniques have been widely deployed in state-of-the-art applications, namely: Signature-based detection (SD), Anomaly-based detection (AD), and Stateful protocol analysis (SPA) [37]. SD-based IDS has to know the fingerprint (pattern) of the attacks beforehand to match the pattern of the incoming attack to the stored fingerprint. Therefore, they cannot learn new kinds of attacks. Also, the system admin must manually update the new attacks' fingerprints. AD-based IDS detects the anomaly by analyzing whether the incoming attack deviates from the network behavior. The mostly used IDS is AD-NIDS, which includes all the machine learning-based IDS.

Despite the capability to learn and detect new network attacks, it suffers from a high false alarm rate (FAR) and goes offline to rebuild the network behavior as it discovers the new attack type. Stateful protocol analysis (SPA), a.k.a. specification-based detection, on the other hand, extracts and crafts the correct behaviors of critical objects as security specifications and compares them against the actual behavior of the network [94]. The difference between SPA and AD is that the former compares the specification against standard security protocols, while the latter compares the behavior against the observed network behavior. SPA is resource-consuming since it has to trace and examine the protocol states. Moreover, it fails to inspect benign protocol behaviors and might not be compatible with the dedicated operating system (OS) and applications.

Mostly, researchers deal with the cyber-attack scenario in the entire smart grid, leaving out

the most vulnerable and critical infrastructure, the EVCS. In other words, there is a technical gap in the cybersecurity studies for the EVCS. This chapter presents the deep learning-powered IDSs for cyberattack and ransomware detection in the EVCS environment.

4.2 Performance Metrics

The detection algorithms for the EVCS cyber-attack detection need assessment metrics to evaluate their performance. The confusion matrix is the most ubiquitous matrix for the performance evaluation of the classifier, which is shown in Table 4.1 below.

Table 4.1 Confusion matrix

Anomaly class↓\Predicted class→	**Anomaly**	**Normal**
Anomaly	TP	FN
Normal	FP	TN

True positive(TP): correctly classified intrusion,

False-positive(FP): non-intrusive behavior wrongly classified as an intrusion,

False-negative(FN): intrusive behavior wrongly classified as non-intrusive,

True negative(TN): correctly classified non-intrusive behavior.

4.2.1 Accuracy

It estimates the correctly classified data out of all datasets. The higher the accuracy, the better the ML model. ($Accuracy \in= [0,1]$)

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

4.2.2 Precision

It estimates the ratio of correctly classified attacks to the number of all identified attacks.

Precision represents the repeatability and reproducibility of the model ($Precision \in= [0,1]$). The higher the precision, the better the ML model.

$$precision = \frac{TP}{TP + FP}$$

4.2.3 True positive rate/Recall

It estimates the ratio of a correctly classified anomaly to all anomaly data. A higher value is desired to be a better ML model and is given by: ($Recall \in= [0,1]$)

$$Recall = \frac{TP}{TP + FN}$$

4.2.4 F1-Score/Measure

It is the harmonic mean of precision and recall. A higher value of the F1 score represents the good ML model ($F1 - score \in= [0,1]$) and given by

$$F1 - score = 2 * \frac{Precision * Recall}{Precison + Recall}$$

4.2.5 False Positive (Alarm) rate

It estimates the ratio of normal data flagged as attacks to the total numbers of normal data. The lower the FAR, the better would be the ML model($FAR \in= [0,1]$) and is given by

$$FAR = \frac{FP}{FP + TN}$$

4.2.6 Receiving Operating Characteristics(ROC)

ROC represents the trade-off between the TPR (y-axis) and FPR (x-axis) for the different thresholds of FPR. The area under the ROC curve(AUC) is used as a comparison metric. Higher

the AUC, the better the ML model and is given by:

$$AUC = \int_0^1 \frac{TP}{TP + FN} \mathrm{d} \frac{FP}{TN + FP}$$

4.3 Deep Learning-based Network Intrusion Detection System for Electric Vehicle Charging Station

4.3.1 Proposed IDS Methodology

Based on the above background, we propose a novel deep learning-based IDS to deal with the DoS attacks in the EVCS. The proposed methods implement two mostly successful deep learning algorithms, namely: the deep neural network (DNN) and LSTM for the binary (DoS attack or not attack (benign)) and multiclass (four different classes of DoS attacks as well as benign class) classification of the DoS attacks in the CICIDS 2018 [95] dataset for EVCS scenario. The proposed DNN- and LSTM-based IDS in the EVCS network have proved to be at least 99% accurate for detecting the DoS attacks with better performance metrics, the latter being the more efficient in terms of precision and recall and F1-score.

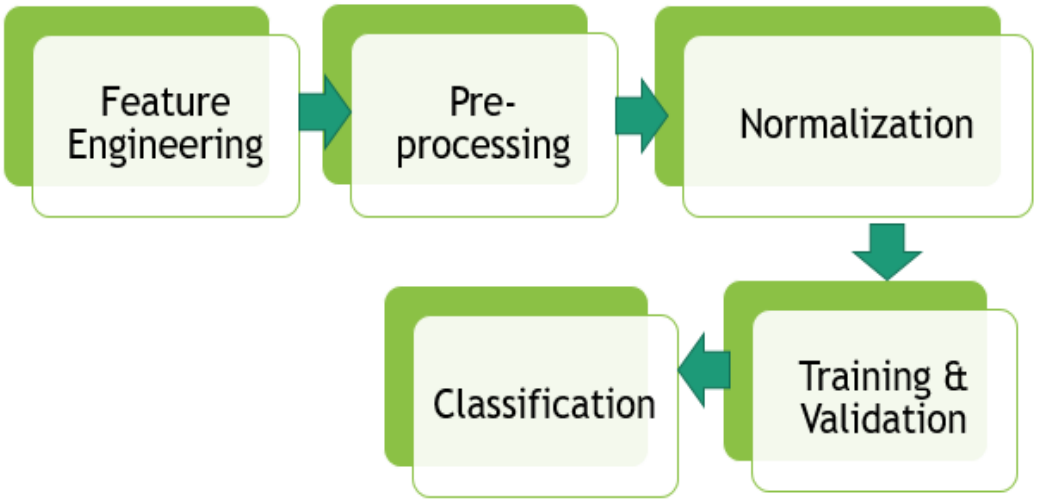


Figure 4.1 Steps involved in the Deep learning approach.

Each supervised deep learning algorithm has to go through some necessary steps, as shown in Fig. 4.1. Training all the features might not be viable due to the computation and storage resources limitation. Therefore, selecting the best training features would save time and computational complexity, called feature engineering. The principal component analysis (PCA), filter-based approaches such as IG filters, and auto-encoder-decoder are implemented in state-of-the-art [96], [97]. Most of the time, the feature sets might have categorical data, integers, and floats. Therefore, pre-processing and normalization generally convert them into a uniform format. Min-max scaling and one hot encoding have been implemented in the algorithm. DNN and LSTM algorithm uses 50% of the data for training, 20% for validation (to check the model's performance on unknown data), and the remaining 30% for testing. The data in training, validation, and classification are mutually exclusive.

The main goals of our algorithms are first to classify whether the incoming vector is an attack or a benign (Normal) data vector and second to classify the incoming data vector into different attack classes (4 DoS attack classes and one benign class). The former is best known as binary classification, and the latter is called multi-class classification. Furthermore, the third is to present a comparative analysis of applied algorithms.

4.3.1.1 Dataset

The mostly used datasets available online are KDDCUP 99 [98], NSL KDD [99], UNSW-NB15 [100], Kyoto [101], WSN-DS [102], CICIDS 2017 [103], and CICIDS 2018. The CICIDS 2018 DoS attack dataset is used for this research since it includes the recent DoS attacks. Table 4.2 represents the number of datasets belonging to different DoS attack classes.

Table 4.2 Datasets overview

Benign and Attack data	Numbers
Benign	1426795
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DoS attacks SlowHTTPTest	139890
DoS attacks-Hulk	461912

4.3.1.2 Deep Neural Network

A three-layered DNN with two hidden layers, each layer with 64 hidden neurons for the binary classification and 128 hidden units for multi-class classification, is implemented for IDS development in EVCS. Apart from that, the hidden layer uses the ReLU function since it has better convergence properties and prevents the problem of the sigmoid function, which tends to produce vanishing and exploding gradients. The de facto standard for the optimizer, Adam, is implemented in the DNN [104]. The only difference between the architecture of binary vs. multi-class DNN is the number of units in the hidden layer, output layer, and the corresponding activation function and loss function. The default activation function in the output layer and the loss: are softmax and categorical cross-entropy, respectively, for multi-class classification; sigmoid and binary cross-entropy, respectively, for binary classification. The proposed model implements the L1-L2 regularizers. These Regularizers apply penalties on layer parameters or layer activity during optimization, and these penalties are incorporated into the loss function that the network optimizes [105].

4.3.1.3 Long Short-term Memory (LSTM)

LSTM is the variant of the RNN developed to eliminate the vanishing gradient problem of RNN and is significantly more complex than traditional neural units. LTSM Cell Architecture: Each cell has four sets of weights that feed into it (instead of one). Output squashing can take any activation function we want, though. It learns 1). What/when to let something in, 2). When to

forget, 3). What/when to let something out. Most of the architecture is similar to DNN shown [65], except for the cell structure. A 76, 16, 16, 16, 1 architecture is used for binary classification, while 76, 32, 32, 32, 5 architecture is used for the multiclass classification with neuron dropout of 10 % between each hidden layer. The architecture mentioned above is read as # of input units=76, # of LSTM cells in first hidden layer= 32, # of LSTM cells in second hidden layer= 32, # of LSTM cell in third hidden layer= 3, and # of output units=5. The first and last layers are the input and output layers with corresponding nodes, while the middle layers represent the hidden layers with corresponding nodes.

4.3.2 Results and Discussion

In this work, all the simulations and codings are created in Python 3.7.4 in the Jupyter lab (version 1.1.4) under the free and open-source Anaconda distribution. Intel® Core™ i5-3470 @ 3.20 GHz processor with 8.00 GB RAM and 64 -bit Windows 10 OS is used in the experiment.

4.3.2.1 Plot-based Responses

Fig. 4.2 represents the variance captured by the singular values. Each singular value represents a prominent feature. The most prominent feature ranges from left to right. At least four features could represent 93.57 % of the variance. The more the variance captured, the more significant features will be. As shown in Fig.4.3., The PCA plot of the features in 2D showed that any linear classifier function could not classify the given datasets, so a deep neural network with hidden layers is the ultimate solution to classify the data

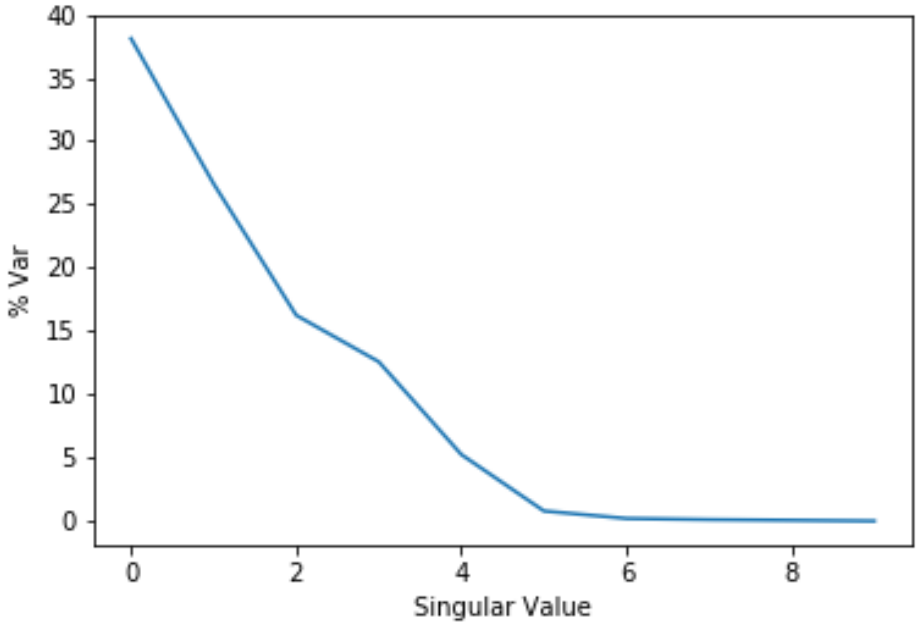


Figure 4.2 Variance captured by singular values.

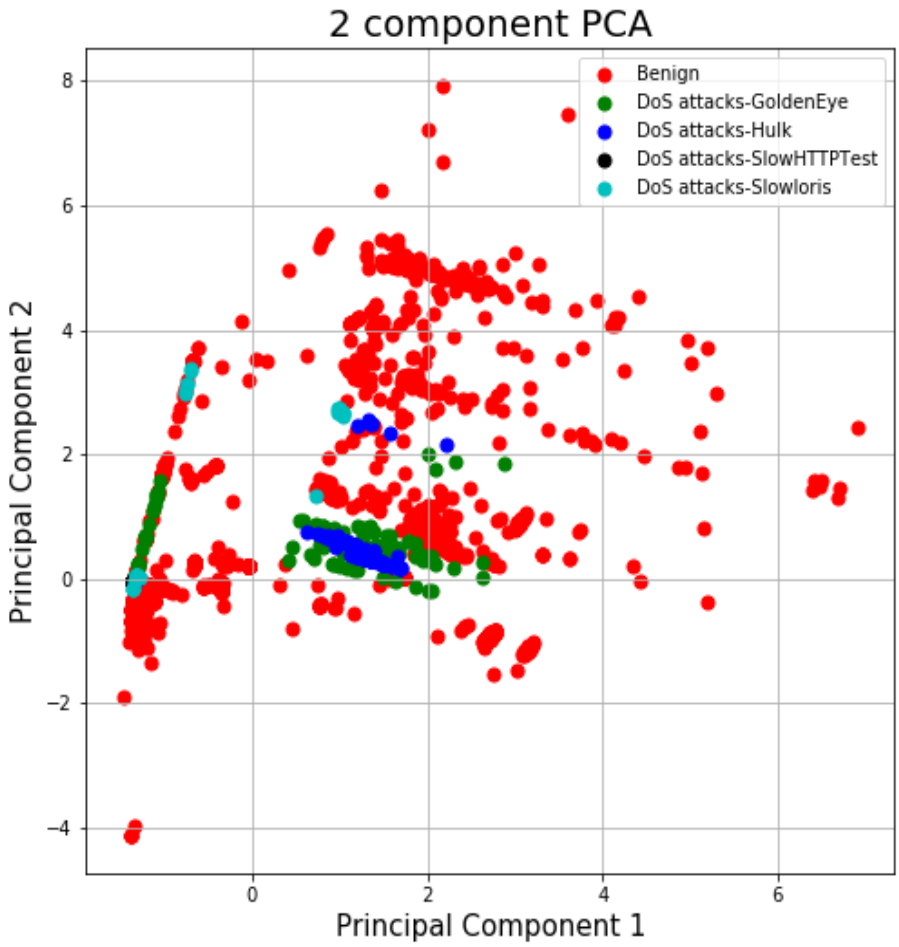


Figure 4.3 Two-component PCA plot.

As shown in Figs. 4.4-4.7, the 99% accuracy has been achieved within less than 10 epochs for the LSTM, while it took 30 epochs and 70 epochs, respectively, for binary and multiclass classification using DNN. It means LSTM is superior in terms of speed and accuracy as compared to DNN. Also, training and validation are smoother for LSTM as opposed to DNN.

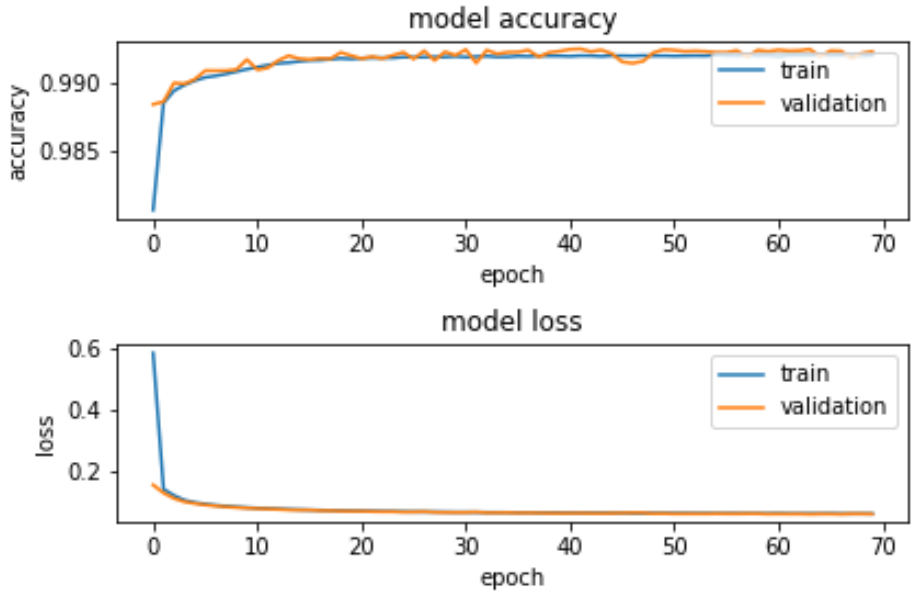


Figure 4.4 Model accuracy vs. model loss for the binary classification using DNN.

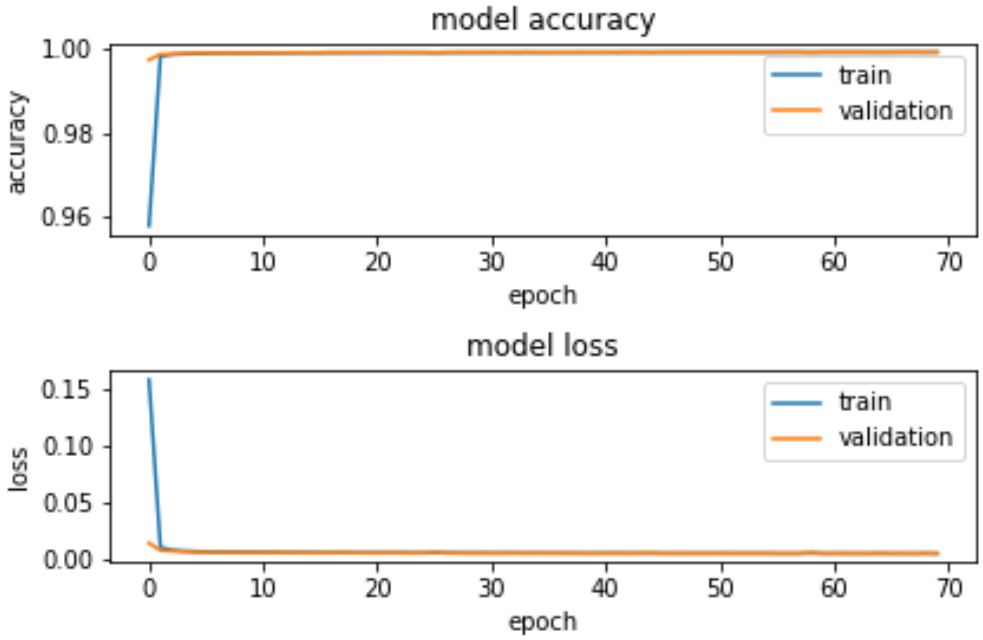


Figure 4.5 Model accuracy vs. model loss for the binary classification using LSTM.

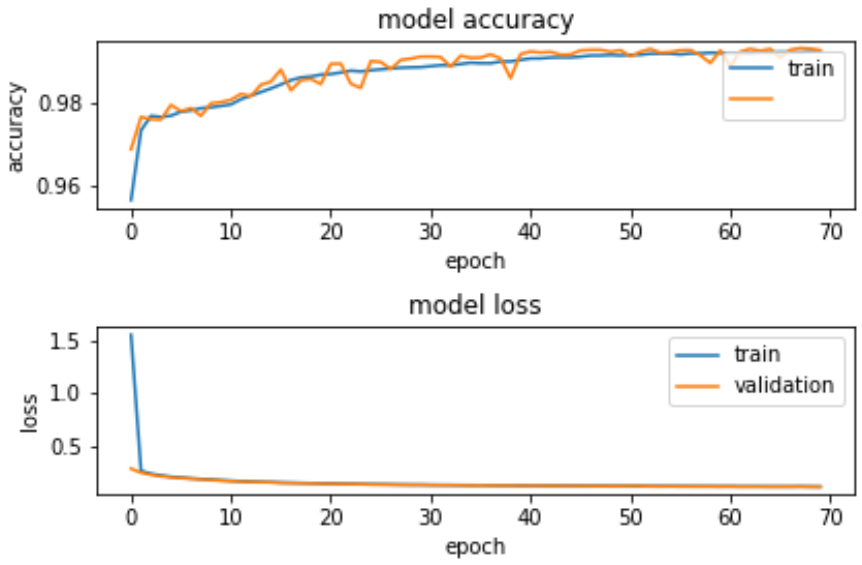


Figure 4.6 Model accuracy vs. model loss for the multiclass classification using DNN.

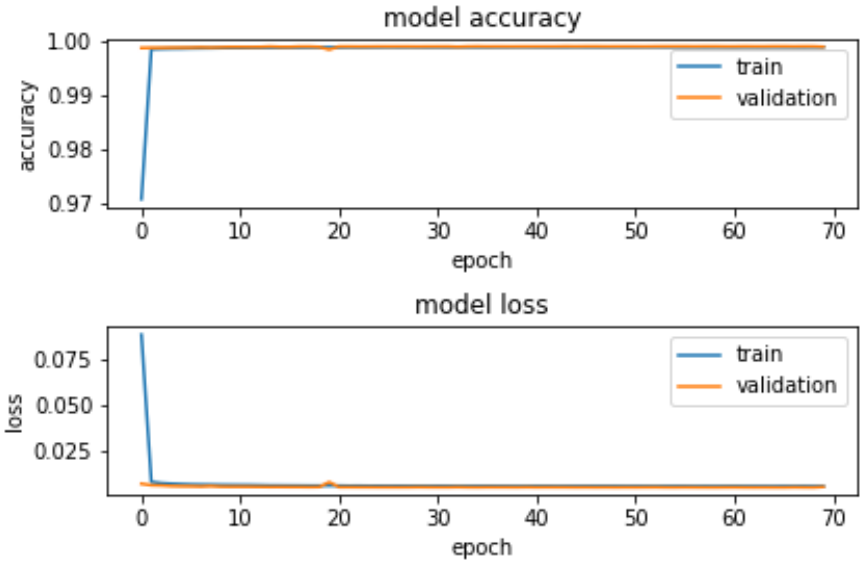


Figure 4.7 Model accuracy vs. model loss for the multiclass classification using LSTM

Fig. 4.8 gives the comparative training and testing accuracy for the binary classification of the data (whether the data is normal (benign) or the attack) using the DNN and LSTM- based IDS. Compared to DNN, The training and testing accuracy for the LSTM-based IDS are superior by 0.76 and 0.67, respectively. Similarly, as illustrated in Fig. 4.9, the training and testing accuracy

for the multiclass classification using LSTM is superior by 0.63 and 0.62, respectively.

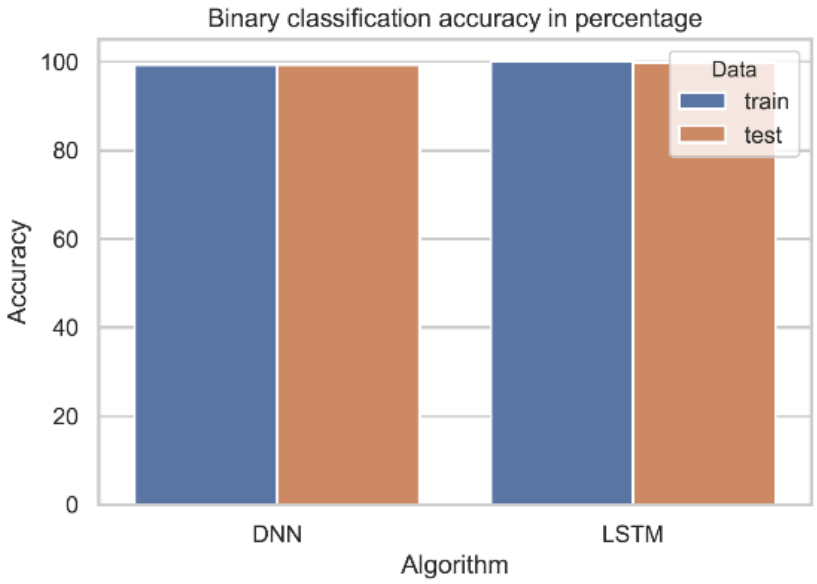


Figure 4.8 Binary classification accuracy.

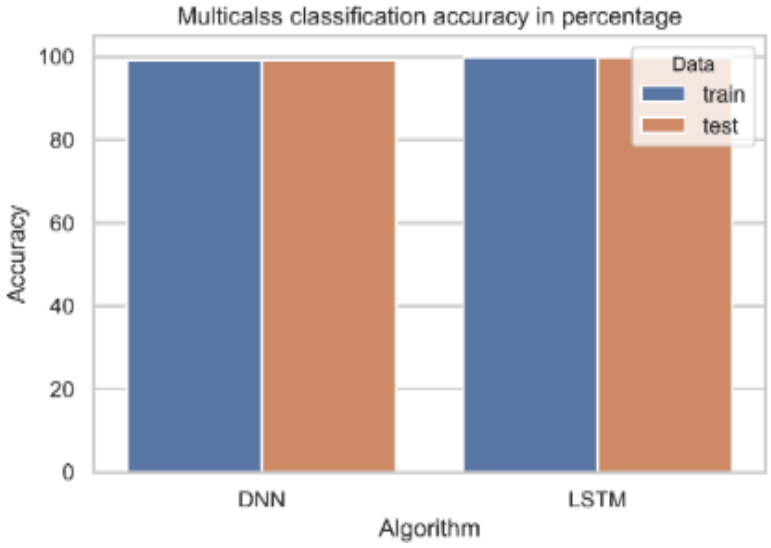


Figure 4.9 Multi-class classification accuracy

4.3.2.2 Performance Evaluation

In this work, to quantify the performance of the proposed detection method, some performance metrics have been considered, such as accuracy, precision, recall, and F-1 score (defined below) from the confusion matrix. The confusion matrix generally reflects how efficiently a particular machine/algorithm classifies the actual data, and it is the most ubiquitous matrix for the performance evaluation of the classifier. The perfect score of all three metrics (precision, recall, and F1-score) suggests that LSTM-based IDS has superior classification performance as compared to DNN-based IDS for the binary classification (attack or benign), as shown in Table 4.3.

Tables 4.4 and 4.5 represent the detailed performance metrics of our proposed classifiers (DNN and LSTM, respectively) with respect to each category (4 different attack categories and a normal/benign category) for multiclass classification. The support in the Tables represents the number of samples of true responses that lie in that class. As shown in Tables 4.4 and 4.5, the LSTM-based IDS is superior in multi-class classification compared to the DNN-based IDS as represented by higher precision, recall, and F1-score. Also, for the imbalanced classes and classes having comparatively less data (for instance: DoS attacks-Slowloris), the LSTM-based IDS exhibits good performance metrics with precision=1, recall=0.99, and F1-score of 0.99 as opposed to the DNN-based IDS with precision=1, recall=0.70 and F1-score= 0.82.

Table 4.3 Classification Metrics for DNN and LSTM for Binary classification

Algorithm	Data	Precision	Recall	F1-score
DNN_Binary	Attack	1	0.99	0.99
DNN_Binary	Benign	0.98	1	0.99
LSTM_Binary	Attack	1	1	1

LSTM_Binary	Benign	1	1	1

Table 4.4 Classification metrics of DNN for multi-class Classification

Attack Class	Precision	Recall	F1-score	Support
Benign	1.00	1.00	1.00	430408
DoS attacks-GoldenEye	0.99	0.80	0.88	12361
DoS attacks-Hulk	0.98	1.00	0.99	138678
DoS attacks-SlowHTTPTest	0.99	1.00	1.00	42022
DoS attacks-Slowloris	1	0.70	0.82	3268

Table 4.5 Classification metrics of LSTM for multi-class Classification

Attack Class	Precision	Recall	F1-score	Support
Benign	1.00	1.00	1.00	430977
DoS attacks-GoldenEye	1.00	1.00	1.00	12348
DoS attacks-Hulk	1.00	1.00	1.00	138126
DoS attacks-SlowHTTPTest	1.00	1.00	1.00	41959
DoS attacks-Slowloris	1.00	0.99	0.99	3327

4.4 Deep Learning-based Host Intrusion Detection System for EVCS

4.4.1 Proposed IDS Methodology

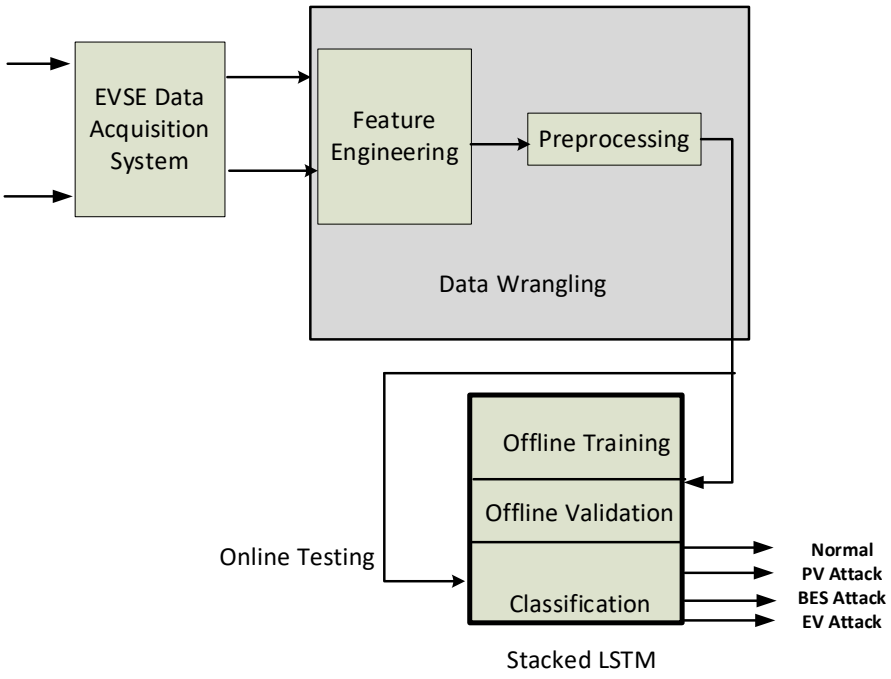


Figure 4.10 Proposed IDS at EVCS.

The proposed IDS is tested and validated with an FDI attack on our EVCS architecture. As shown in Fig. 4.10, the proposed IDS constantly records electrical and control signals (fingerprint) logs through the Data Acquisition System. Then the Data Wrangling block chooses, processes, and fetches the features into stacked LSTM in the compatible format. First, the output class for each data sample has been assigned. That can be done by appending {'0'}, {' 1'}, {'2'}, {'3'}, {' 4'} at the end of dataset, i.e., 37^{th} column of each class. Now, these strings in the output class have been converted to categorical data for classification purposes. The dataset can be split into two mutually exclusive sets: train (70%) and test (30%) class. 20% of training data were further split into validation data. Training data in deep learning is used to fit the nonlinear convex curve using input-output mapping based on forward and backward propagation. Validation data are generally

used for hyperparameters and model tuning. Testing data is used to see the generalization of the trained model. Our deep LSTM model is 3-layered with a 10% dropout between each stacked LSTM layer, and it has an output layer with four nodes. The deep LSTM architectures and parameters are shown in Table 4.6 below.

Table 4.6. stacked LSTM architecture

Layer(type)	Output Shape	Param #
LSTM_1	(None, None,64)	25856
Dropout_1	(None, None,64)	0
LSTM_2	(None, None,64)	33024
Dropout_2	(None, None,64)	0
LSTM_3	(None,64)	33024
Dropout_3	(None,64)	0
Dense_1	(None,64)	260

Each stacked hidden layer has 64 LSTM units. The model is compiled using the categorical cross-entropy loss function and Adam optimizer, which are de facto choices for multiclass classification. The model is trained for 20 epochs with a batch size of 1000 samples, which took almost 32.91 minutes.

The primary goal of the proposed system is to detect the different classes of the bypassed cyberattack on different components of the EVCS system using only the electrical fingerprint and making it autonomous against the cyberattack.

4.4.1.1 Data Set

We have collected 36 features from our EVCS architecture at a single timestamp. The features include all the electrical and control signals used in the operational EVCS system. The sampling time of data collection is set at $T_s = 10\ \mu s$, thus the sampling frequency is $f_s = \frac{1}{T_s} = 10^5\ samples/s$. Since the total simulation time is set to 15 seconds, the total number of samples

belonging to a single attack class is $15f_s = 15x10^5 = 1.5\ M\ samples$. For four different attack classes ('Normal', 'PV Attack', 'BES Attack', 'EV Attack') we have dataset {x} of sample size of [1500000, 36, 4]. Table 4.7 presents the electrical fingerprint used for IDS.

Table 4.7. Datasets overview for HIDS

Components	**Features**	**Total #**
PV panel	$i_{pv}, v_{pv}, p_{pv}, i_{diode}, del_{in}, T, irradiance$	7
MPPT Boost converter	$s_{boost}, \{i, v\}_{switch}, duty_{pv}, i_{bus}, v_{bus}, timestamps$	7
BES	$soc, i, v, s_p, s_n, i_{ref}, duty_{bes}, v_{ref}, \{i, v\}_{switch}x2$	12
EV	$soc, i, v, duty, i_{ref}, v_{ref}$	6
Diodes	$\{i, v\}x2$	4

4.4.1.2 Stacked/Deep LSTM

The LSTM is the generalized state machine and de-facto RNN primarily designed for the regression or classification of sequential data, i.e., sequence learning. Stacking these individual LSTM cells into the hidden layers forms the Deep LSTM. A single LSTM unit is much more complex than a traditional neural unit. It has four gates: input gate, output gate, forget gate, and cell gate [106]. The LSTM cell takes the input feature x_t along with cell state c_{t-1} and hidden state h_{t-1} from previous LSTM units and outputs the current cell state c_t and hidden state h_t.

The deep LSTM has stacked layers of multiple single LSTM cells with four gate parameters and an output parameter described below by (4.1)-(4.5) [107].

$$\text{Input gate parameters: } \begin{pmatrix} W_{xi} \\ W_{hi} \end{pmatrix} \in \mathbb{R}^{(D+H)X\,H}, b_i \in \mathbb{R}^H \quad (4.1)$$

$$\text{Forget gate parameters: } \begin{pmatrix} W_{xf} \\ W_{hf} \end{pmatrix} \in \mathbb{R}^{(D+H)X\,H}, b_f \in \mathbb{R}^H \quad (4.2)$$

Cell parameters: $\begin{pmatrix} W_{xc} \\ W_{hc} \end{pmatrix} \in \mathbb{R}^{(D+H)X\,H}, b_c \in \mathbb{R}^H$ (4. 3)

Output gate parameters: $\begin{pmatrix} W_{xo} \\ W_{ho} \end{pmatrix} \in \mathbb{R}^{(D+H)X\,H}, b_o \in \mathbb{R}^H$ (4. 4)

Network output layer parameters: $W_{hK} \in \mathbb{R}^{HX\,K}, b_K \in \mathbb{R}^K$ (4. 5)

Where $\boldsymbol{W}_{(.)}$ is the weight matrices, $\boldsymbol{b}_{(.)}$ is the bias vectors, $\boldsymbol{D}$ is the dimension of the input signal, $\boldsymbol{H}$ is the number of LSTM units, and $\boldsymbol{K}$ denotes the number of output classes. The respective outputs of the input, forget, cell, and output gates $\{i_t, f_t, \widetilde{c_t}, o_t\}$ in the forward pass can be written as follows in (4.6)-(4.10).

$$i_t = \sigma\left[\begin{pmatrix} W_{xi} \\ W_{hi} \end{pmatrix}^T [x_t, h_{t-1}] + b_i\right] \quad (4. 6)$$

$$f_t = \sigma\left[\begin{pmatrix} W_{xf} \\ W_{hf} \end{pmatrix}^T [x_t, h_{t-1}] + b_f\right] \quad (4. 7)$$

$$\widetilde{c_t} = \tanh\left[\begin{pmatrix} W_{xc} \\ W_{hc} \end{pmatrix}^T [x_t, h_{t-1}] + b_c\right] \quad (4. 8)$$

$$o_t = \sigma\left[\begin{pmatrix} W_{xo} \\ W_{ho} \end{pmatrix}^T [x_t, h_{t-1}] + b_o\right] \quad (4. 9)$$

$$h_t = o_t * \tanh(c_t) \quad (4. 10)$$

Where σ is a nonlinear function. The current LSTM outputs: cell and hidden states $\{c_t, h_t\}$ are passed to the next timestamps to iterate through the above equations. The probability vector $\{p_t\}_{k=1}^{k=K}$ for class *K* can be computed by using the SoftMax function as in (4.11).

$$p_t = \text{SoftMax}\left(W_{hK}^T h_t + b_K\right) \quad (4. 11)$$

$$\widehat{K} = \underset{k}{\text{argmax}}\ p_{tk} \quad (4. 12)$$

The predicted class $\widehat{K}$ would be the one with the highest probability at timestamp t, as shown

below in (4.12).

4.4.2 Results and Discussion

The deep learning algorithms are created in Python 3.7.4 in the Jupyter lab (version 1.1.4) under the free and open-source Anaconda distribution. Intel® Core™ i7-9750 @ 2.60 GHz processor with 16.00 GB RAM and 64 -bit Windows 10 OS is used.

Fig. 4.11 represents the accuracy and loss progression during the training and validation of the model. The proposed model achieves more than 99.9999% accuracy within the fifth epoch for training and validation data. The loss is diminished to 6.11e-06 within the fifth epoch for both training and validation datasets. This signifies a smoother progression during the training, and our model is ready to classify the previously unseen datasets of the attack.

The classifier's testing or generalization performance can be better presented with the confusion matrix, as shown in Fig.4.12 The distribution of test data among different attacks is almost equiproportional, i.e., $\cong 25\%$ from each class. The y-axis represents the predicted class, and the x-axis represents the actual class. The numbers and % in each cell represent the number of samples and % of samples belonging to that cell. For instance, for the first cell, among 449,234 normal samples fetched to our model. It correctly predicts 449,232 sample instances, leaving only one sample misclassified as a PV attack and another sample misclassified as BES Attack, which is almost 100% classification accuracy. Each row in the last column represents the total number of samples fetched to the classifier. Each column of the last row sums up the total number of classified samples belonging to different classes. Our classifier detects PV attacks, BES attacks, and EV attacks with 100% accuracy while detecting normal data with 99.99999% accuracy.

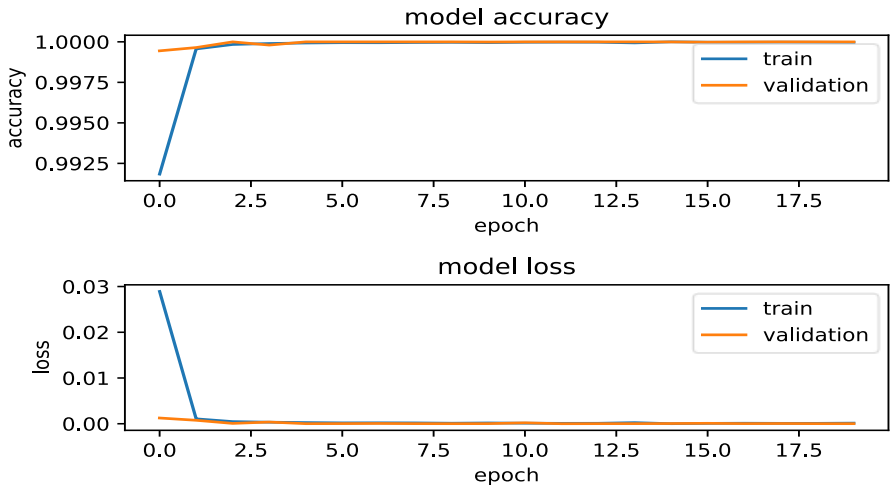


Figure 4.11 Accuracy and loss during Training and Validation progression

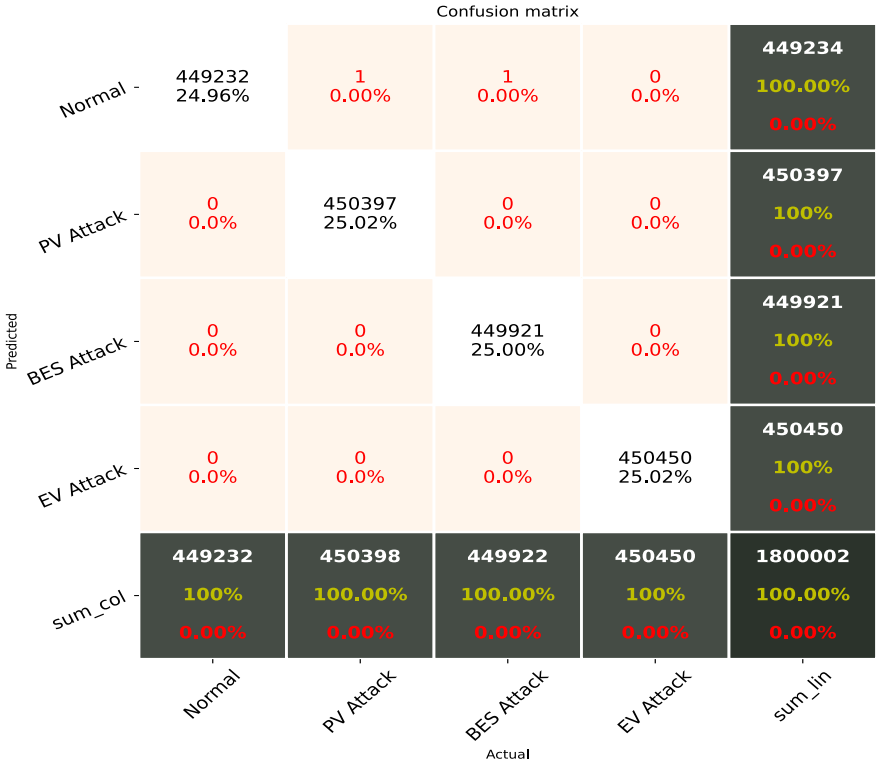


Figure 4.12 Confusion Matrix for assessing model performance

As per Table 4.8 below, the proposed classifier has almost 100% precision implying the model's repeatability, 100% recall implying the capability to correctly classify attacks among

different classes of attack, and a 100 % F1-score indicating superior sensitivity and separability of the model.

Table 4.8. Classification Metrics

Attack Class	Precision	Recall	F1-score	support
Normal	1.00	1.00	1.00	449232
PV Attack	1.00	1.00	1.00	450398
BES Attack	1.00	1.00	1.00	449922
EV Attack	1.00	1.00	1.00	450450

All the proposed IDS performance metrics are superior to our past IDS [33]. The most important capability of our proposed IDS is that it can detect the attempt of attack that bypasses the cyber layer and is not noticed in the monitoring station. For instance, the FDI attack at BES from 6-8 seconds, as shown in Fig 3.9, does not seem to change any electrical parameters, though the attack is there. But our IDS can detect it with 100% precision and recall, and f1-score does not misclassify a single sample. It's because of including the control signals as features.

4.5 Ransomware Detection using Deep Learning in the SCADA System of Electric Vehicle Charging Station

4.5.1 SCADA-Controlled EVCS

SCADA facilitates the management of remote access to real-time data and channels. It issues automated or operator-driven supervisory commands to remote stations (field devices) [108]. The underlying control system of most critical infrastructures, such as power, energy, water, manufacturing plants, traffic lights, and nuclear plants, is SCADA [109]. SCADA consists of sensors, Programmable Logic Controllers (PLC), actuators, Remote Terminal Units (RTU), a

supervisory station, a backend server, a human-machine interface, and a communication link. It needs constant vigilance of the target physical plant through the communication link. Sensors are primarily used for data acquisition in the plant's physical environment or monitored processes, which are embedded in the various dynamics of the process.

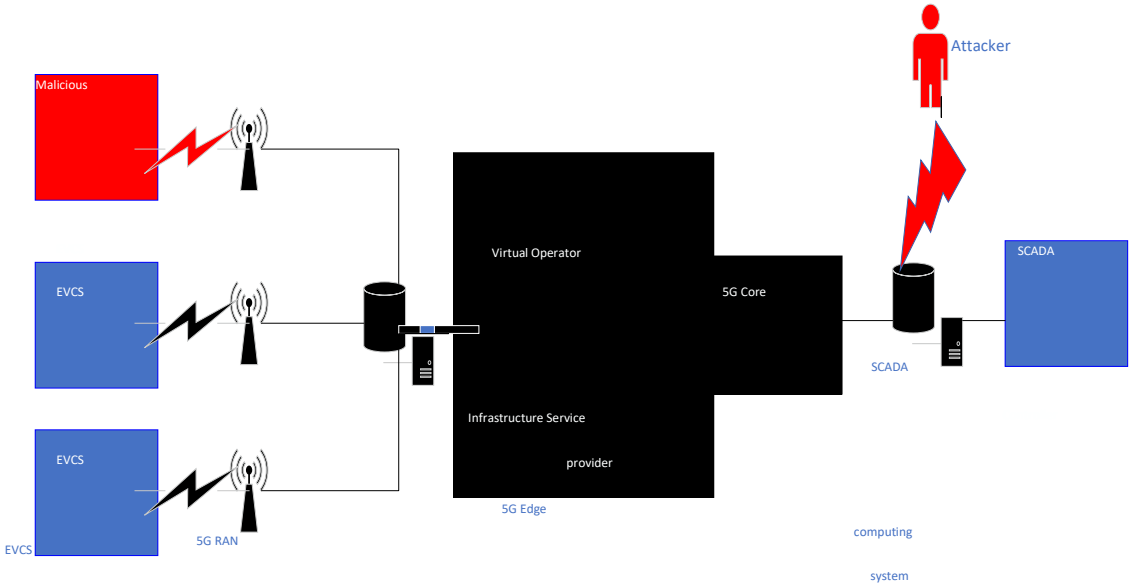


Figure 4.13 SCADA system connected to EVCS through 5G

Fig. 4.13 represents the high-level block diagram of remote SCADA communicating/controlling the multiple EVCS through 5G communication infrastructure. It also depicts the threat actors in the system. The SCADA constantly monitors the state of charge (SoC) of the battery energy storage (BES) at EVCS to control the charging.

4.5.2 Ransomware Attack Modelling on EVCS

Criminally motivated organizations such as WIZARD SPIDER run ransomware-as-a-service(RaaS) to target big game hunting [110]. An attacker encrypts the critical files: access

control files, historian data, input scripts, and the communication packet at SCADA. The attack motif may not necessarily be a full shutdown of the system but might interrupt normal operations. An operator may not pass the supervisory command or ride-over command to the field devices at EVCS. While the victim pays the ransom, it is not guaranteed that the hacker will give a decryption key. Once the attacker gains access to the cyber-physical system's critical files, there are two ways they can harm the operations first, by initiating a DDoS attack, and second by an FDI attack. Recent attacks [101]–[103] reported are relatively easy attacks to start. The severity of ransomware attacks on critical infrastructures depends on how long they captivate the network and how much false data they can inject.

We simulate the system in MATLAB Simulink, where the remote SCADA controls the charge and discharge of the BES of the EVCS by issuing charge/discharge commands to the ideal switches. The control commands are based on the SoC of the BES. The control is designed so that the BES should discharge, i.e., charge to EV if SoC > 80 % or charge from PV or DC source if SoC < 35 % between the timeframe of 50 seconds to 150 seconds. The ransomware-triggered DDoS on SCADA generally delays the control commands to/from the EVCS. We have simulated ransomware-induced delay ranging from zero to five minutes in the charging behavior. Secondly, we have simulated the effect of ransomware-triggered FDI attacks by manipulating SoC thresholds set to control charge/discharge.

4.5.3 Proposed Framework for Ransomware Detection

The prime target for ransomware could be any field devices (such as PLC, RTU, IoT devices for process dynamics control, and data acquisition), control, and monitoring systems such as supervisory stations and HMI. Reconnaissance of the vulnerabilities in the field devices might need domain knowledge and configuration in the air-gapped system. Also, poorly developed

protocols in field devices have easily exploitable authentication, authorization, and access control issues [111]. However, internet-facing field devices could be easily scanned and exploited using available scanning tools such as Shodan, ZoomEye, and Censys. The easy target might be HMI or supervisory station because they are the one who controls and monitors all field devices. That's why ransomware attacks in these computing systems that access the SCADA backend could be dangerous. The ease of attack comes here as these components could be treated more or less like the IT system. Besides, the attacker might not need in-depth design and domain knowledge, unlike in PLC and RTU. We propose the novel ransomware monitoring and detection system in Fig. 4.14. It monitors and detects ransomware attacks in SCADA, power generation/transmission and distribution networks, EVCS networks, and CAEV. The unit ransomware detection framework (RDF) architecture is presented in Fig. 4.15 for the SCADA. Likewise, RDF could be implemented for all remaining three layers beneath SCADA.

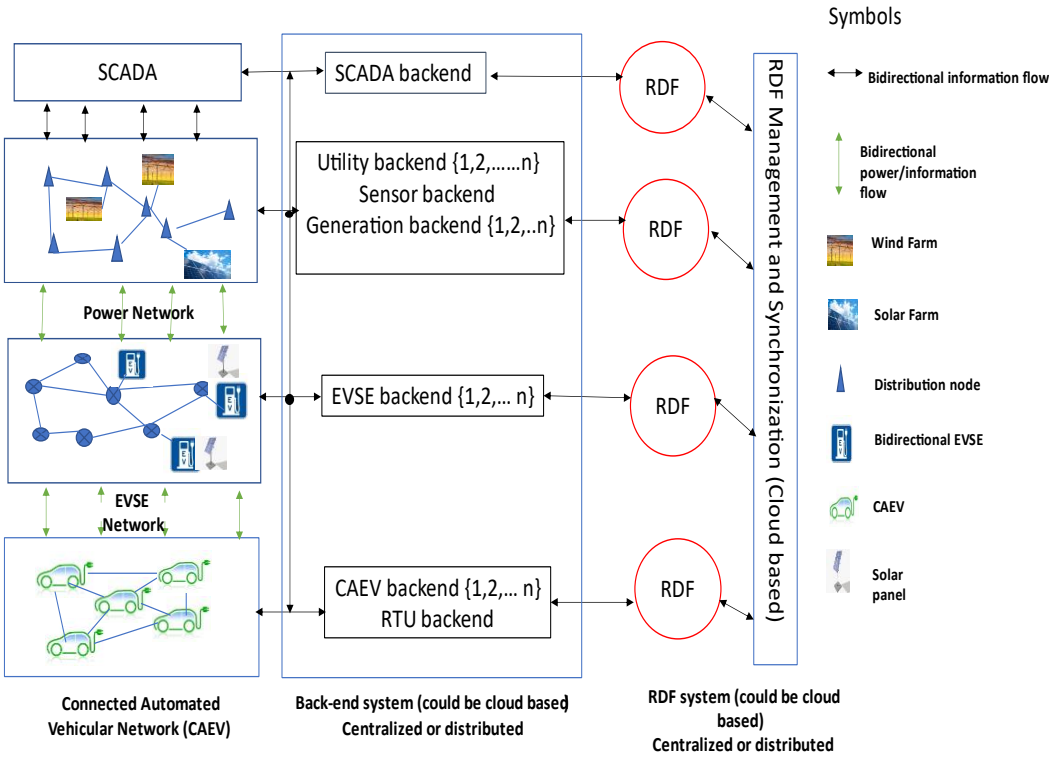


Figure 4.14 Proposed ransomware detection system for the smart grid architecture.

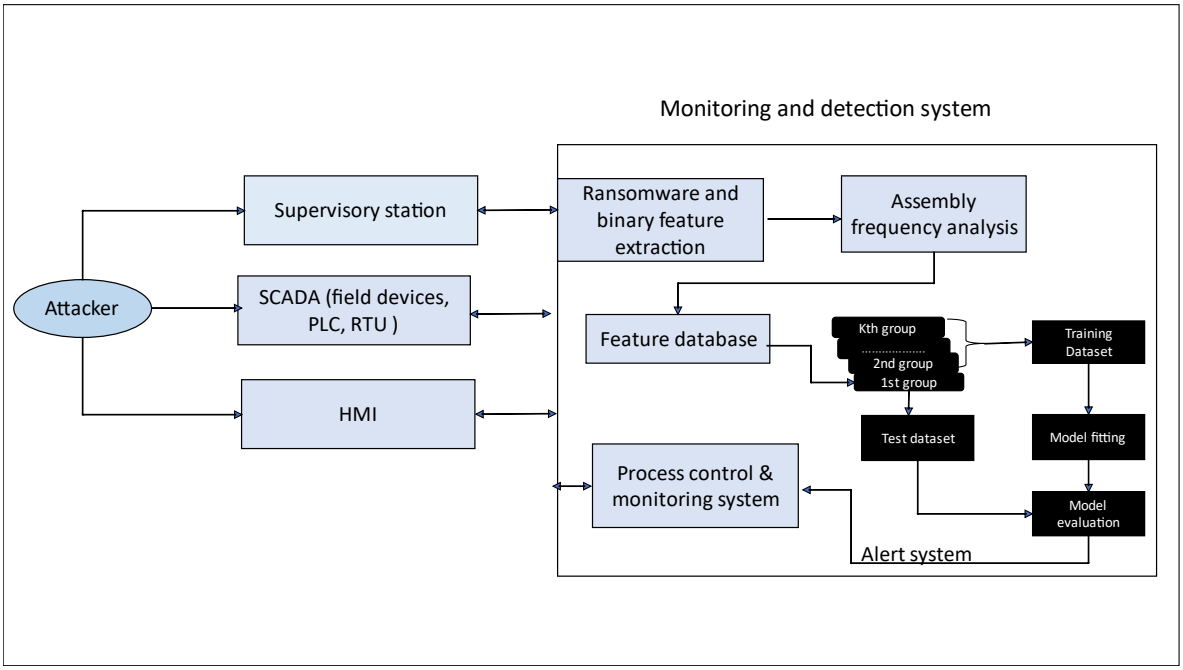


Figure 4.15 The internal architecture of the proposed RDF monitoring and detection system

The monitoring and detection system has two phases offline and online. In the offline phase, the ransomware and benign samples are collected from the attack. Important features are extracted from the collected samples by using assembly frequency analysis. Finally, the feature database is built to train and validate the deep learning-based model. Training is generally the curve fitting process based on weight optimization. Simultaneously, validation is how to tune the architectures and hyperparameters to get the best model before deploying it. Deep learning does not provide a sophisticated way to automate the model's tuning process. In the online phase, the feature from the real-time traffic is extracted using the same method as in training and is fed to the model to detect whether it is ransomware or a benign file. Once the ransomware is detected, an appropriate control strategy is activated to turn the backups on, isolate the physical layer; or shut down the system in the worst-case scenario.

4.5.3.1 Datasets

We have collected 561 ransomware samples and 447 normal samples, up to 1008. Ransomware binaries were collected from VirusTotal, whereas normal samples were extracted from the Windows operating system. The normal samples have similar sizes as ransomware (50KB – 8 MB), and files with cryptographic behavior (Filezilla, Winscp, OpenSSH, and so on) are also included. The ransomware considered here is Crypto ransomware. Feature selection is made by using frequency analysis of assembly instructions. Individual assembly instruction, including the grouping count, was taken. Assembly Grouping is grouped as Data transfer, Arithmetic, Logical, shift, and so on. A dynamic binary instrumentation technique was used to extract assemblies. It's a dynamic running of malware samples in a controlled virtual unit via the PIN framework. Further feature extraction was done extensively using custom python programming (Grouping, frequency generation, unique features, and CSV file generation).

4.5.3.2 Deep Learning Architectures and parameters setting for simulations

For the binary classification, a three-layered DNN with two hidden layers, each layer with 64 hidden neurons, is implemented for ransomware detection in SCADA. The ReLU activation function has been used because of its superior convergence property to the sigmoid function that comes with the problem of exploding and vanishing gradients. Adam, the de facto standard for the optimizer, is implemented [33]. The output layer uses the sigmoid activation function and binary cross-entropy loss for ransomware classification. The L1-L2 regularizers apply penalties on layer parameters or layer activity during optimization and are incorporated in the loss function.

LSTM is developed to eliminate the vanishing gradient problem of RNN and is much more complex than traditional neural units [40]. Each LSTM cell has four sets of incoming weights. Output squashing can take any activation function. LSTM shares a similar model architecture to

DNN except for the cell structure. The model has 140, 64, 64, 64, and 1 architecture for classification, with a neuron dropout of 10 % between each hidden layer. The architecture is read as # of input units=140, # of LSTM cells in first hidden layer= 64, # of LSTM cells in second hidden layer= 64, # of LSTM cell in third hidden layer= 64, and # of output units=1. The first and last layers are the input and output layers with corresponding nodes, while the middle layers represent the hidden layers with corresponding nodes. 1D CNN is the variant of the CNN designed to convolve 1D data vectors rather than convolving 2D or higher. Our model consists of two convolution layers with a kernel size of 64 and filter length of 3, each followed by a max-pooling layer. A fully connected hidden layer with 128 hidden units is between the max-pooling and output layers, with a unit dropout rate of 50%.

4.5.4 Simulation results and discussion

4.5.4.1 Simulation Setups

Before starting the simulation, data preprocessing was done to scale down diverse features with a magnitude between 0 and 1 using the Keras standard preprocessing. scale() library. Also, the categorical output classes ransomware vs. normal are binarized to 0 and 1, respectively. Out of 1008 samples, 30% of data are preserved for testing, the other 30% are preserved for validation, and finally, the model is trained with the remaining 40 % of the samples for a single run of the experiment. Each of these train, test and validation categories is mutually exclusive. Moreover, the training and validation are done with 10-fold stratified cross-validation to check the model's consistency and reproducibility. Our experiment is done with a batch size of 100 with 70 epochs for all deep learning algorithms.

With ransomware attackers having access to the critical process file of SCADA, they can hijack the system for the time they want and inject or manipulate the control variables. These

impacts are observed under the ransomware-driven DDoS attack trying to deprive legitimate users and FDI attack to decimate the EV charging.

4.5.4.2 Ransomware-driven DDoS attack

As mentioned in section 4.5.2, the charging behavior of BES is observed as the DDoS attack penetration increases from no delay to a delay of up to five minutes. The five SoC transition edges are recorded with a tuple of < SoC, time > at those edges between the control action period of 50 to 150 seconds; as shown in Fig. 4.16, The SoC0 represents the SoC without attack, i.e., normal behavior. SoCi represents the SoC behavior with i minutes of delay due to DDoS. With further data analysis as the SoC0 as a reference signal, the state transition due to the attack is delayed by a minimum of 0.139 % to a maximum of 4.84 %. This also forces SoC to exceed the control thresholds, potentially impacting the battery dynamics.

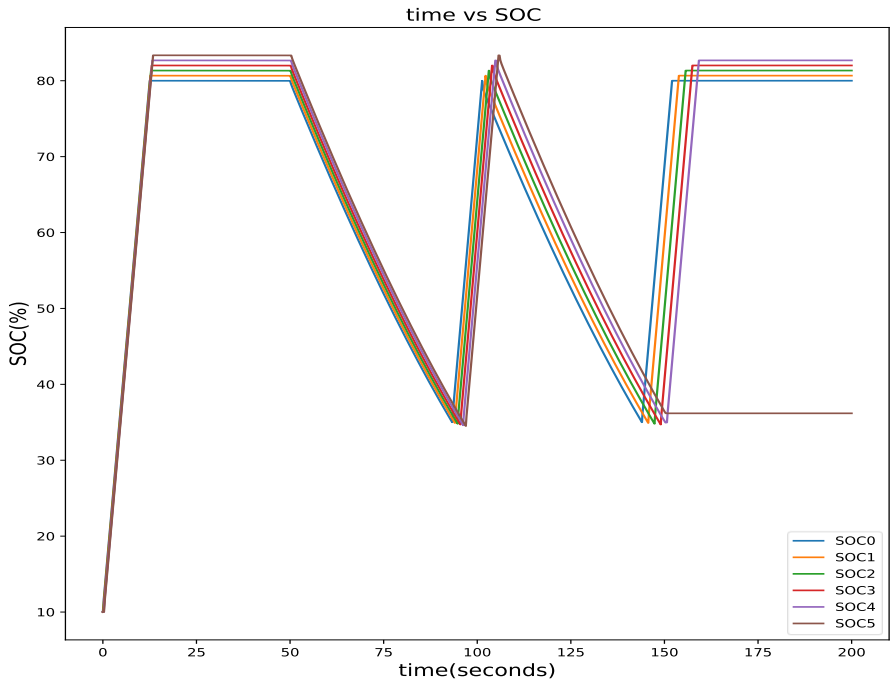


Figure 4.16 Ransomware-driven DDoS attack impact on charging behavior

The severity of SoC threshold crossing increases as the attack penetration increases from a minimum of 0.29 % to a maximum of -54.73 % compared to the normal operating SoC. With a five-minute delay induced by DDoS penetration, the charging profile becomes so worse that it just stays above the lower threshold of 35 % SoC without charging again after 150 seconds. Therefore, it can be concluded that even a simple DDoS driven by ransomware could produce erroneous control commands, which is enough to detriment the EV charging.

4.5.4.3 Ransomware-driven FDI attack

Similar architecture has been implemented to model the FDI attack, except the ransomware attacker manipulates the SoC thresholds to make control decisions at the state transition diagram of SCADA. The Fig. 4.17 represents the different SoC profiles S0C0, SoC1, SoC2, SoC3 with respective state transition thresholds tuples of (35,80), (10,90), (5,95), and (0,100). The tuple has a minimum and maximum value of SoC that should not be exceeded and issue either charging or discharging command at the instant of crossing them. The BES at EVCS has specific energy and power densities with limited cycles. The FDI attack can abruptly change the charging behavior and damage the BES or physical system in the worst-case scenario, as depicted in the figure below.

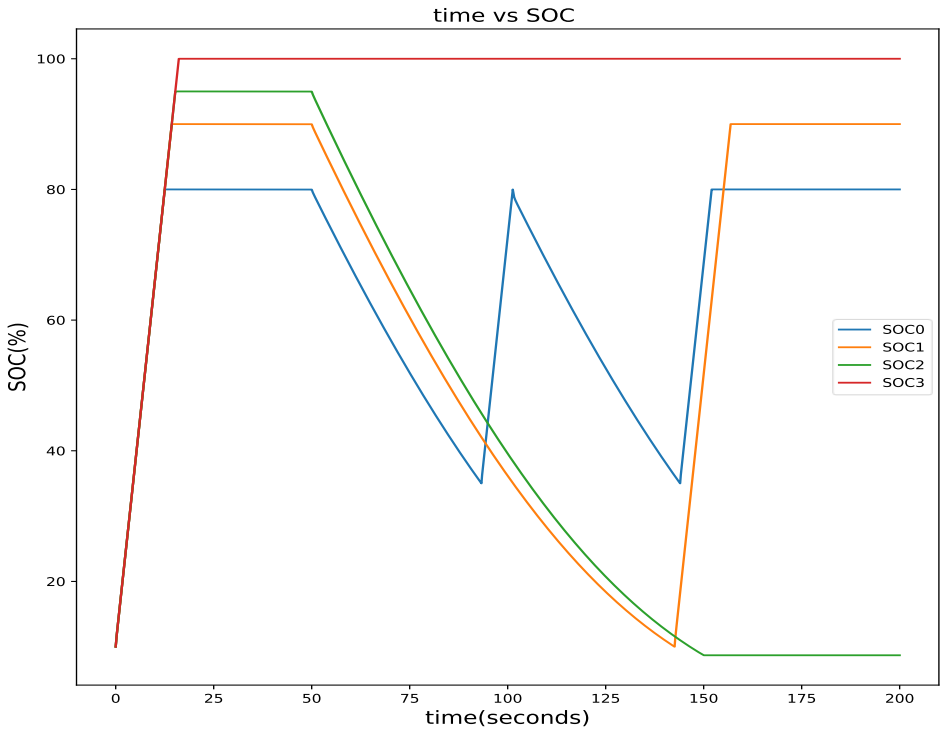


Figure 4.17 Ransomware-driven FDI attack impact on charging behavior

4.5.4.4 Deep learning-based analysis

As shown in Fig. 4.18, all models achieved at least 98 % accuracy, with DNN being very smooth, representing the perfect fitting during the training phase. 1D CNN has a lot of noise after ten epochs. Though LSTM initially shows small fluctuation for a few epochs, it is the most smooth algorithm that achieves the desired accuracy within ten-eleven epochs. Therefore, one can conclude that finely tuned LSTM looks superior for one experiment, though model replicability is the issue of all deep learning techniques. Let's see the results of 10 stratified cross-validations in Table 4.9. The area under the curve (AUC) refers to the degree of separability of the model to distinguish between different classes. LSTM seems the right option for AUC because of its highest AUC and the lowest standard deviation. However, CNN seems superior in terms of accuracy.

The Training of DNN is way faster than at least five times faster than LSTM and sixteen times faster than CNN for 70 epochs, as shown in Table 4.10. It is because of the least units and parameters used in DNN compared to CNN and LSTM. However, As evident from Fig. 4.18, the desired accuracy could be achieved using fewer epochs than 70, reducing LSTM and CNN's training time. Table. 4.11 shows the mean precision, recall, f1-score, and FAR with standard deviation, all in % after 10-fold stratified cross-validation. The CNN model achieves the best f1-score with minimum FAR.

Table 4.9 The area under the curve (AUC) and Accuracy (ACC) of 10-fold stratified cross-validation for RDF

DL methods	**AUC(Mean)**	**AUC (Std)**	**ACC (mean)**	**ACC (std)**
DNN	98.17 %	2.35 %	98.30 %	1.52 %
CNN	98.16 %	1.10 %	98.73 %	1.27 %
LSTM	98.94 %	0.72 %	97.59 %	1.91 %

Table 4.10 Training Time for RDF

DL methods	**Training time**
DNN	1.349 seconds
CNN	16.581 seconds
LSTM	5.98 seconds

Table 4.11 Performance Metrics After 10-Fold Cross-Validation for RDF

DL methods	**Precision Mean | std in %**	**Recall Mean | std in %**	**F1-score Mean| std in %**	**FAR Mean| std in %**
DNN	98.45 | 1.71	97.92 | 1.94	98.17 |1.46	1.88 |2.07
CNN	99 | 1.22	97.33 | 1.77	98.35 |1.14	1.30 |1.59
LSTM	98.70 |1.75	97.66 | 2.17	98.16 |1.47	1.56 |2.10

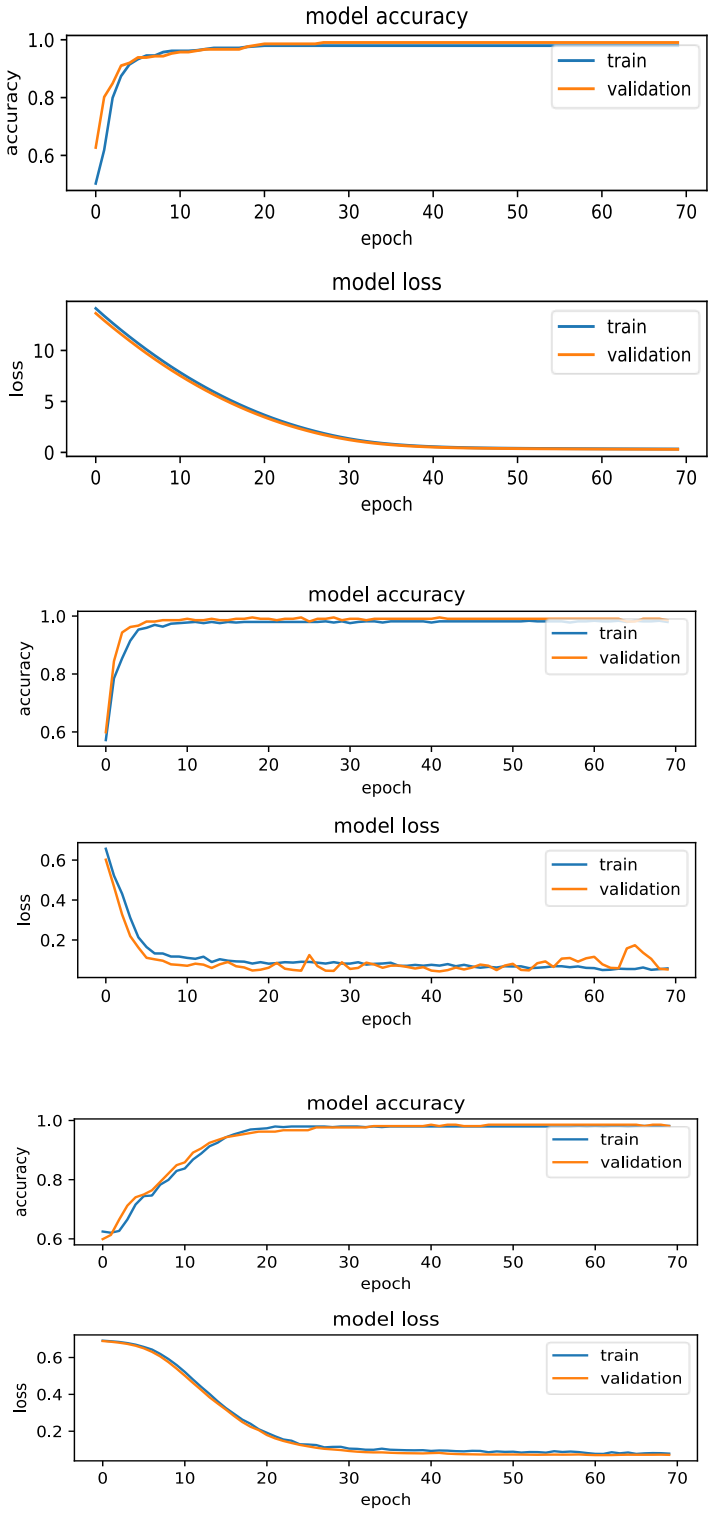


Figure 4.18 Model accuracy vs. loss for a single experiment for DNN, 1D CNN, and LSTM top to bottom.

4.6 GAN-based Network Intrusion Detection System for Electric Vehicle Charging Station

4.6.1 Proposed EC-WCGAN Methodology

The 5G core network uses three-way handshakes of TCP protocol to initiate communication between the SCADA server and EVCS or EV. The client EVCS sends a TCP packet with an SYN flag on through 5G. Upon receiving the SYN packet, the SCADA server acknowledges the connection request and sends back the SYN-ACK signal leaving half open port up to TCP connection timeout. The EVCS acknowledges the SYN-ACK by sending an ACK to the SCADA server, and the communication starts. Before the half-open port timeouts, the malicious EVCS (either by impersonating or via spoofed IP) floods the SCADA server by sending myriads of SYN requests to create many more half-open connections.

As evident from our prior works [40], [41], the DDoS can disrupt the availability of the critical control signal passing from SCADA to the PV controller, BES controller, and EV controller. As a result, low-frequency and high-frequency oscillations are induced in the bus, BES, and EV's generated power, voltage, and current. The proposed detection algorithm can be installed on the SCADA server to inspect the incoming network packets and detect the DDoS attacks on EVCS infrastructure.

The proposed method can be implemented to monitor the traffic flow in the SCADA server of EV infrastructure to detect malicious DDoS attacks. The data are the feature vectors of normal/benign and attack packets fetched to the proposed detection model. The following sections present the operational algorithm and model architectures for GAN and EC-WCGAN with gradient penalty.

4.6.1.1 Generative Adversarial Network (GAN)

Deep Generative models can generate synthetic data by learning the underlying data

distribution of the real data. Among VAE, CAE, and GAN, GAN is the most popular model for synthetic data generation. GAN is the adversarial learning-based generative model that trains both Generator G (ought to capture the data distribution) and Discriminator D (ought to estimate the probability of a sample coming from training $p_r(x)$ rather than the generated data) $p_g(z)$ simultaneously under the two-player minimax game-theoretic setting. G and D are the differentiable functions represented by a parameterized multilayered neural network. The generator model can be considered a counterfeiter trying to generate a fake currency without getting caught. In contrast, the discriminator model can be considered as police detecting fake currency [112]. The minimax objective of a GAN is described in eq. (4.13).

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{r(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_g\,(z)}[\log(1 - D(G(z)))] \qquad (4.13)$$

Researchers have used GAN and its variants to address the data imbalance problem in various domains and aid the generated data for training the classification model to enhance the performance metrics [113]–[116]. On the other hand, there are several architectural modifications of GAN to aid the classifications and generation. A shared discriminator architecture such as the GAN has two final layers: one for discrimination and the other for classification [117]. However, this approach might not be optimal because the network's performance decreases if the discriminator is given two incompatible tasks classification and discrimination.

4.6.1.2 WCGAN with External Classifier (EC-WCGAN)

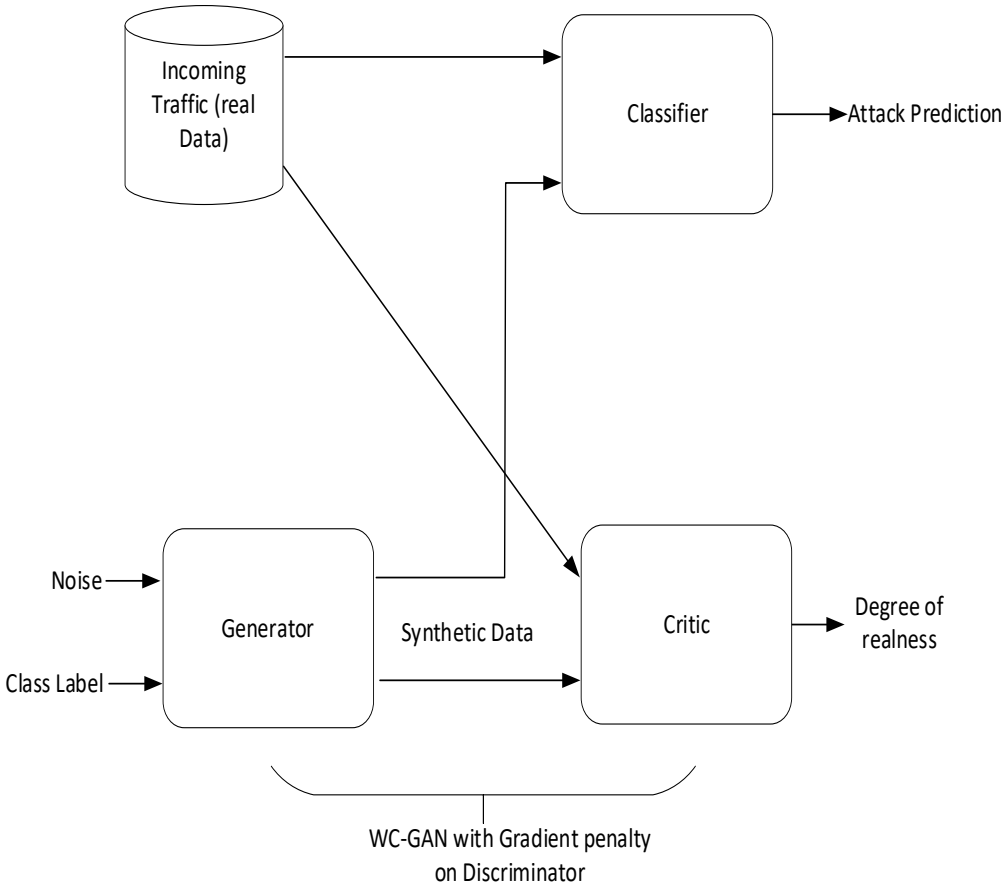


Figure 4.19 EC-WCGAN method for DDoS attack detection.

The proposed EC-WCGAN for EV charging infrastructure consists of three nets: a generator, a discriminator, and a separate classifier as in Fig.4.19. A generator is a WC-GAN capable of generating data of the given random noise vector and class label with stable and improved training. A Discriminator (or Critic) is a neural net that discriminates between real and generated data. Likewise, the external classifier is the neural net trained and supervised to classify the real and generated data into different classes.

Algorithm 1: Training EC-WCGAN with gradient penalty on the discriminator

Input: The gradient penalty coefficient λ, the number of critic iterations per generator iterations n_{critic}, the batch size m and Adam hyperparameters α, β_1, β_2, class label

Initialize →critic parameters w_0, Generator parameters θ_0, Classifier parameters Θ_0

While θ has not converged, do

for *t= 1 ,….,* n_{critic} do

for *i= 1,…., m do*

sample real data $x \sim P_r$*, noise variable* $z \sim P_z$*,*

a random number $\epsilon \sim U[0,1]$

$\tilde{x} \leftarrow G_\theta(z)$

$\hat{x} \leftarrow \epsilon x + (1-\epsilon)\,\tilde{x}$

$L^i \leftarrow D_w(\tilde{x}) - D_w(x) + \lambda(\,|\,|\nabla_{\hat{x}}\, D_w(\hat{x})\,|\,|_2 - 1)^2$

end for

$w \leftarrow Adam(\nabla_w \frac{1}{m}\sum_{i=1}^{m} L^i, w, \alpha, \beta_1, \beta_2)$

end for

sample a batch of noise variables $\{z^i\}_i^m \sim p(z)$

$\theta \leftarrow Adam(\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} -D_w\big(G_\theta(z)\big), \theta, \alpha, \beta_1, \beta_2)$

sample batch of real data $x \sim P_r$*, generated data*

$z \sim G_\theta(z)$*, class label y*

$\tilde{x} \leftarrow G_\theta(z)$

$L^i \leftarrow CE(C_\Theta(x), y) + \lambda\, CE\big(C_\Theta(\tilde{x}), argmax\,(C_\Theta(\tilde{x}) > t)\big)$

$\Theta \leftarrow Adam(\nabla_\Theta \frac{1}{m}\sum_{i=1}^{m} L^i, \Theta)$

end while

The architectures of all three neural nets are independent. The generator, critic, and classifier have three hidden layers with ascending numbers of hidden units (256, 512, 1024) with 30 % dropout in a generator, with descending numbers of hidden units (256, 512, 1024) in critic and with (128,256,128) hidden units with 30% dropout in the classifier. The neural architecture remains the same in the binary and multiclass classification of the DDoS attacks. The generator and critic use "$LeakyReLU$" activation function with a negative slope coefficient (α) equal to 0.2 for hidden layers, while the classifier uses the normal "$ReLU$" activation function. The generator

and critic use "$tanh$" activation for its terminal layers, while the classifier uses "$softmax$" activation. The $adam$ optimizer has been used to optimize the parameters of the model. We have trained the model with a batch size of 128, many critics parameter updates per generator update (n_critic=5) with noise dimension=30, confidence threshold=0.2, and adversarial weight =0.1. The main goals of our algorithms are first to classify whether the incoming vector is an attack or a benign (Normal) data vector and second to classify the incoming data vector into different attack classes (4 DDoS attack classes and one benign class). The former is best known as binary classification, and the latter is termed multiclass classification. Furthermore, the third is to present a comparative analysis of applied algorithms.

4.6.1.3 Data set

There are no disclosed instances of DDoS attacks on the real EV infrastructure so far. Therefore, most cybersecurity researchers test their algorithms on datasets that closely fit the attack scenarios and the infrastructure setup. Also, the network architecture of the CPS system is not much different than the IT infrastructures. These datasets are created by emulating cyberattacks on the network testbed and represent the most up-to-date attack data. The CICIDS 2018 DDoS attack dataset is used for this research since it includes the recent DDoS attacks and closely matches the infrastructure setup of EV charging infrastructures.

Before fetching the data into the EC-WCGAN, all the feature vectors are standardized using the Standard Scaler, which transforms the feature vectors into another feature space with zero mean and unit variance. The principal component analysis (PCA) has been implemented to convert 77 hyperdimensional feature space to 30-dimensional lower hyperdimensional space. The L_2 normalizer has been implemented to scale individual samples to have a unit norm. After that, the attack classes have been one-hot encoded for appropriate representation. Finally, 70% of the data

has been used in training, and 30% has been used in testing.

4.6.2 Results and Discussion

The 5G-induced DDoS has been simulated in NetSim and MATLAB Simulink. In this work, all the coding for the detection and classification are run in Python 3.7.13 in the Google CollabPro with 12.68 GB RAM and 225.89 GB disk. The article serves two-fold goals, the first one predicting the DDoS attack and the second categorizing the incoming traffic into different classes of DDoS attack and normal packets for the EV charging infrastructures.

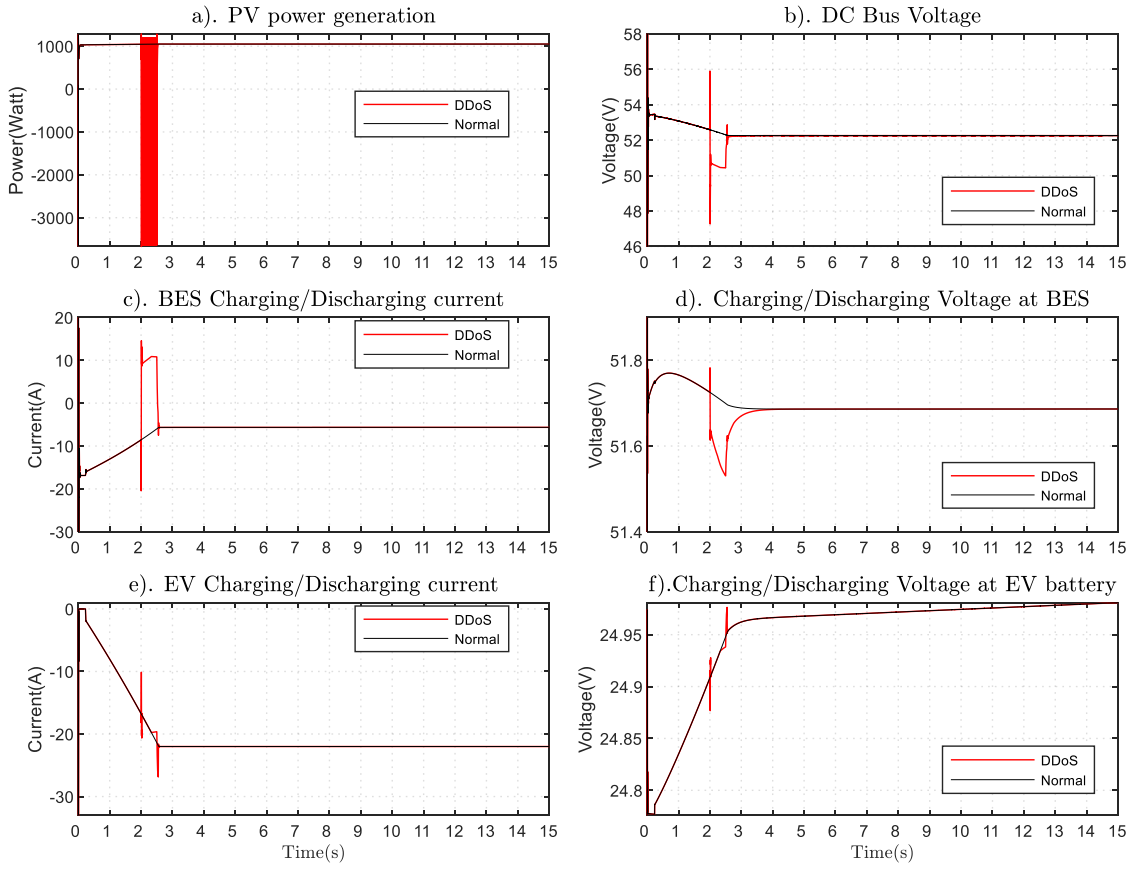


Figure 4.20 Impacts of DDoS attacks launched at PV controller from 2-2.5 seconds, BES controller from 6-6.5 seconds, and EV controller at 10 -10.5 seconds.

Fig.4.20 represents the impact of the DDoS attack on the cyber-physical system of EVCS caused by the 500 ms delay in the 5G communication system. The attacker launches the attack on

the PV controller at t=2 seconds, lasting 500 ms. The observed impacts are the high-frequency oscillations in the power signal as in Fig. 4.20a, momentary voltage swing at DC bus voltage as in Fig. 4.20b, BES current surge as in Fig. 4.20c, and BES voltage drop as in Fig. 4.20d. Similarly, current and voltage spikes in the EV charging have been observed in Fig. 4.20e and Fig. 4.20f, respectively. However, due to PI controllers fixed upper and lower saturation thresholds, the DDoS attack at BES and EV controller has no impact. In contrast, the improperly tuned PI controllers with no saturation thresholds are found to be exploited.

For the DDoS detection problem as well as the multiclass classification problem of DDoS, the EC-WCGAN is trained with the same architecture as described in section 4.5.1. However, the proposed architecture is trained for 200 epochs for the DDoS detection problem and 50 epochs for multiclass classification.

4.6.2.1 Plot-based Responses

As in Fig. 4.21, at the start of training, the generator and the critic suffer from a high loss as they have to optimize high Wasserstein loss due to the random noise vector passed to the generator. And as the training progresses, the loss decreases and is stabilized. The same holds for multiclass classification as well as in Fig. 4.22. Classifiers in binary and multiclass classification are trained with the actual data as long as the generator produces a plausible synthetic sample. That's why classifier error does not change much.

As per Table 4.12 and Table 4.13, The classifier trained with WCGAN has more than 99% performance metrics such as precision, recall, and F1-score for DDoS detection and classification problems.

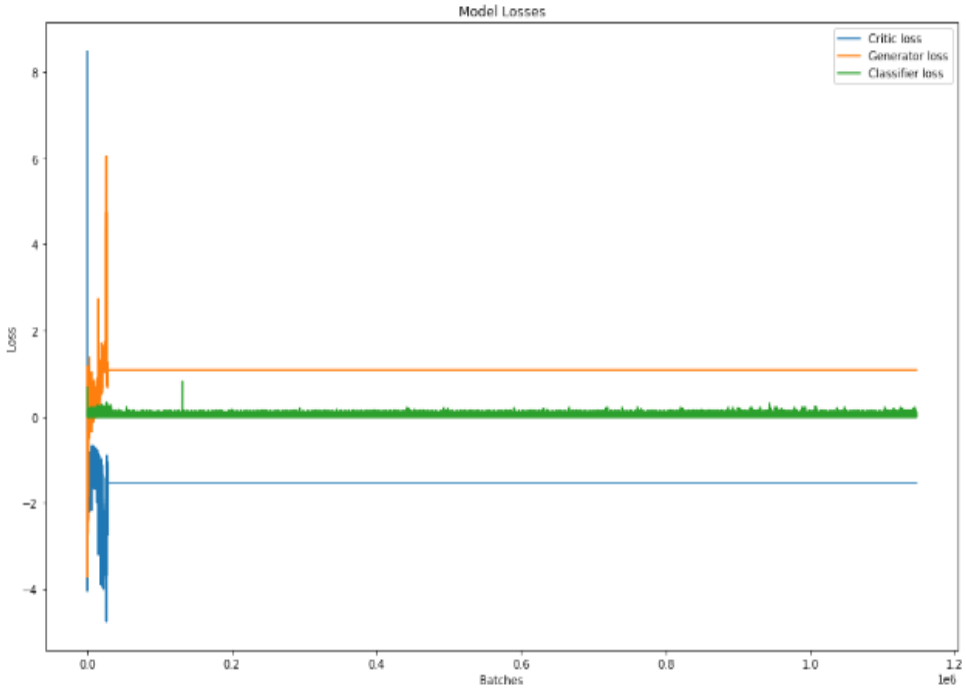


Figure 4.21 Generator, Critic, and Classifier loss during the training of binary classification (DDoS attack or Normal operation)

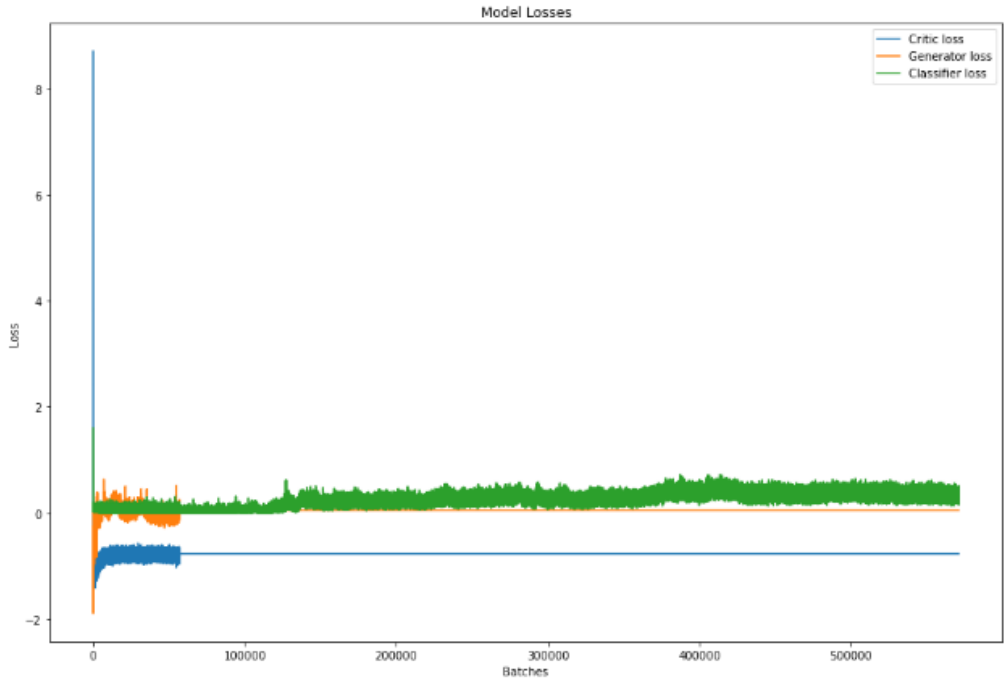


Figure 4.22 Generator, Critic, and Classifier loss during the training of multiclass classification.

Table 4.12 Classification metrics for DDoS Detection using EC-WCGAN

Attack Class	Precision	Recall	F1-score	support
Benign	0.999935	0.998677	0.999306	430912
DDoS Attack	0.997097	0.999857	0.998475	195825

Table 4.13 Classification metrics for multiclass classification using EC-WCGAN

Attack Class	Precision	Recall	F1-score	support
Benign	0.999916	0.998689	0.999302	430912
DoS attacks-GoldenEye	0.995257	0.999127	0.997188	12600
DoS attacks-Hulk	0.996080	0.999645	0.997859	138018
DoS attacks-SlowHTTPTest	0.999881	1.00000	0.999940	41927
DoS attacks-Slowloris	0.994787	0.989024	0.991897	3280

4.6.2.2 Performance comparison with DNN and LSTM Algorithm

Unlike our previous work at [33], the proposed method surpassed the deep learning-based IDS performance metrics in detection and classification tasks. We observed that the low sample class did not perform better, but the proposed model seems to be a good fit as it enhances the performance of the low sample class. Also, the proposed model has similar or superior performance compared to the LSTM-based IDS in the same paper. Moreover, The proposed model only uses 30 best features as opposed to all 76 features used by our previous work.

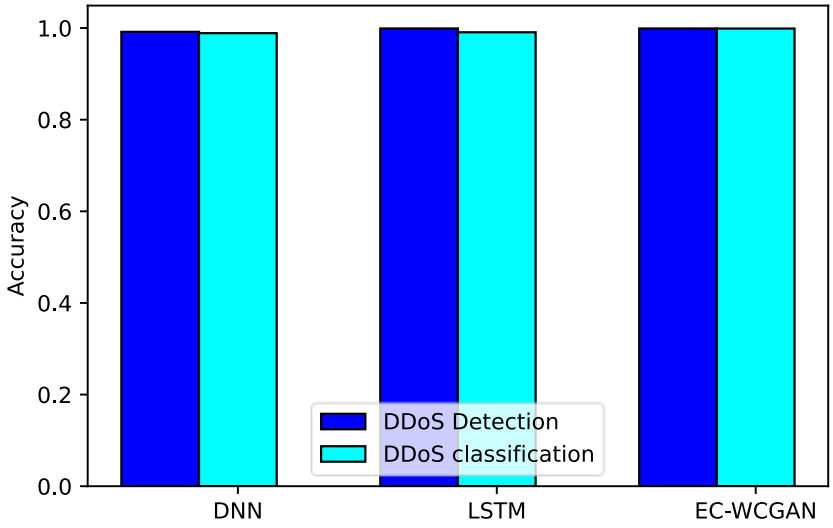


Figure 4.23 Comparative performance of EC-WCGAN with existing DL models

The EC-WCGAN has a 0.73 % increment in detection accuracy and a 5.5 % decrement in loss with respect to DNN. Also, it exhibits comparable performance metrics with LSTM, trailing the detection accuracy by 0.0035% and loss by 0.019%. The blue bars in Fig. 4.23 presents the binary DDoS detection accuracy of various DL and proposed method. Similarly, EC-WCGAN has the highest DDOS classification accuracy, with a 1.01 % increment in classification accuracy with respect to DNN and a 0.81% increment with respect to LSTM. Also, the proposed method has a 10.2% reduction in average loss as compared to the DNN and a 0.098% decrement with respect to LSTM.

4.7 Chapter Conclusion

This section presents different cyberattack detection algorithms for the EVCS in the network and the host. We integrated the DL-based classifier with WCGAN based model and trained the classifier with the generator to solve the DDoS detection and classification task in EV infrastructure. The proposed EC-WCGAN-based IDS outperformed another DNN-based IDS in

terms of accuracy, precision, recall, and F1-score for the binary as well as multiclass classification and achieved the desired accuracy of around 99.9%. Also, the proposed model has comparable performance with LSTM and improved DDoS classification accuracy. The main advantage of the proposed method is it is well-suited for low-sample data classification. Unlike DL-based models such as DNN and LSTM, EC-WCGAN performs better with just 30-dimensional features as opposed to 76-dimensional features in DL methods. This application could safeguard the EV infrastructure and its stakeholders from possible cyber threats. Adding new kinds of attack data in training, the proposed model could easily scale up to detect more diverse attacks. It ensures the scalability and interoperability of our model. The next chapter will focus on designing cyber-defense and mitigation strategies for efficient attack recovery planning.

Chapter 5 Mitigation of Adverse Effects of Cyber-Attacks on Electric Vehicle Charging Station

5.1 Introduction

Cyber recovery planning for EVCS is a less studied and less explored area of cyber-physical systems, as transportation electrification is in the nascent deployment phase. The sophisticated computational intelligence detection is the primary step for cyber-physical security of EVCS, but challenges remain to address the defense and mitigations of attack impacts. Cyber defense and mitigations of cyberattack impacts during and after the attacks are the two important pillars of recovery planning. Therefore, the mitigation of adverse effects of cyberattacks on EVCS should be addressed appropriately. All the past works do not contribute enough to the attack recovery, defense, and corrections in the EVCS. The major contributions of this chapter are as follows: It performs the impact analysis of the APT attacks engineered with domain expertise on standalone PV-powered EVCS. Also, it develops and analyzes the cyber-enabled physical attack strategy to impact the entire charging process stealthily. A novel data-driven controller clone with a TD3-based algorithm that could correct or take over the legacy controllers under APT detection has been proposed. Each agent is independent and can correct attacks on the corresponding legacy controller of EVCS. The performance of the proposed TD3 based clones has been compared with that of the benchmark DDPG clone. We envisioned the concept of embedded and distributed intelligence for critical legacy controllers. This chapter also introduces the more visible and less complex Bruteforce attack mitigation strategy with a human or automated agent in the loop for attack recovery planning. In addition, it introduces the concept of Controller clone for CPS attack recovery planning.

5.2 System Model and Mathematical Formulation

We have designed the PV-powered off-the-grid standalone EVCS prototype comprised of PV, BES, and EV with an associated control strategy in Chapter 3. It has the SCADA system communicating with three isolated field controllers: PV, BES, and EV. The EVCS architecture, control circuitry, system formulation, and component modeling are available in Chapter 3 as well. These field controllers are responsible for the reliable and safe operation of EVCS and hold exploitable technical vulnerabilities. Using social engineering and/or reverse engineering, the adversary can poison the control signals reaching the physical controllers at EVCS either at the network level of the SCADA system or the physical infrastructure layer. On that note, the threat actors with domain expertise can launch vicious APT attacks on these legacy controllers. To deal with these APT, Reinforcement Learning can be the reasonable control paradigm. Fig. 5.1. depicts the working mechanisms of an individual DRL-based controller agent in EVCS and is valid for all controller agents: PV, BES, and EV agents. The detailed functionality and deployment of these

agents with states, rewards, and action information will be discussed in section 5.4.3.

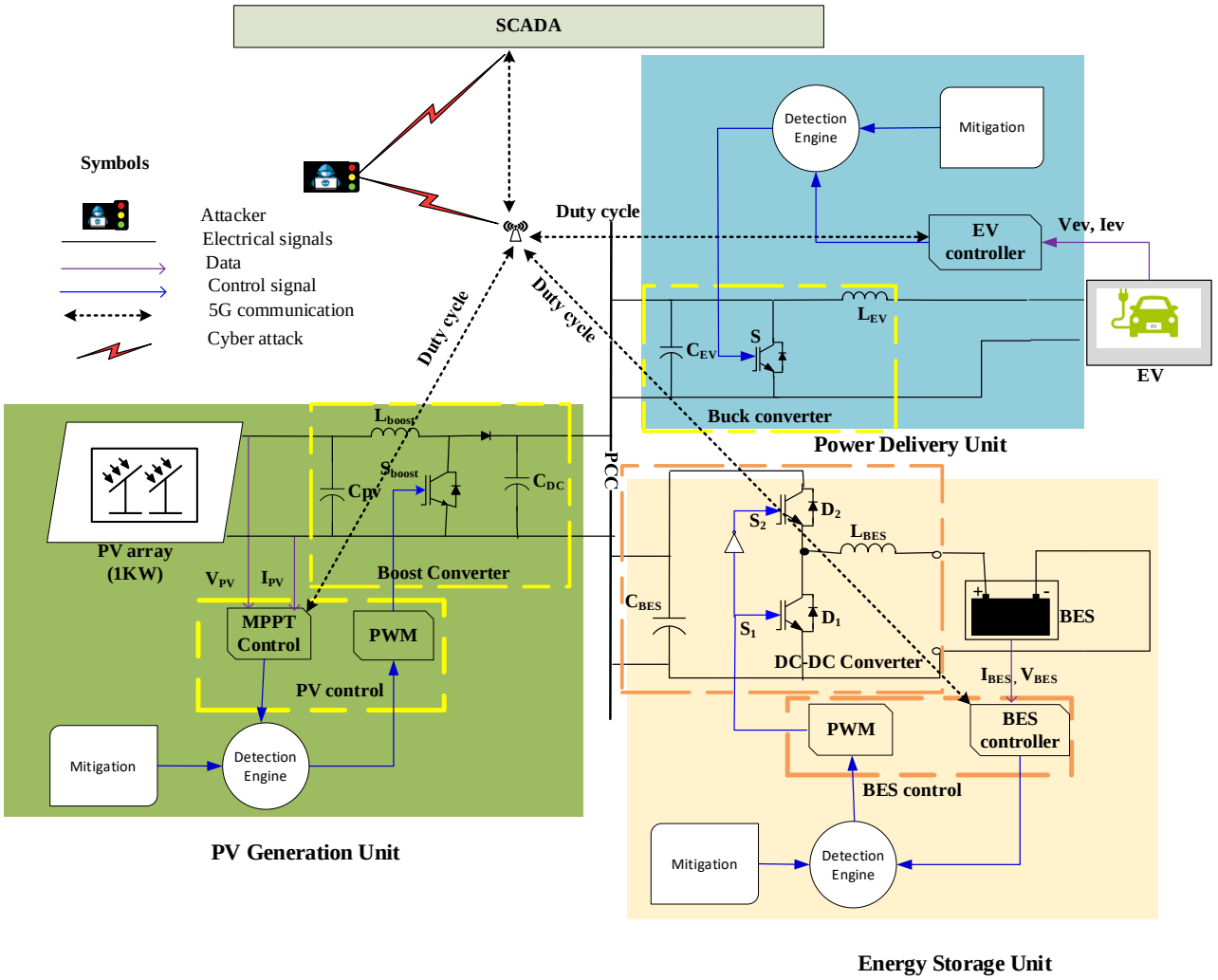


Figure 5.1 Proposed Detection and Defense based on DRL.

5.2.1 Reinforcement Learning

Reinforcement learning is a goal-directed, direct, adaptive control that maps observation into the actions to maximize the expected scalar reward founded on the notion of trial-and-error search and delayed reward [118]. As per Fig. 5.2., At each discrete time step t, with the given observation states $s_t \in$ S, the agent selects actions $a_t \in A$ with respect to policy π: $S \rightarrow A$ receiving reward r_t and new state of the environment s_{t+1}. The discounted sum of rewards R is given in (5.1) where the discount factor $\gamma \in [0,1]$ represents the priority of short term rewards.

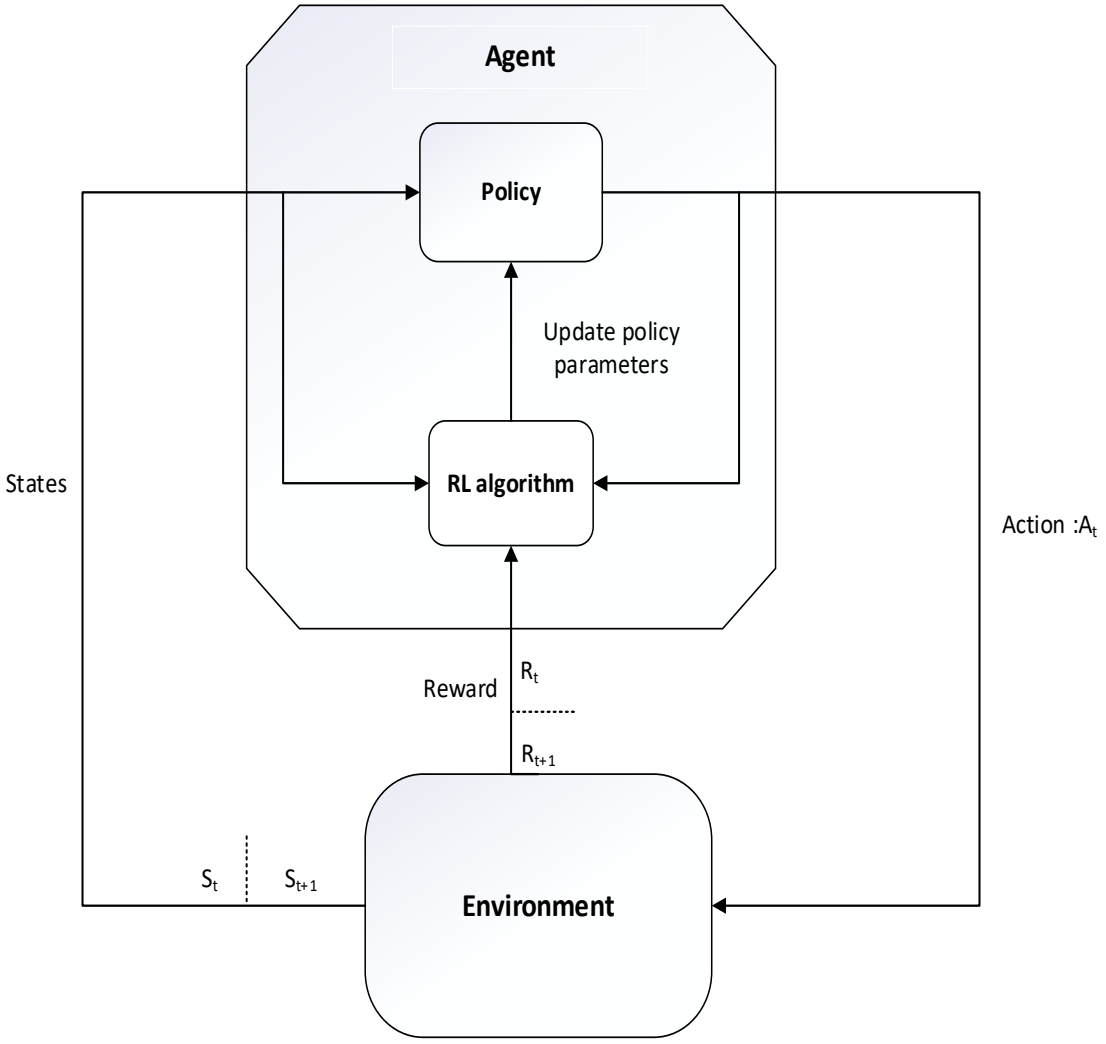


Figure 5.2 The agent environment interaction of RL in a Markov Decision Process.

$$R \triangleq \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i) \tag{5. 1}$$

Theorem 1: Markov Property

Given the present, it states that the future is independent of the past. A state $s_t \in S$ is Markov if and only if as in (5.2)

$$P[s_{t+1}|s_t] \triangleq P[s_{t+1}|s_1, \ldots\ldots, s_t] \tag{5. 2}$$

Markov Decision Process (MDP) formally characterizes an environment in an RL where the

current state describes the process completely. An MDP is a tuple of $M = \langle S, A, P, R, \gamma \rangle$ Where S is a finite set of states, A is a finite set of actions, and P is a state transition probability referring to the likelihood of going to the next state s' from current state s after taking action a defined in (5.3).

$$P_{ss'} = P[s_{t+1} = s' | s_t = s, a_t = a] \tag{5. 3}$$

The goal of reinforcement learning is to find the optimal policy π_ϕ with parameter ϕ that maximizes the expected return as in (5.4). The parameterized policy π_ϕ for continuous control can be updated by taking the gradient of the expected reward $\nabla_\phi J(\phi)$ as in (5.5). The actor or policy can be updated using a deterministic policy gradient algorithm in Actor-Critic methods. The critic or the value function , $Q^\pi(s, a)$, is defined as the expected return while taking action a in state s and following the policy π after that as in (5.6). One of the ways to learn the value function is to use a temporal difference learning in the Bellman equation that relates the value of the current state-action pair to the value of the next state-action pair as (5.7). As there are too many states and actions to store in memory in the large MDP, the value function can be approximated by using a differentiable function approximator $Q_\theta(s, a)$. It is parameterized by θ as in (5.8). The parameter θ can be updated by using Monti Carlo or Temporal difference learning. In this learning of DQN, θ is updated by using a secondary frozen target network $Q_{\theta'}(s, a)$ to maintain the fixed objective y over multiple updates, as in (5.9). The actions a' are selected from the target actor-network $\pi_{\phi'}(s')$. The parameter θ' of the target network can be updated by exactly matching the weight of the current network or by some proportion τ at each timestep as in (5.10).

$$J(\phi) = \mathbb{E}_{s_i \sim p_\pi, a_i \sim \pi}[R_0] \tag{5. 4}$$

$$\nabla_\phi J(\phi) = \mathbb{E}_{s \sim p_\pi}[\nabla_a Q^\pi(s, a)|_{a=\pi(s)} \nabla_\phi \pi_\phi(s)] \tag{5. 5}$$

$$Q^{\pi}(s,a) = \mathbb{E}_{s_i \sim p_{\pi}, a_i \sim \pi}[R_t|s,a] \tag{5. 6}$$

$$Q^{\pi}(s,a) = r + \gamma \mathbb{E}_{s',a'}[Q^{\pi}(s',a')], \qquad a' \sim \pi(s') \tag{5. 7}$$

$$Q_{\theta}(s,a) \approx Q^{\pi}(s,a) \tag{5. 8}$$

$$y = r + \gamma Q_{\theta'}(s',a'), \qquad a' \sim \pi_{\phi'}(s') \tag{5. 9}$$

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{5. 10}$$

5.3 Attack Modeling

The attacker's primary goal is to disrupt, damage, or freeze the critical controllers of EVCS. The attacker is assumed to poison/manipulate the critical parameters with sophisticated attacking tools and domain expertise. Most legacy controllers generate a critical control signal, i.e., the duty cycle that controls the switching of the high-frequency transistor switches. It is assumed that the attacker can control the number of controllers ($N_c \in \mathbb{R}$), the attack duration ($T_a \in \mathbb{R}$) and Types of the attack $S_a = \{(A_t, E_a)\}$ once it exploited the critical control signals $\mathcal{C} = \{C_1, C_2, .., C_n\}$ from controllers $N_1, N_2, .., N_n$. The attacker chooses the set of exploited resources ζ from another set $\mathcal{M} = \{N_c, T_a, S_a, \mathcal{C}\}$ in such a way as to minimize the critical functionality CF of the process as in (5.11). The attack Type S_a can be a tuple of attack time $A_t = \{sim, diff\}$ and engineered attack Types $E_a = \{\tau_1, \tau_2\}$ where *sim* and *diff* refer to the attack that can be launched simultaneously and at different times, respectively, with attack Types $\tau_1 \; and \; \tau_2$ defined in (5.12) and (5.13).

$$\underset{\zeta \subset \mathcal{M}}{\operatorname{argmin}}\ CF \tag{5. 11}$$

$$\tau_1 = \mathrm{T}(\alpha) \tag{5. 12}$$

$$\tau_2 = c \tag{5. 13}$$

Where T is some random function parameterized by parameters α and c is some scalar constant. The function T is envisioned to generate the statistical randomness in the attack. With critical control signals of the controllers $\mathcal{C} \in [low\_limit, upper\_limit]$, it is wise for a stealthy attacker to design a similar kind of pseudorandom attack that intersects with the range of $\mathcal{C}$. Pseudorandom number *PRN (low_limit, up_limit, rep)* fluctuates between the lower and upper bound, and repeating *rep* times serve the purpose. Similarly, c is the average of the upper and lower limit of the $\mathcal{C}$. After finding the sets of optimal ζ, Finally, the attacker algebraically combines the attack signal $E_a$with the critical parameter set $\mathcal{C}$ as per (5.14).

$$attack_{signal} = \mathcal{C} \pm E_a \text{ subjected to } \zeta \tag{5. 14}$$

Table 5.1 Parameters for attack modeling

Parameters	**Value**
N_c	3
T_a	2 seconds
S_a	$\{sim, diff\} \times \{\tau_1, \tau_2\}$
$\mathcal{C}$	$\{\mathfrak{D}_{PV}, \mathfrak{D}_{BES}, \mathfrak{D}_{EV}\}$

CF	Observed normalcy or stability of the process variables such as power, bus voltage, BES, and EV voltages and currents
τ_1	PRN(0, 1,10)
τ_2	0.5
$Type\ I\ attack = \mathcal{C} \pm E_a$ s.t. $\zeta(.\tau_1)$	$\{\mathfrak{D}_{PV} + \tau_1, \mathfrak{D}_{BES} - \tau_1, \mathfrak{D}_{EV} - \tau_1\}$
$Type\ II\ attack = \mathcal{C} \pm E_a$ s.t. $\zeta(.\tau_2)$	$\{\mathfrak{D}_{PV} - \tau_2, \mathfrak{D}_{BES} - \tau_2, \mathfrak{D}_{EV} - \tau_2\}$

We pragmatically chose the $\zeta\ and\ CF$ of the attack for this case, as in table 5.1, after repeated experimentation. Type I and Type II attacks are carefully engineered APT attacks with domain expertise. The Type I attack imposes the low-frequency attack on the duty cycles, while the Type II attack imposes the constant duty cycle attack.

5.4. Proposed Mitigation Techniques

5.4.1. Controller clone-based mitigation employing the DRL TD3 algorithm

The data-driven digital clones for the physical controllers employing DRL-based TD3 algorithms are trained and deployed in each critical controller that controls a dynamic system's critical functionality. Moreover, the clones employing TD3 agents are compared with the benchmark DDPG algorithm. Upon the threat incidence or operational anomaly, the rule-based or DL-based detection engine deploys the corrected control signal from the clones. It takes over the legacy controllers until the threat has been eliminated. The RL-based autonomous defense agent is employed for each controller whose primary purpose is to generate the corrected control signal upon incidences of cyberattacks and system anomalies. These controllers are designed for the mere to extreme adversarial setups such as APT or malware that could freeze/control the legacy

controllers. The detailed functionality and deployment of these agents with states, rewards, and action information will be discussed in sections 5.4.1.3-5.4.1.5.

5.4.1.1 Twin Delayed Deep Deterministic Policy Gradient (TD3)

Actor-critic RL learns value function (as in value-based RL) and policy (as in policy-based RL) and is proven with better convergence properties, the ability to learn stochastic policy, and efficacy in hyperdimensional or continuous action space. The function approximation error in actor-critic RL leads to overestimated value estimates and suboptimal policies [119]. TD3 is the off-policy actor-critic RL designed for continuous action space to minimize the impact of overestimation bias on both actor-critic networks by implementing three tasks. The first is Clipped Double -Q Learning, where TD3 uses a minimum of two Q-values to form the target. The second one is the delayed policy updates of the target network. And the third one is the Target policy smoothing, where TD3 adds noise to the target action so that the target policy cannot exploit Q function error by smoothing out Q along the gradient of action.

Algorithm 5.1: TD3 Algorithm

Each proposed standalone control agent for EVCS follows the strict training protocol as follows.

Initialize critic networks $Q = [\, Q_{\theta_1},\ Q_{\theta_2}]$ and actor-network π_ϕ with random parameters $\theta_1, \theta_2\ and\ \phi$

Initialize target networks $\theta_1' \leftarrow \theta_1,\ \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$

Initialize replay buffer $\mathcal{B}$

for t=1 to T do

Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma)$

Store transition tuple $\langle s, a, r, s', d\rangle$ into $\mathcal{B}$ where d is the signal to indicate s' is the terminal state

If s' is the terminal state, reset environment state

Else randomly sample mini-batch of N transitions

$\langle s, a, r, s', d\rangle$ from $\mathcal{B}$

Compute the target actions and compute targets:

$$a'(s') = clip\big(\, \mu_{\phi'}(s') + clip(\epsilon, -c, c), a_{low}, a_{high}\big), \epsilon \sim \mathcal{N}(0,1)$$

$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\theta'_i}(s', a'(s'))$

Update critics Q-function by using one step of gradient descent:

$\theta_i \leftarrow \text{argmin}_{\theta_i} \nabla_{\theta_i} \frac{1}{|\mathcal{B}|} \sum_{\langle s,a,r,s',d \rangle \in \mathcal{B}} (Q_{\theta_i}(s, a) - y(r, s', d))^2$

If t mod policy_delay, then

Update ϕ by the deterministic policy gradient:

$$\nabla_\phi J(\phi) = \frac{1}{|\mathcal{B}|} \sum \nabla_a Q_{\theta_i}(s, a)|_{a=\mu_\phi(s)} \nabla_\phi \pi_\phi(s)$$

Update target networks:

$\theta'_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$

$\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$

End if

End for

5.4.1.2 Graphical representation of TD3 algorithm

TD3 uses twin critic networks (critic 1 and critic 2) inspired by DRL with clipped Double Q-Learning, where it takes the smallest Q-value of two critic networks to remove the overestimation bias in $Q_{\theta_i}(s, a)$. The concept of target networks is introduced to stabilize the agent training. The target network provides a stable objective and greater coverage of the training data, as DNN requires multiple gradient updates to converge [119]. Without the fixed target, the accumulated residual errors after each update produce divergent values when paired with a policy maximizing the value estimate. Therefore, TD3 uses delayed updates of the actor-network (policy update) compared to the critic network (value update), resulting in more stable training.

The replay buffer stores the history of agent experience and randomly fetches the data in mini-batches to update actor and critic networks. There are six neural networks in TD3: two critic networks, two target networks for two critics, an actor network, and a corresponding target network for the actor. Fig. 5.3 summarizes the graphical abstract of a TD3 agent.

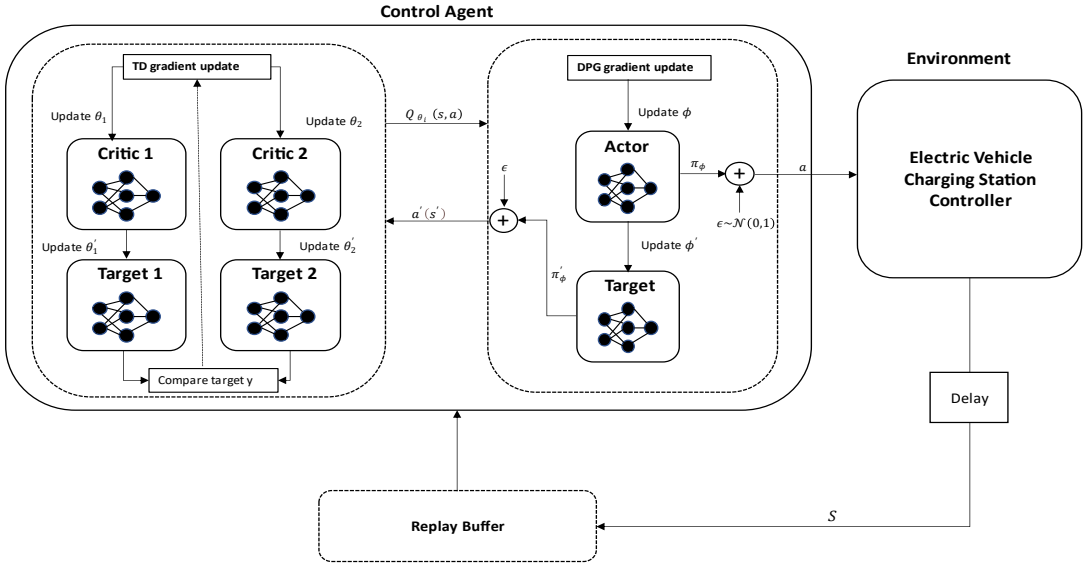


Figure 5.3 Graphical representation of a TD3 agent.

5.4.1.3 PV agent

The design goal of the PV agent is to take over the infected MPPT controller and implement the optimized control policy to generate the duty cycle $\mathfrak{D}_{PV}(t)$ needed by the boost converter to have the least impact on the system. A PV agent continuously monitors the error $e_P(t)$ and integrated errors $e_{int\_P}(t)$ between the PV output power $P_{PV}(t)$ and reference power $P_{ref}(t)$ as in (5.15) and (5.16). The objective of the PV agent is to find the optimal policy for the duty cycle that correctly transforms observation space into action space by maximizing the cumulative scalar reward.

The output or action of the PV agent is the duty cycle with a linear quadratic regulator (LQR) as the instantaneous reward or cost function $r_{PV}(t)$ as in (5.17). The $\alpha = 0.01$ and $\beta = 1$ on $r(t)$ represent the negative penalty terms imposed on error and action, respectively. T_S is the sampling time and is the same for each agent.

$$e_P(t) = P_{ref}(t) - P_{PV}(t) \tag{5. 15}$$

$$e_{int\_P}(t) = \sum_{T_s} e_P(t) \tag{5. 16}$$

$$r_{PV}(t) = \alpha e_P(t)^2 + \beta \mathfrak{D}_{PV}(t)^2 \tag{5. 17}$$

The rule/threshold-based detection engine derived pragmatically for the PV agent will determine the attack event if observed power falls beyond the range (1020,1045) AND the duty of MPPT falls beyond the range (0.200,0.201).

5.4.1.4 BES agent

The design goal of the BES agent is to generate the corrected duty cycle $\mathfrak{D}_{BES}(t)$ for the buck-boost converter under the threat incidence. Similar to the PV agent, the BES agent observes the states of the error $e_V(t)$ and integrated errors $e_{int\_V}(t)$ between the desired reference bus voltage $V_{bus\_ref}(t)$ and the bus voltage $V_{bus}(t)$ as in (5.18) and (5.19). The optimal control policy that maps the observation space to the action space is found by minimizing the expected value of the cost function $r_{BES}(t)$, which is the linear quadratic regulator function. The $\alpha = 0.01$ and $\beta = 1$ on $r_{BES}(t)$ represent the negative penalty terms imposed on error and action, respectively, as in (5.20).

$$e_V(t) = V_{bus\_ref}(t) - V_{bus}(t) \tag{5. 18}$$

$$e_{int\_V} = \sum_{T_s} e_V(t) \tag{5. 19}$$

$$r_{BES}(t) = \alpha e_V(t)^2 + \beta \mathfrak{D}_{BES}(t)^2 \tag{5. 20}$$

The rule/threshold-based detection engine derived pragmatically for the BES agent will determine the attack event if observed power falls beyond the range (1020,1045) and the PI controller's duty falls beyond the range (0.7,0.71).

5.4.1.5 EV agent

The design goal of the EV agent is to generate the corrected duty cycle $\mathfrak{D}_{EV}(t)$ for a buck converter if the legacy EV charger got infected. Similar to the previous agent, the EV agent observes the states of the error $e_{VEV}(t)$ and integrated errors $e_{int\_VEV}(t)$ between the desired reference battery voltage $V_{batt\_ref}(t)$ and the bus voltage $V_{batt}(t)$ as in (5.21) and (5.22). The optimal control policy that maps the observation space to the action space is found by minimizing the expected value of the cost function $r_{EV}(t)$, which is the linear quadratic regulator function. The $\alpha = 0.01$ and $\beta = 1$ on $r_{EV}(t)$ represent the negative penalty terms imposed on error and action, respectively as in (5.23).

$$e_{VEV}(t) = V_{batt\_ref}(t) - V_{batt}(t) \tag{5. 21}$$

$$e_{int\_VEV} = \sum_{T_s} e_{VEV}(t) \tag{5. 22}$$

$$r_{EV}(t) = \alpha e_{VEV}(t)^2 + \beta \mathfrak{D}_{EV}(t)^2 \tag{5. 23}$$

The rule/threshold-based detection engine derived pragmatically for the EV agent will determine the attack event if observed power falls beyond the range (1020,1045) and the PI controller's duty falls beyond the operating range (0.54,0.55). Table 5.2 summarizes the observations, reward, and action information of multiple TD3 agents.

Table 5.2 Summary of multiple independent agents

Agents	Observations ($\mathcal{S}$)	Reward ($\mathcal{R}$)	Action($\mathcal{A}$)
PV Agent	$\{e_P, e_{int\_P}\}$	$\{r_{PV}\}$	$\{\mathfrak{D}_{PV}\}$
BES Agent	$\{e_V, e_{int\_V}\}$	$\{r_{BES}\}$	$\{\mathfrak{D}_{BES}\}$
EV Agent	$\{e_{VEV}, e_{int\_VEV}\}$	$\{r_{EV}\}$	$\{\mathfrak{D}_{EV}\}$

5.5 Experimental setups for the TD3-Based method

The TD3-based agents are built with specific neural architectures for critics and actor neural networks with similar architecture for the target neural network. Then, the layerwise actors' and critics' neural networks with their targets are properly engineered and parameterized with desired activation functions and appropriate initial weights and biases. Finally, the hyperparameters are carefully selected to train the agents optimally after the series of training up to 500 episodes. Similarly, all the hyperparameters and neural architectures are kept the same in the DDPG agents for accurate comparison.

5.5.1 Configurations of TD3 Critic networks

A TD3 critic estimates the optimal Q-value based on the observations and actions received by the parameterized DNN. Fig. 5.4 depicts the structure of a single critic network we have created. Before concatenating those features, the state and action information goes through some local neural network transformations. After concatenation, it goes through another set of neural networks to produce the Q-value function. The network that takes state info has three fully connected hidden layers with respective hidden units of 64, 32, 16, and the ReLU activation layer between them. Also, the action info is passed through the fully connected neural network with 64 hidden units. After the concatenation, the transformed state and action info pass-through two fully

connected hidden layers with 64 and 32 hidden units, respectively, with the ReLU layer in between to produce the Q-value. We then create the critic representation using specified neural networks and options.

5.5.2 Configurations of TD3 Actor networks

The actor-networks in the TD3 agent decide which action to take based on the observations, i.e., policy mapping. We have created a DNN with three fully connected hidden layers with respective hidden units of 64, 32, and units equal to the number of actions, i.e., 1 in our case, with ReLU layers in between. In addition, a sigmoid layer is added since the output of the action ranges from 0 to 1 for the duty cycle in our case. Finally, the scaling layer scales the output from the sigmoid layer with a scale of 1 and a bias of 0.5. The scale is set to the range of the action signal, and the bias is set to half a range. We then create the actor representation using specified neural networks and options as in Fig. 5.5. Table 5.3 presents the options of actor-network, critic network as well as training of agent. Table 5.4 presents the hyperparameters setting to administer the training.

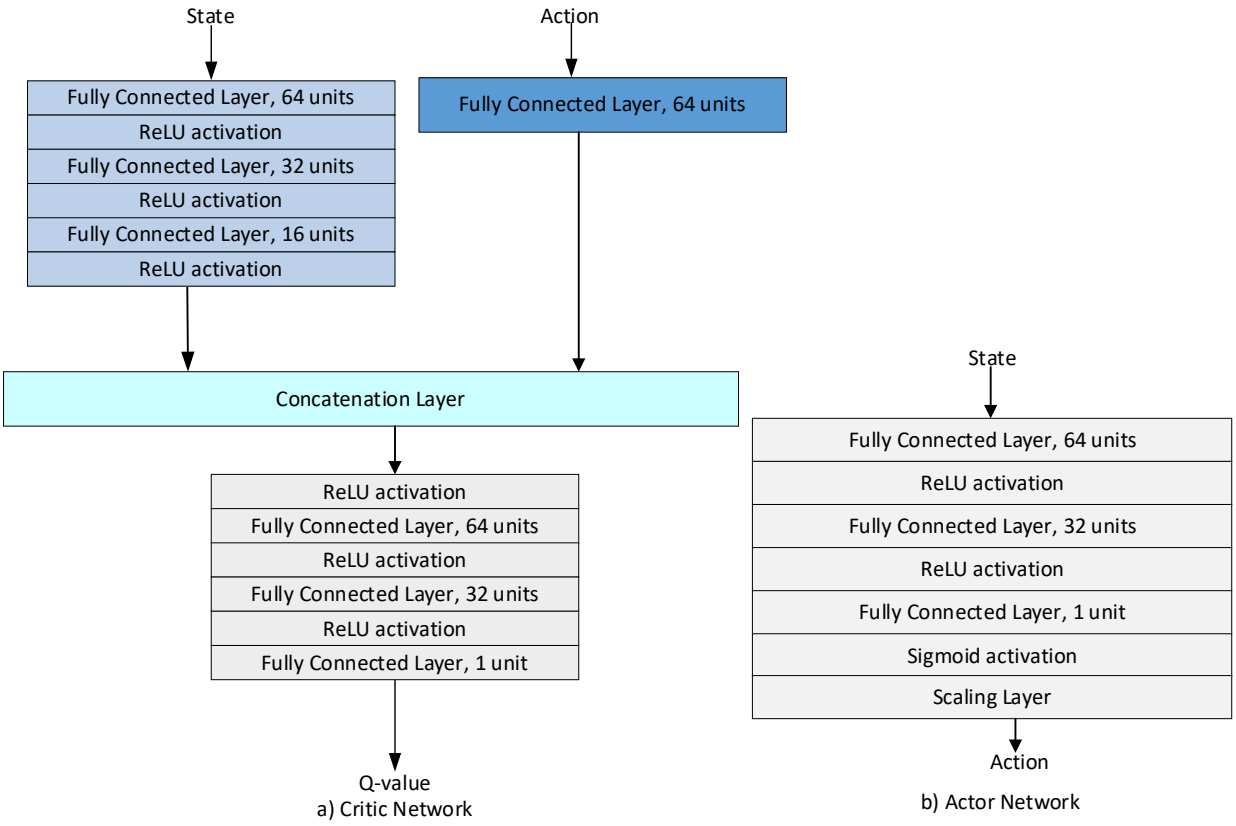


Figure 5.4 Structure of proposed a) Critic-Network b) Actor-Network.

Table 5.3 Actor-Critic Network parameters

	Optimizer	**Learning Rate**	**Gradient Threshold**	**L2 Regularization Factor**
Critic	Adam	0.001	1	0.0001
Actor	Adam	0.001	1	0.00001

Table 5.4 Training parameters setting

Discount factor	**0.99**
Experience buffer Length	10^6
Mini-batch size	128
Number of steps to look ahead	10
Target smooth factor	$5e^{-3}$
Target update frequency	2
Exploration variance	0.01
Target Policy smooth variance	0.2

5.5.3 Training an agent

The agent trains by randomly selecting mini-batches of size 128 with a discount factor of 0.99 towards the long-term reward from the replay buffer or experience buffer with a maximum capacity of 1e6. The target critics and actors are time-delayed clones of the critics and the actor networks with a smoothing factor of 0.005 that update every two agent steps during training. The agent uses a Gaussian action noise model with a specified noise variance and decay rate to explore the action space during training. The agent also uses the specified Gaussian noise model to smooth the target policy updates. Each training consists of 500 episodes, with each episode consisting of nearly 170 steps. The agent training is terminated when the agent receives an average cumulative reward of more than 800 over 100 consecutive episodes.

5.6 Benchmark Deep Deterministic Policy Gradient (DDPG)

Deep Q Network (DQN) has been a proven RL method capable of solving complex problems on par with human-level performance, as proven in Atari video games. However, DQN only solves the problem with high-dimensional observation space and low-dimensional discrete action space. For the continuous control problem that requires an iterative optimization process at every step to find the action that maximizes the action-value function, DQN can not be applied straightforwardly. The DDPG is a model-free, online, off-policy actor-critic algorithm that can learn the optimal policies to maximize the expected cumulative rewards in high dimensional continuous action spaces. While training, a DDPG agent updates the critic and actor parameters at each time step. It stores past experiences in the circular experience buffer, and the agent updates the critic and actor parameters using mini-batches of experiences selected randomly from the buffer. After that, in each time step, the action chosen by the policy with a stochastic noise model is perturbed.

Algorithm 5.2: DDPG Algorithm

Initialize critic networks $Q(s,a|\theta^Q)$ and actor-network $\mu(s|\phi^\mu)$ with random parameters θ and ϕ

Initialize target networks Q' and μ' with $\theta' \leftarrow \theta, \phi' \leftarrow \phi$

Initialize replay buffer $\mathcal{B}$

for t=1 to T, do

Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\sigma)$ and execute a in the EVCS environment

Store transition tuple $\langle s,a,r,s',d\rangle$ into $\mathcal{B}$ where d is the signal to indicate s' is the terminal state

If s' is the terminal state, reset environment state

Else randomly sample mini-batch of N transitions $\langle s,a,r,s',d\rangle$ from $\mathcal{B}$

Compute the target:

$$y(r,s',d) = r + \gamma(1-d)Q'(s',\mu'(s'))$$

Update critic Q-function by using one step of gradient descent:

$$\theta \leftarrow \text{argmin}_\theta \nabla_{\theta_i} \frac{1}{|\mathcal{B}|} \sum_{\langle s,a,r,s',d\rangle \in \mathcal{B}} \left(Q_{\theta_i}(s,a) - y(r,s',d) \right)^2$$

Update the actor policy ϕ by the deterministic policy gradient:

$$\nabla_\phi J(\phi) = \frac{1}{|\mathcal{B}|} \sum \nabla_a Q_{\theta_i}(s,a)|_{a=\mu_\phi(s)} \nabla_\phi \pi_\phi(s)$$

Update target networks:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$$

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi'$$

End for

5.7 Computational performance Comparison of DDPG and TD3

The proposed digital clones with DRL agents: the PV agent, the BES agent, and the EV agent, train individually as the agents should learn to act independently employing both DDPG and TD3 algorithms. The motive behind designing the independent agents is that they should be able to work with legacy controllers (in case only a few controllers got infected) and other trained RL agents (all legacy controllers got infected). We train the agents as configured in sections 5.4.3.3, 5.4.3.4, and 5.4.35 independently for both DDPG and TD3 algorithms.

Since we are about to engineer independent agents, there won't be a collaborative or adversarial learning paradigm such as the concept of hierarchical (global and local) rewards. Instead, we retrain the EV agent with the trained RL agents PV and BES agents so that it can upgrade the learned policy. After that, we test various combinations of DDPG/TD3 agents and legacy controllers to assess the control actions on the EVCS. Finally, the trained agents are deployed with the tendency to upgrade the policy while EVCS is running. All the computations and simulations are performed in Matlab 2022 b and Simulink 10.6 model version 5.6 in Dell XPS 15 7590 machine with i7-9750H CPU @2.6 GHz and 16GB RAM. Each agent took approximately 6.68 hours of training for the 500 episodes under DDPG and TD3. The TD3 training progress in average rewards is shown in Fig. 5.5 with stabilized reward within 99 episodes for the PV agent, 136 episodes for the BES agent, and 101 episodes for the EV agent. However, the clones trained with DDPG exhibit poor convergence stability due to its higher sensitivity towards the hyperparameter settings though it is trained with the same hyperparameters and observation spaces as in TD3. The optimal episodes for training DDPG agents are 398 for the PV agent, 22 for the BES agent, and 348 for the EV agent after analyzing the rewards and Q-values as in Fig. 5.5, 5.6, and 5.7. The incremental bias and suboptimal policy seen in DDPG training are due to the overestimation of Q-values as it updates the Q-value as in DQN, as evident in episode rewards of Fig.5.6 and estimated Q-values in Fig. 5.7. That’s the reason the DDPG has myriads of suboptimal overshoots till the end of training progression. Therefore, the near-optimal policy under DDPG training can be found in 398 epochs for the PV agent, 22 episodes for the BES, and 351 episodes for the EV agent under the horizon of 500 episodes.

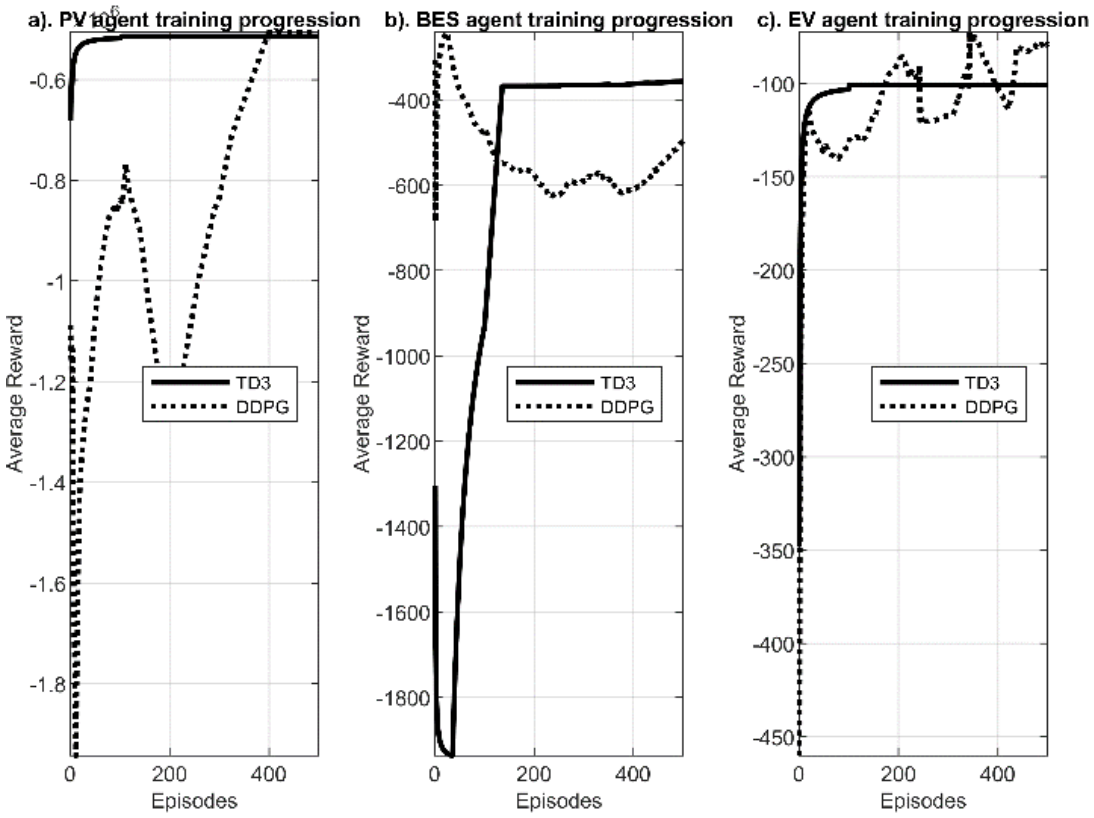


Figure 5.5 Training performance of the DDPG and TD3 Agents in terms of average rewards.

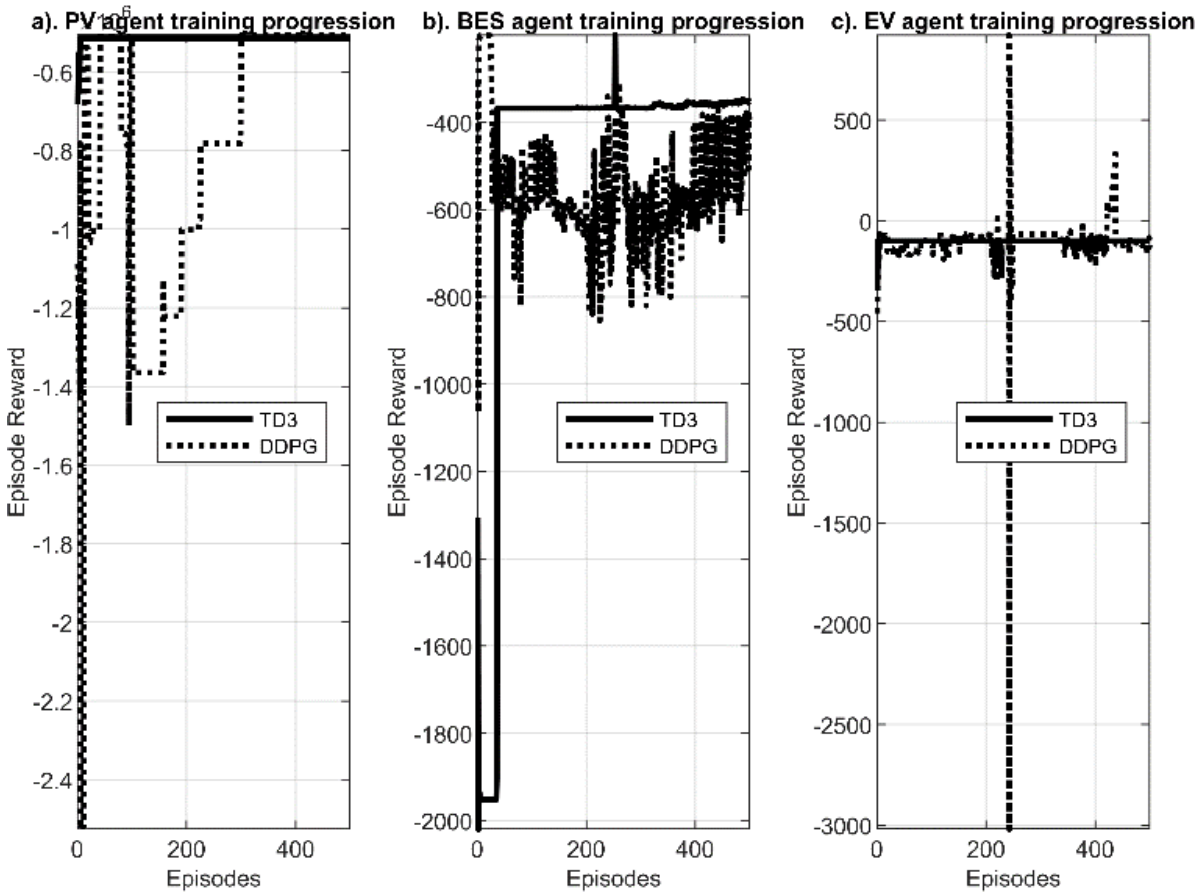


Figure 5.6 Training performance of the DDPG and TD3 Agents in terms of episode rewards.

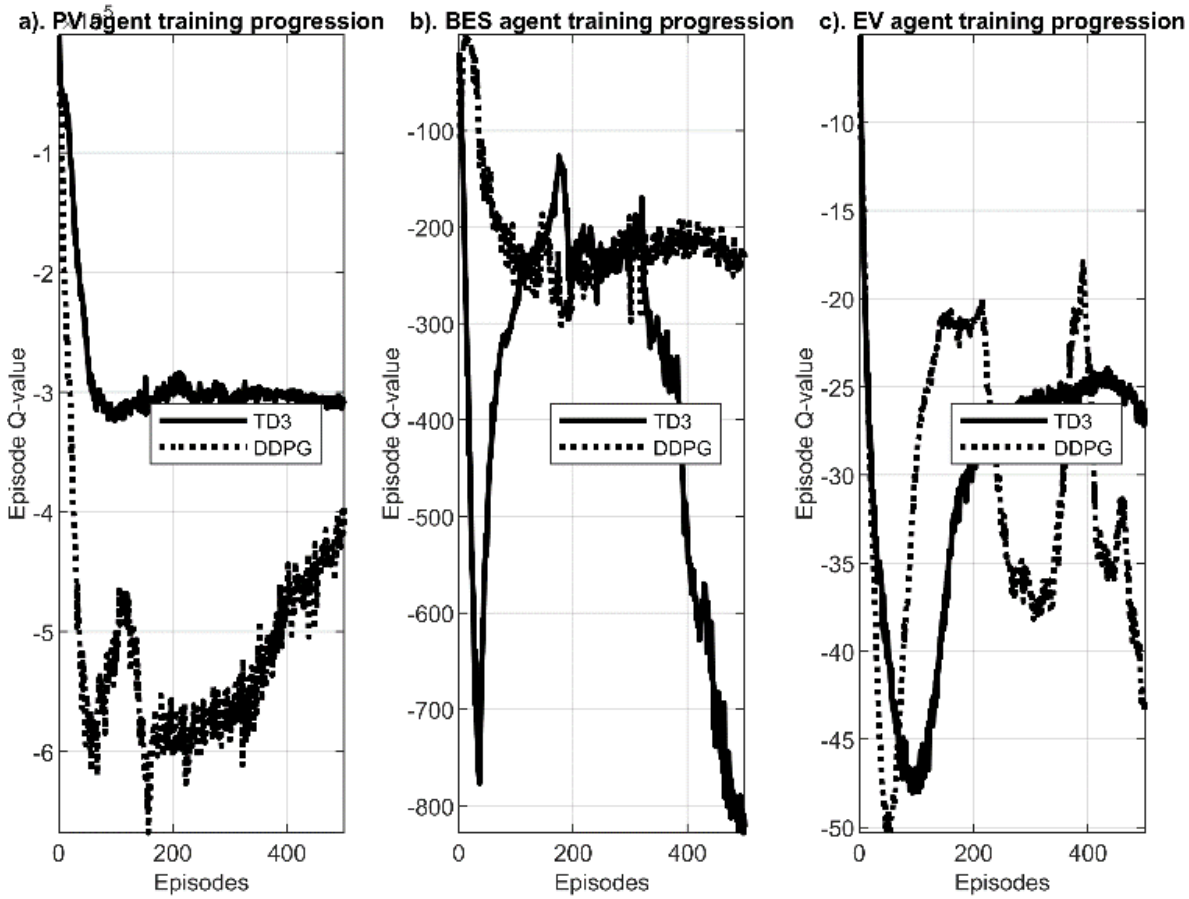


Figure 5.7 Training performance of the DDPG and TD3 Agents in terms of episode Q-values.

Unlike heavily parameterized, complex, and black box algorithms, more reliable, interpretable, and visible attack recovery or mitigation approaches are proposed to deal with the impact of cyberattacks on EVCS. The Bruteforce and Controller clone-based mitigation approaches are discussed in subsections 5.8 and 5.9.

5.8 Bruteforce mitigation

The Bruteforce mitigation is the manual or automatic override of the control signal under cyber-attack detection. This approach might be good for the stable and convergent linear time-invariant system. This approach learns the critical control signals from analyzing repeated experimentations with human domain experts. These learned signals are released as a corrective

measure to the physical controllers under threat detection. The operator can estimate the corrective duty cycles for each Controller $\mathcal{C}_{corrective}$ by examining the repeated experimentation or by using the Monte-Carlo simulation as $\mathcal{C}_{corrective} = \{\overline{\mathfrak{D}_{PV}}, \overline{\mathfrak{D}_{EV}}, \overline{\mathfrak{D}_{BES}}\}$. In case of attack detection, the operator can override the $\mathcal{C}$ with $\mathcal{C}_{corrective}$. The working logic of brute force mitigation is given below.

If $(Upper\ Threshold < \mathfrak{D}_{()} < Lower\ Threshold)$ && $(Upper\ Threshold < CF_{()} < Lower\ Threshold)$

Continue with Legacy controllers $\mathcal{C}$ *, i.e., duty=*$\mathfrak{D}_{()}$*;*

Else

Correct the duty cycle with the $\mathcal{C}_{corrective}$ *, i.e., duty =*$\mathfrak{D}_{(corrected)}$*;*

end

After careful examination of the process, the desired control signals, the duty cycle for PV, BES, and EV controller, are 0.2, 0.7, and 0.55, respectively, for the stable operation of EVCS. This approach has less complexity and less implementation cost. This method can restore the EVCS operation under a constrained environment, with the following limitations.

- This method is purely static and not intelligent (it does not have learning capability); therefore, it is not adaptive.
- Small changes in operational conditions or minor flaws can fail the model, i.e., very high failure susceptibility.
- This model is prone to failure under the APT or ransomware attack that can freeze the controllers where manual overriding is no longer an option.

5.9 Controller clone-based mitigation

We develop the concept of Controller clones/twins to deal with the freezing of controllers due to worst-case cyberattacks such as APT and Ransomware. Unlike brute force mitigation, this model has an exact clone of the controllers, meaning the same operational technologies and configurations in case one fails; the clone can take over. The controller clones are the reserved physical backup controllers preserved for the cyberattack recovery and are given by $C_{clone} = \{C\}^{N}_{i=1}$ where N is the number of operating controllers. The control logic of this method is given below.

If $(Upper\ Threshold < \mathfrak{D}_{()} < Lower\ Threshold)$ && $(Upper\ Threshold < CF_{()} < Lower\ Threshold)$

Continue with Legacy controllers $\mathcal{C}$ *, i.e., duty=*$\mathfrak{D}_{()}$*;*

Else

Replace the duty cycle with the $\mathcal{C}_{clone}$

end

For the EVCS, exact physical copies of controllers with the same configurations and operating principles are deployed under the attack detection on the operating controllers. This method outperforms the Bruteforce-based control. Also, the EVCS has to pay a huge price for attack recovery compared to the Controller clone implementation cost. Therefore, the industry can easily adopt the mitigation. However, this method has the following limitations.

- The Controller clones share the same vulnerabilities as the operating ones; hence, they can be easily exploitable.

- These are adaptive; however, they are not intelligent enough to have learning capabilities.
- Scalability is the issue.
- Changes in operational setpoints and configurations update need the retuning of the controllers.
- High economic overhead is needed to set up the entire clone of the operating controllers.

5.10 Simulation Results and Discussion

5.10.1 Type I and Type II Attacks

Fig. 5.8 summarizes the Type I and Type II attacks launched at different controllers simultaneously and at different times. The Type I attack is the low-frequency attack at the duty cycles of the controllers, while Type II is the constant attack. The BES duty cycle is found to be more vulnerable to both kinds of attacks than the duty cycles of other controllers. The Type I attack has an irreversible impact on the BES controller as opposed to the Type II attack on the BES controller and both attacks on other controllers.

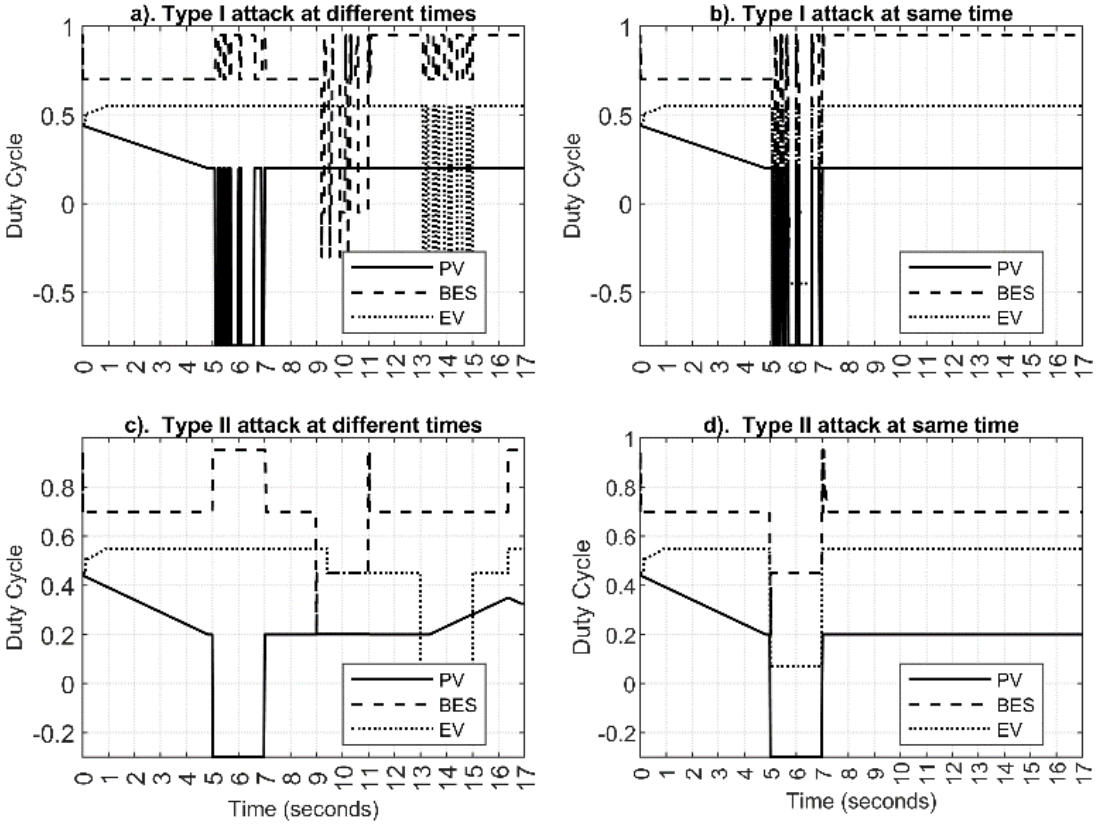


Figure 5.8 Impacts of Type I and Type II attacks on duty cycles of different controllers.

5.10.2 Mitigation results and analysis of proposed TD3-based clones vs. DDPG clones

5.10.2.1 Type I Attack on Different Times and Mitigation Analysis

The Type I attack was launched in three different controllers PV controller at 5-7 seconds, BES controller from 9-11 seconds, and EV controller from 13-15 seconds, as shown in Fig. 5.9. Tables 5.5, 5.6, 5.7, 5.8, 5.9, and 5.10 present the corresponding statistics of important electrical parameters. The Type I attack has impacted all the critical electrical parameters. It forces the power to have approx. 2.99k times the normal range, 7.5k times the normal interquartile range (IQR), and a median less than 18.4 Watts to the median at regular operation. The proposed mitigation restores the power with approximate errors of 0.002 watts in the median, 0.0001 watts in IQR, and -2.44 watts in range with the one at normal operations, as evident in Table 5.5.

Similarly, the Type I attack has an inverse impact on bus voltage with a range elevation of

approximately 158 V, IQR elevation of 1.63 V, and median reduction of 0.0052 V compared to the base operating conditions. The proposed mitigation can restore the bus voltage with approximate errors of 0 V in the median, 0.0001 V in IQR, and 0.5288 V in the range with the one at normal operations, as per Table 5.6.

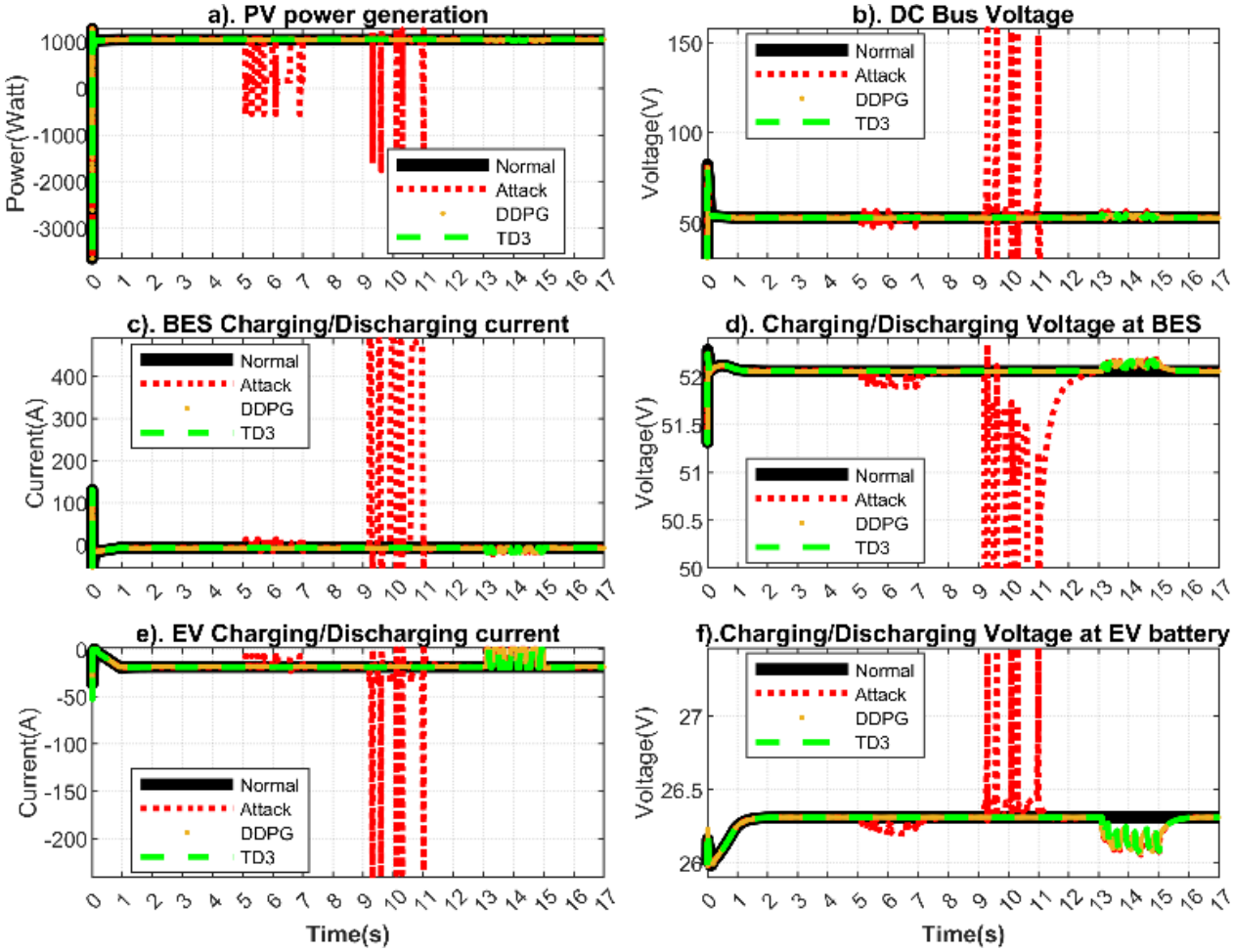


Figure 5.9 Impacts of Type I attack launched at PV controller from 5-7 seconds, BES controller from 9-11 seconds, and EV controller from 13-15 seconds and the mitigation performance during the attack

Table 5.5 PV power statistics in Watt during normal, attack and mitigation

	Range	**IQR**	**median**
Normal	[1043.59,1044.60]	[1043.593,1043.599]	1043.5996
Attack	[1768.23,1255.32]	[998.97,1043.71]	1025.1726
Mitigation	[1040.15,1043.60]	[1043.594,1043.5998]	1043.5969

Table 5.6 DC bus voltage statistics in Volts during normal, attack and mitigation

	Range	IQR	median
Normal	[52.7593, 52.761]	[52.7596,52.7607]	52.7605
Attack	[-0.23051,157.502]	[52.2102,53.8464]	52.7553
Mitigation	[52.608,53.1379]	[52.7596,52.7608]	52.7605

Also, as per table 5.7, the Type I attack has an inverse impact on BES current with a range elevation of approximately 683 A, IQR elevation of 14 A, and median increment of 0.1 A compared to the base operating conditions. The proposed mitigation can restore the BES current with approximate errors of 0.0001 A in the median, 0.0013 A in IQR, and 2.4159 A in the range with the one at normal operations.

Table 5.7 BES current statistics in Ampere during normal, attack, and mitigation

	Range	IQR	median
Normal	[-6.947,-6.9435]	[-6.9435,-6.944]	-6.9452
Attack	[-193.294,489.396]	[-8.816,- 6.358]	-6.8273
Mitigation	[-9.143,-6.723]	[-6.9456, -6.944]	-6.945

Likewise, the Type I attack has an inverse impact on BES voltage with a range elevation of approximately 4.3434 V, IQR elevation of 0.5747 V, and median decrement by 0.0914 V compared to the base operating conditions. The proposed mitigation can restore the BES current with approximate errors of 0.000 V in the median, 0.0102 V in IQR, and 0.0251V in the range with the one at normal operations evident from Table 5.8.

Table 5.8 BES voltage statistics in Volts during a normal, attack, and mitigation

	Range	IQR	median
Normal	[52.0586,52.0588]	[52.0586,52.0588]	52.0587
Attack	[47.9982,52.3418]	[51.5319,52.1066]	51.9673
Mitigation	[52.0586,52.0839]	[52.0586,52.069]	52.0587

Table 5.9 shows that the Type I attack has an inverse impact on EV current with a range elevation of approximately 241.2919 A, IQR elevation of 8.0385 A, and median increment of

2.0851 A compared to the base operating conditions. The proposed mitigation can restore the EV current with approximate errors of 3e-4 V in the median, 0.0015 A in IQR, and 4.0534 A in the range with the one at normal operations.

Table 5.10 implies that the Type I attack has an inverse impact on EV voltage with a range elevation of approximately 1.4074 V, IQR elevation of 0.1438 V, and median decrement of 0.0602 V compared to the base operating conditions. The proposed mitigation can restore the BES current with approximate errors of 4.0000e-04 V in the median, 0.0193 V in IQR, and 0.0472 V in the range with the one at normal operations.

Table 5.9 EV current statistics in Ampere during normal, attack, and mitigation

	Range	**IQR**	**median**
Normal	[-18.674,-18.668]	[-18.674,-18.669]	-18.6713
Attack	[-241.298,5.26e-5]	[-19.531, -11.488]	-16.5862
Mitigation	[-18.9473,-14.887]	[-18.674,-18.671]	-18.6716

Table 5.10 EV voltage statistics in Volts during normal, attack, and mitigation

	Range	**IQR**	**median**
Normal	[26.312,26.313]	[26.312,26.313]	26.3126
Attack	[26.056,27.465]	[26.199,26.344]	26.2524
Mitigation	[26.265,26.313]	[26.293,26.313]	26.3122

5. 10.2.2 Type I Attack simultaneously on all controllers and mitigation analysis

The Type I attack was launched simultaneously in three different controllers at 5-7 seconds, as shown in Fig. 5.10. The Type II attack that launches at different times has impacted all the critical electrical parameters.

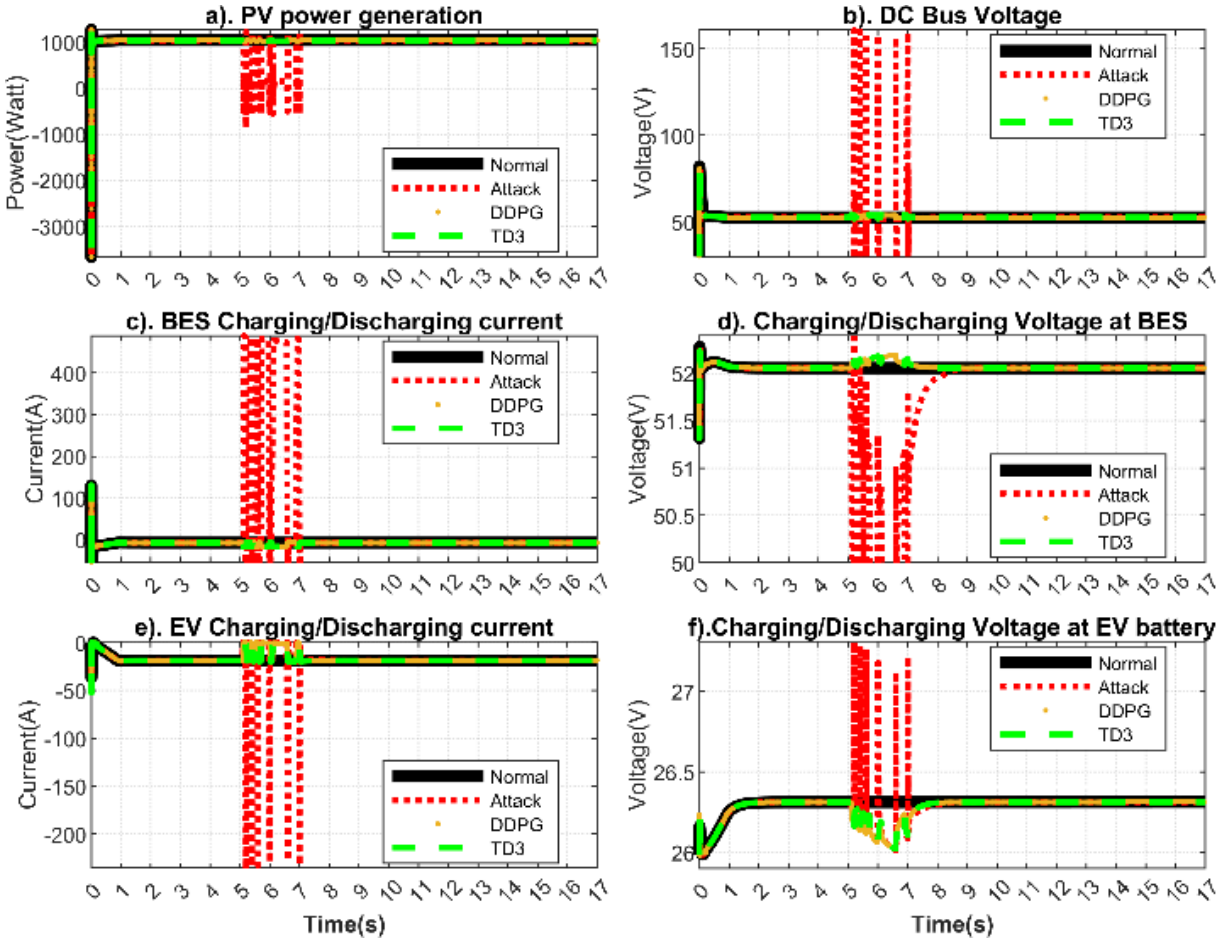


Figure 5.10 Impacts of Type I attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds and the mitigation performance during the attack.

5.10.2.3 Type II Attack on different times and mitigation analysis

The Type II attack was launched in three different controllers PV controller at 5-7 seconds, the BES controller from 9-11 seconds, and the EV controller from 13-15 seconds, as shown in Fig. 5.11.

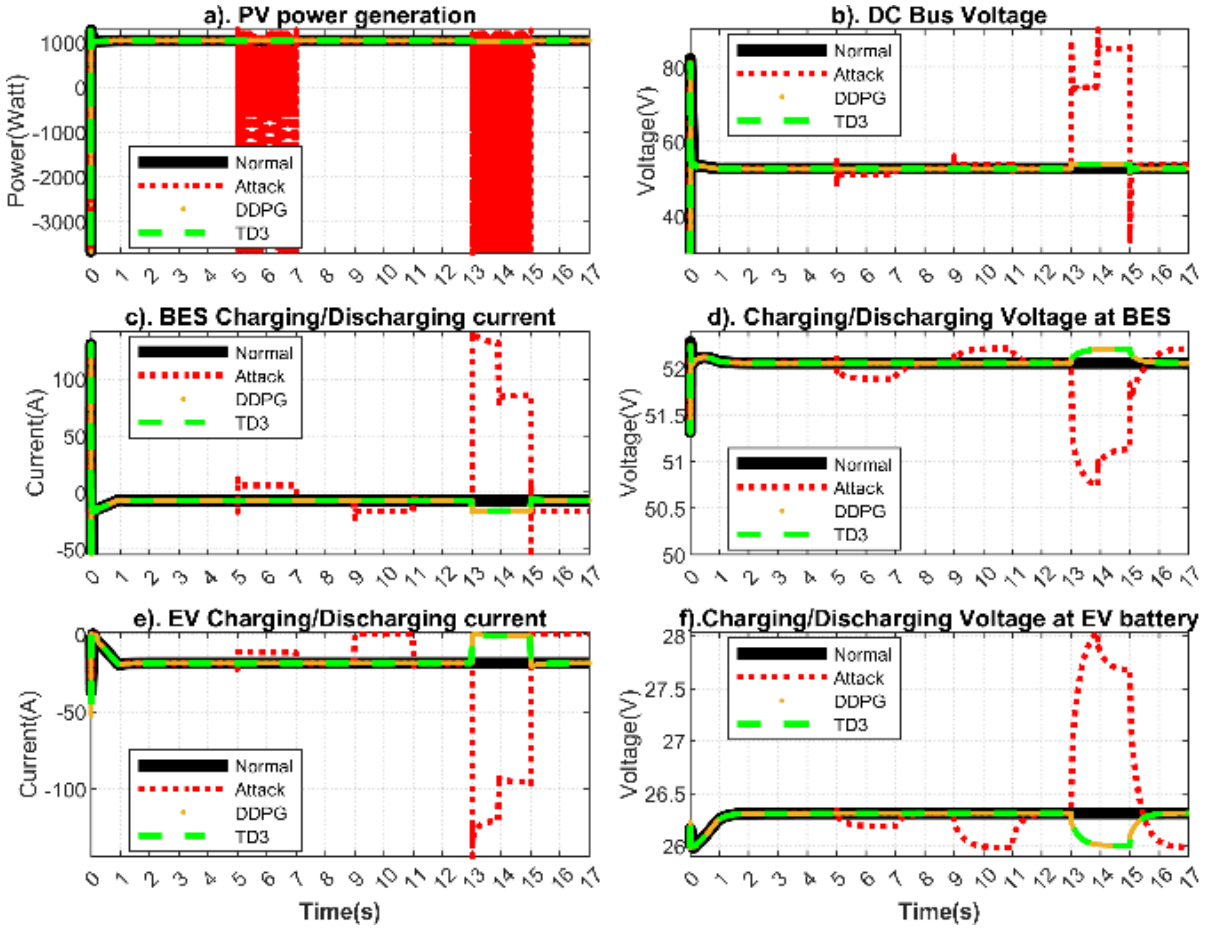


Figure 5.11 Impacts of Type II attack launched at PV controller from 5-7 seconds, BES controller from 9-11 seconds, and EV controller from 13-15 seconds and the mitigation performance during the attack.

5.10.2.4 Type II Attack simultaneously on all controllers and mitigation analysis

The Type II attack was launched simultaneously in three different controllers at 5-7 seconds, as shown in Fig. 5.12.

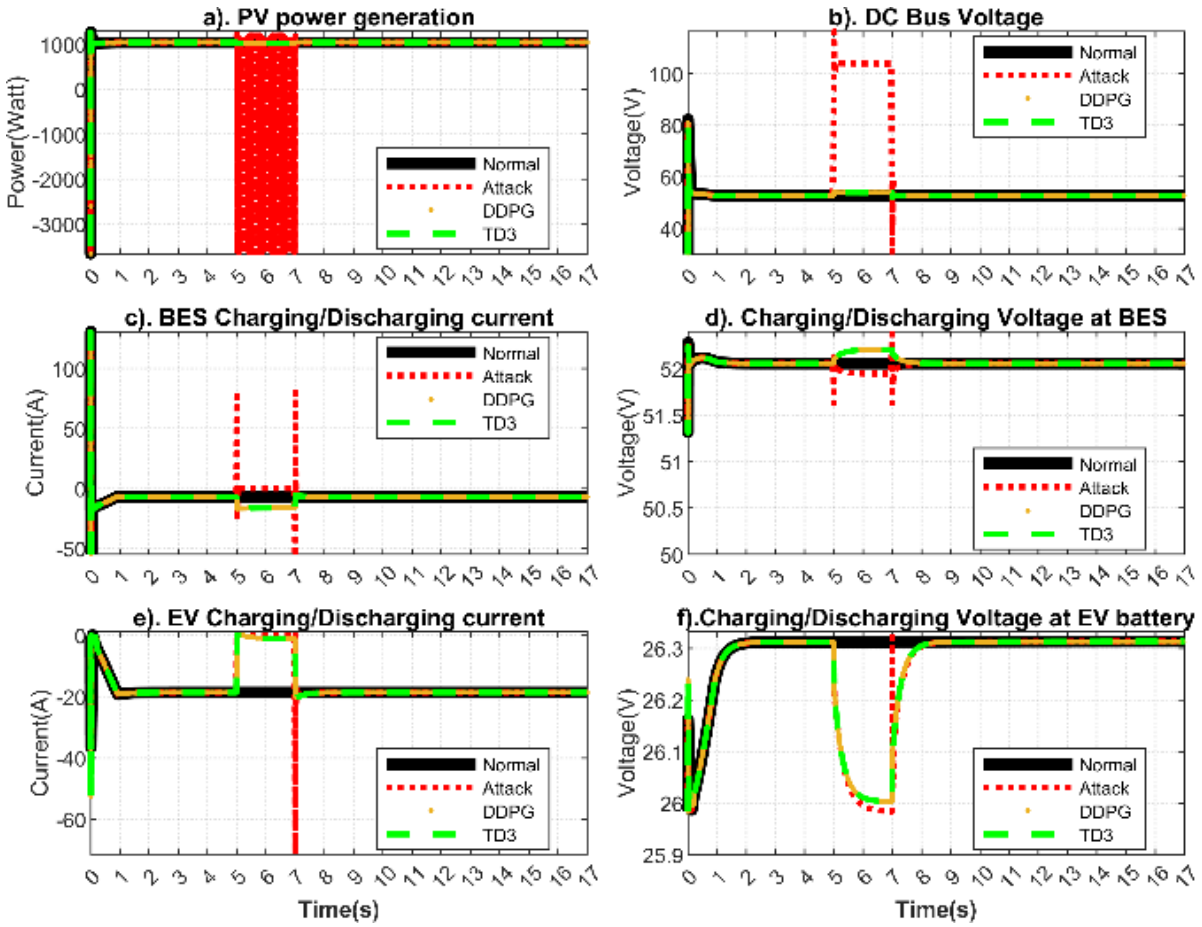


Figure 5.12 Impacts of Type II attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds and the mitigation performance during the attack.

5.10.2.5 Performance Comparison of Various Proposed Methods

Fig. 5.13 depicts the control actions, i.e., duty cycles of the legacy controllers employed in the EVCS and trained clones with DDPG and TD3 algorithms. Under normal operation, the legacy MPPT controller at PV stabilizes the duty cycle to 0.2, around 4.8 seconds. In contrast, the digital clone trained with DDPG and TD3 settles at a duty cycle of 0.4 and 0.495, respectively, from the beginning, as in Fig. 5.13 a. Fig. 5.13 b clearly shows the superior control action of TD3 duty cycle converged to 0.99 from the beginning as compared to DDPG BES clone converged to same after 4.2 seconds. The digital clone of the EV controller trained with DDPG and TD3 has produced the same control action, i.e., the duty cycle of 0.5. The legacy controllers are manually tuned heuristic-

based control that stabilizes the EVCS operation; however, it takes some time to stabilize the duty cycles. Unlike, the control actions taken by DDPG and TD3 are data-driven as they optimize the control policy based on maximizing the expected long-term rewards. Therefore, clones are free to choose the duty cycles that perfectly drive the normal operation of the EVCS.

Table 5.11 presents the features of the proposed mitigation methods with respect to other related works. Our proposed air-gapped TD3-based mitigation has surpassed the various state-of-the-art methods in attack detection with online mitigation with embedded intelligence.

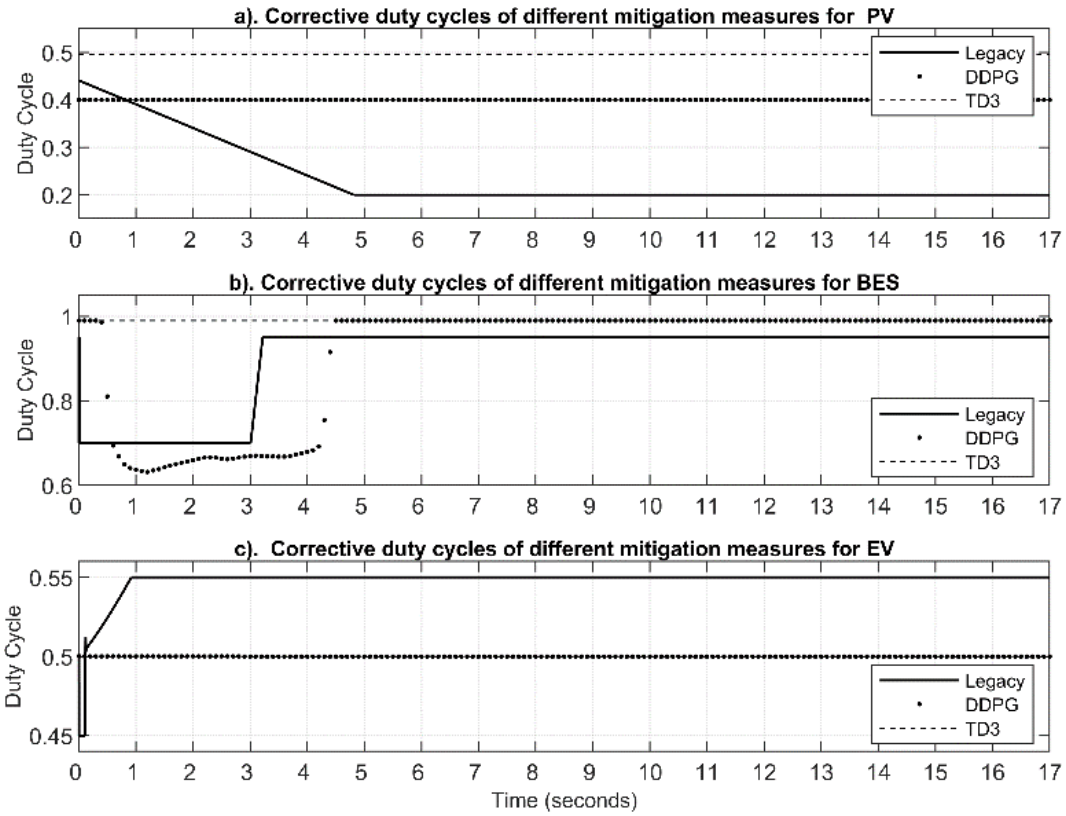


Figure 5.13 Control actions of Legacy controllers, DDPG clone, and TD3 clone mitigations.

Table 5.11 Comparison between the proposed and the STATE-OF-THE-ART algorithms

Solution	Attack detection	Coordinated Attacks	Online Mitigation	Embedded Intelligence	Air gapped
TD3 (our work)	√√	√√	√√	√√	√√
DDPG (Benchmark)	√√	√√	√√	X	√√
HIDS for EVCS	√√	√√	X	√√	√√

[40]					
NIDS for EVCS [7]	√√	√√	X	X	X
Weighted attack defense tree [46]	√√	√√	X	X	X

5.10.3 Mitigation results and analysis of the Bruteforce and controller clone

Fig. 5.14 depicts the control actions, i.e., duty cycles of two proposed methods under the Type-II attack at all controllers from 5-7 seconds. For the PV duty cycle correction, the Bruteforce mitigation provides a constant duty cycle of 0.2 no matter what happens to the process. In contrast, the controller clone settles at a duty cycle of 0.2 after nearly 5 seconds. Hence, one can implement the heuristics-based Bruteforce method without needing additional controllers for simplicity. In contrast, the controller clone-based approach provides the same functionalities as the deployed controllers with added cost. Fig. 5.15 presents the performance of the proposed mitigation methods during the Type I attack launched at different controllers simultaneously on different operating parameters of the EVCS. Fig. 5.16 shows the performance evaluation of proposed mitigations against the Type II attacks. Table 5.12 compares the performance of the proposed models with superior mitigation performance for Controller clones. The performance scores such as R-squared, Mean Absolute Error (MAE), Median Absolute Error (MDAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) of the Mitigation duty cycles are obtained with respect to the duty cycle of PV, BES and EV controller during the normal operating conditions.

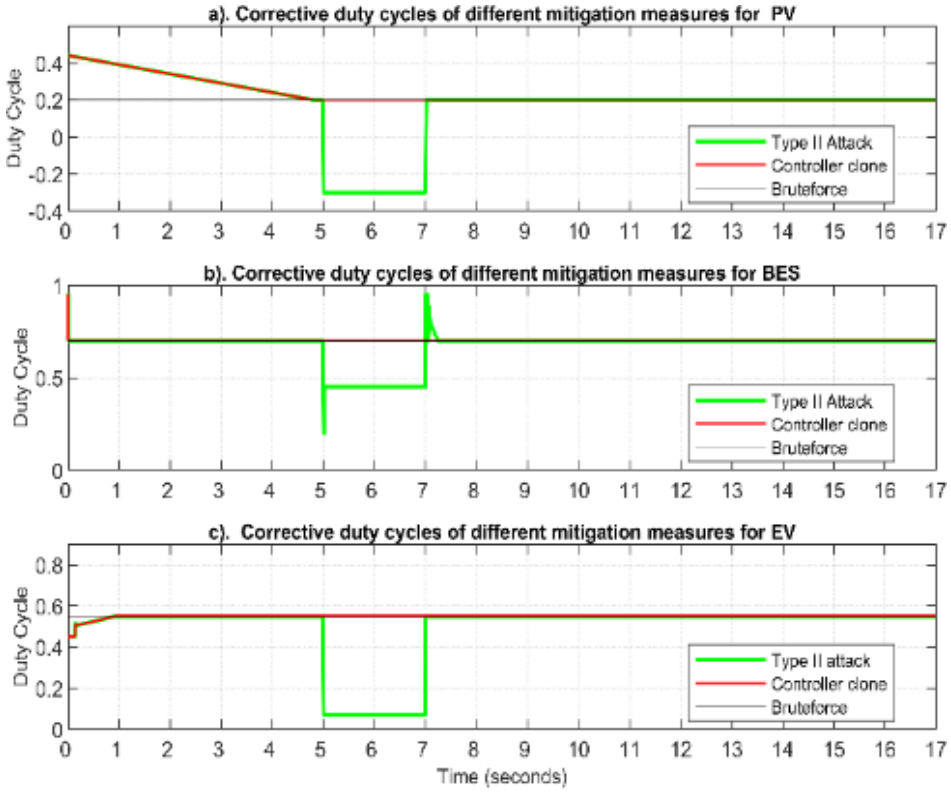


Figure 5.14 Corrective duty cycles of Bruteforce and Controller clone mitigations

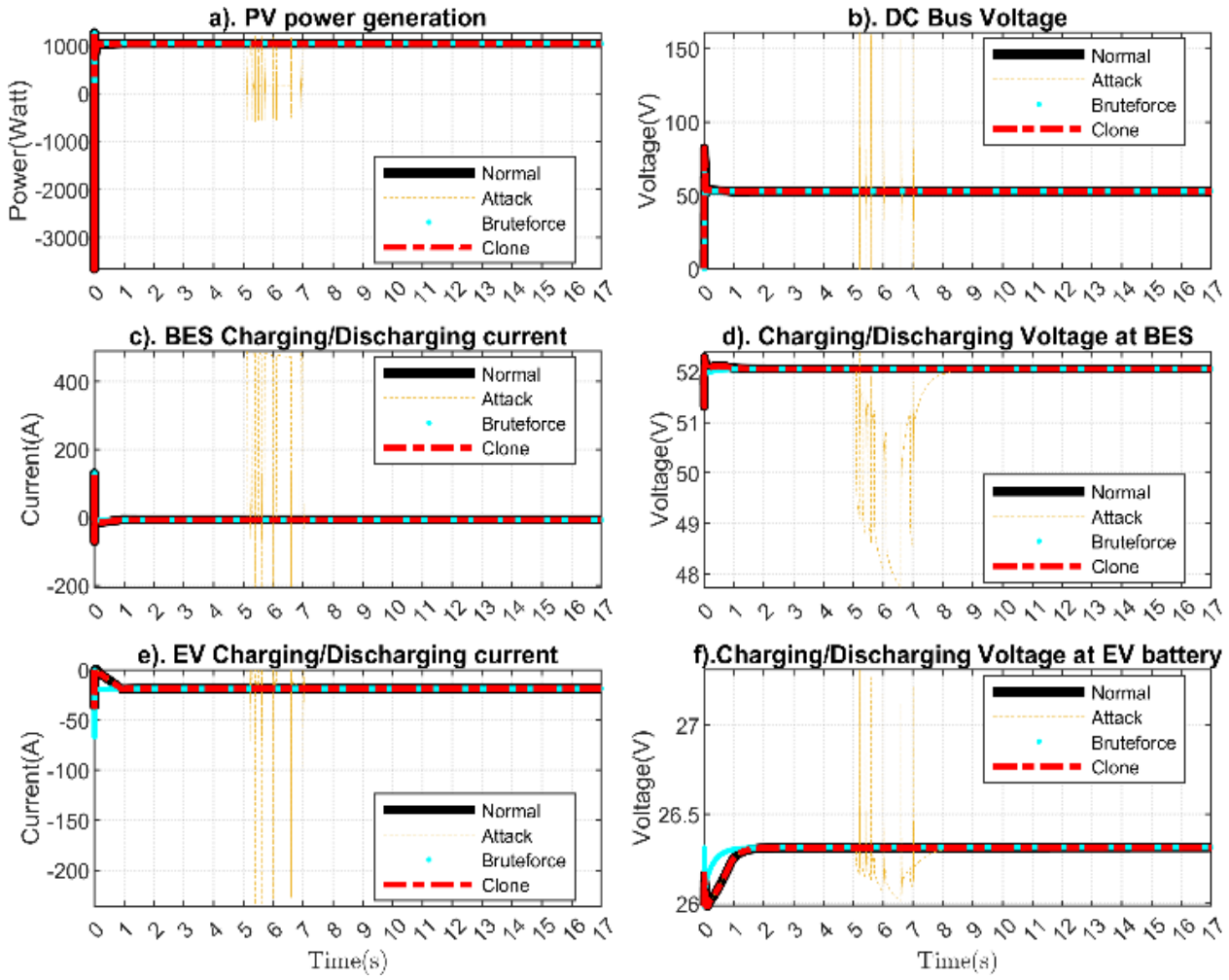


Figure 5.15 Mitigation performance during the Type I attack launched at PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds.

Table 5.12 Performance comparison of proposed Bruteforce and Clone Mitigations

	Bruteforce(PV,BES,EV)	**Clone(PV,BES,EV)**
R-squared	-0.317,-0.001,-0.037	1,1,1
MAE	0.039,0,0.002	0,0,0
MDAE	0,0,0	0,0,0
MSE	0.006,0,0	0,0,0
MAPE	0.11,0,0.04	0,0,0

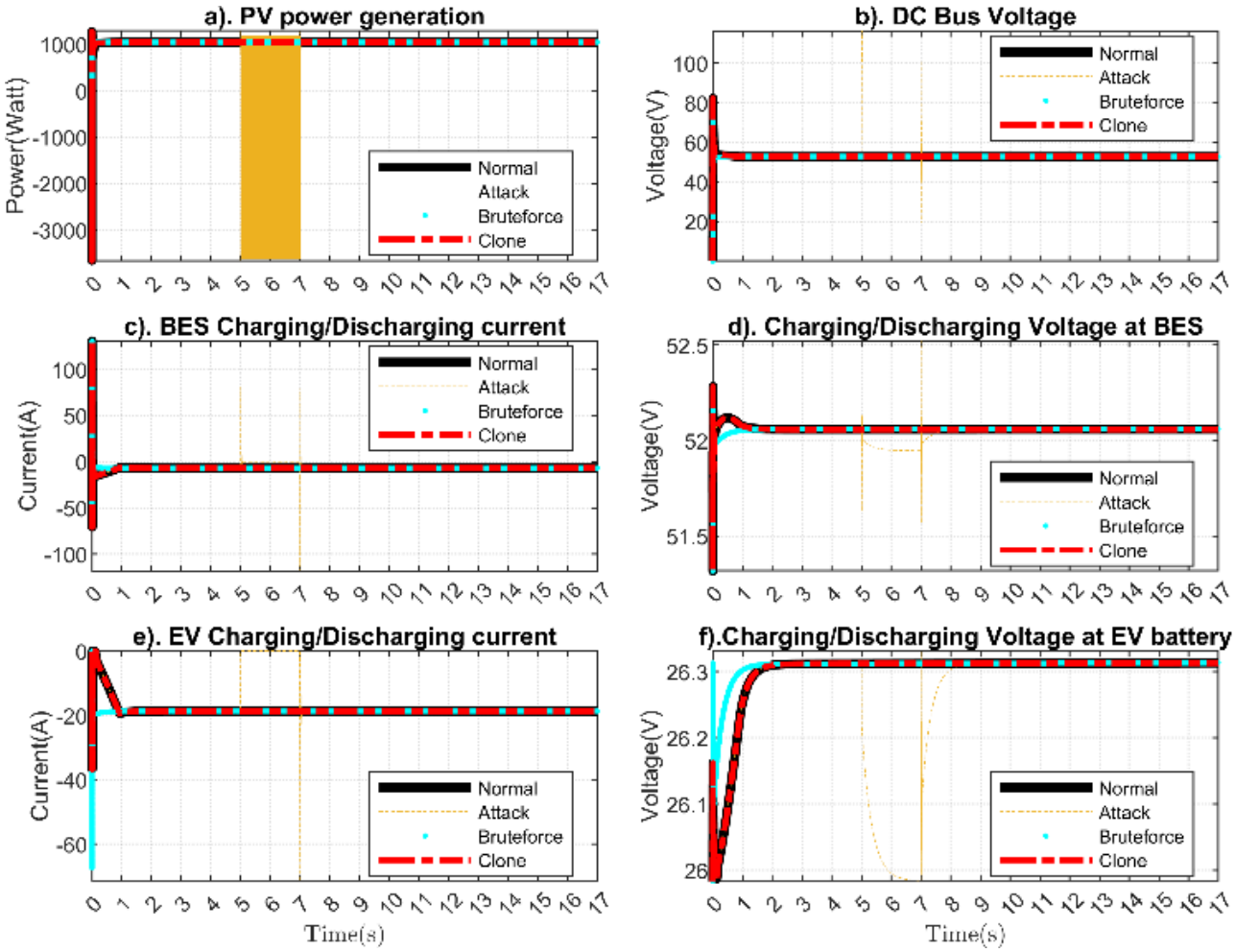


Figure 5.16 Mitigation performance during the Type II attack launched at the PV controller from 5-7 seconds, BES controller from 5-7 seconds, and EV controller from 5-7 seconds.

5.11 Chapter Conclusion

The chapter proposed and assessed different classical and data-driven mitigation approaches for the cyber-physical attack at EVCS. The repetitive low-frequency attack (Type-I) on all controllers, at different times or simultaneously, has adverse impacts on the critical functionalities of all controllers with the tendency to damage the EVCS with an upsurge/down surge in electrical signals. The proposed mitigations successfully restore the EVCS operation by correcting the control signals of legacy controllers. The constant attack (Type-II) on controllers at different times or simultaneously corrupts and damages the electrical components related to

the legacy control actions. The proposed methods attempt to correct the control signals. The proposed methods successfully responded to the APT attacks with the least error margin. Clone-based mitigation surpasses the performance of Bruteforce mitigation as examined with various error measures. The proposed Bruteforce-based mitigation is simple but can't adapt to changes in system dynamics. The Controller clone can restore the operation; however, it shares the same vulnerabilities as the legacy controllers. Data-driven approaches were proposed to solve the problem of the previous two methods. The proposed TD3-based software clones can take over the legacy controllers under APT attacks or even under anomalous behavior. The TD3-based clones are superior to DDPG-based clones in terms of convergence, stability, hyperparameter sensitivity, and mitigation actions. The proposed multiple clones can learn and relearn the control policy online, resulting from changes in the EVCS environment or configurations. This is not the case in traditional legacy controllers. The detection engine deploys the agent based on the infection of the legacy controller. The sophistication and accuracy of the detection engine can be upgraded by using AI. The proposed clones successfully restored regular operation under the APT attacks and system anomaly on the legacy EVCS controllers. Finally, the proposed digital clones with TD3 outperform the benchmark DDPG-based clones in terms of stability, convergence, and mitigation performance.

The next Chapter presents the deep learning perspective and threat intelligence in the CAV paradigm.

Chapter 6 A Deep Learning Perspective on Connected Automated Vehicle Cybersecurity and Threat Intelligence

6.1 Introduction

The connected and autonomous vehicle (CAV) is the next-generation mobility service—powered by intelligent automation and robust communication--aimed at replacing human-maneuvered vehicles with the software agent matching or even exceeding the human-level intelligence, control, and agility with the least decision errors possible. US National Safety Council (NSC) [120] reported a 24 % spike in roadway death rates from 2019 despite miles driven dropping 13%, the highest increment in 96 years in the US since 1924. Most of the time, it's from human error. NSC estimated that 4.8 million additional roadway users were seriously injured, costing $474 billion in 2020. The next generation of transportation and mobility envisions a safe, reliable, agile, automated, trustworthy, and service-based mobility architecture. The architecture should be able to eliminate human errors by using intelligent decision-making software agents based on the situational and behavioral information collected by sensors and transceivers through communication. Apart from that, the service-based architecture removes the concept of vehicle ownership and includes more diversity, such as disabled and older people. CAV is the evolving technology to achieve future mobility and transportation goals.

Commutation vehicles nowadays are not merely electromechanical entities but also complex software agents with electronics [121]. Connected means the vehicle exchanges data between the systems and networks (to other vehicles and infrastructures) for predictive maintenance, dynamic insurance policy, passenger information, fleet management, comfort, and situational and behavioral awareness [122]. Fully autonomous means the vehicle conducts dynamic driving tasks automatically in real time without the driver's intervention [123].

The connected vehicle generates 25 GB of data per hour, even at a lower level of autonomy. Integrating RADAR, LIDAR, Camera, Ultrasonics, Motion sensors, GNSS, and IMU into the vehicle can generate 40 Gbit/s data leading up to 380 TB to 5100 TB of data in a year [124]. This wealth of high-volume, high-speed data needs gigantic storage, intense computation, and astute processing. As the data volume vehemently upsurges, software, hardware, and data privacy and security become critical. Increased connectivity elevates the attack surfaces of the CAV, while automation lacks the sophisticated human-level agility and intelligence for threat mitigation. The inherent vulnerabilities come from the untested supply chain, such as hardware, software, and infrastructures.

Deep learning has been unprecedently successful in deciphering the complex nonlinear spatiotemporal pattern of highly stochastic data. Given the volume, veracity, and velocity of the data, deep learning could be handy in designing cyberattack detection and mitigation in the CAV environment. Fig. 6.1. shows the current trends of publications queried under "(((ALL=(Connected Vehicle)) OR ALL=(Automated vehicle)) AND ALL=(Cybersecurity)) OR ALL=(Deep learning)" in the web of science for 2017 through 2020. It resulted in 127,042 publications so far, with growing interest per year.

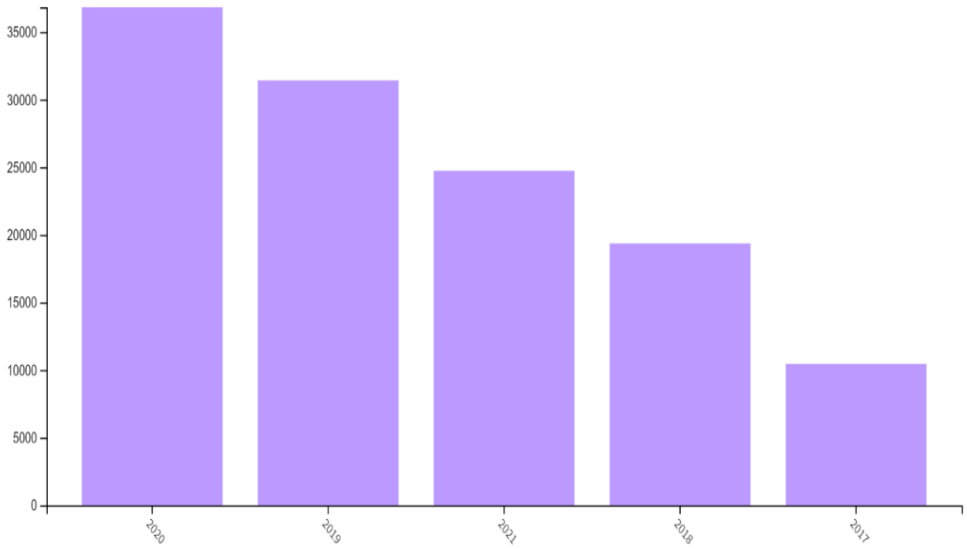


Figure 6.1 The number of publications on connected automated vehicles over the last five years.

This chapter deals with the deep learning perspective on CAV cybersecurity and threat intelligence. Also, We proposed novel end-to-end deep CNN-LSTM-based computational intelligence for cyberattack detection and classification in the CAV ecosystem. The proposed model has been successfully trained, tested, and evaluated on the CAV-KDD dataset and compared against other deep learning models such as Deep Neural Networks (DNN), Convolutional Neural Network (1D-CNN), LSTM. The proposed model outperformed the aforementioned deep learning models in terms of various performance metrics and increased model complexity.

6.2 CAV technological enablers: Automation and Connectivity

The key technological enablers for CAV are Automation and Connectivity. Vehicle driving automation system performs part or all of the dynamic driving tasks (DDT). The Society of Automotive Engineers (SAE) defined the six levels of automation for vehicles ranging from no

automation (Level 0) to full automation (Level 5) in its 2021 release [125]. In Levels 0-2, the driver drives the entire or part of DDT, while in Levels 3-5, ADS performs the entire DDT upon engagement. Fig. 6.2. shows the overall connectivity architecture with a wireless network interface, physical interface, and In-vehicle network in between and is adapted from [126]. CAV is co-evolving with next-generation network architectures and communication protocols. It can exploit the latency, speed, and bandwidth of recent cellular communication, such as 5G. The recent advancement in Wi-Fi 6 could be used in place of high user density. Also, inbuilt GPS has been widely used for navigation.

Moreover, Bluetooth, RFID, ZigBee, and V2X communication have extended the range of connectivities and applications. The in-vehicle network has mainly a high-speed infotainment system for information dissemination and entertainment; and a Powertrain network for core functionalities of the vehicle. These are mostly composed of electronic control units (ECUs) connected through local control area network (CAN) buses. Physical network interface provides ports to connect the phone, USB, CD, AUX from the infotainment system, and ODB-II from the powertrain system. This ODB- II can be extended via physical or wireless network interfaces to OEM, drivers, and computerized intelligence in the vehicle. The hitherto advancement in connectivity and automation has enabled the following vehicle functionalities as per the US department of transportation (DoT) [127].

a) Collision warning: forward collision warning, lane departure warning, rear cross-traffic warning, and blind-spot warning

b) Collision intervention: automatic emergency braking, pedestrian automatic emergency braking, automatic rear braking, and blind-spot intervention.

c) Driving control assistance: adaptive cruise control, lane-centering, and lane-keeping

assistance.

d) Additional systems: Automatic high beams, backup camera, and automatic crash notification.

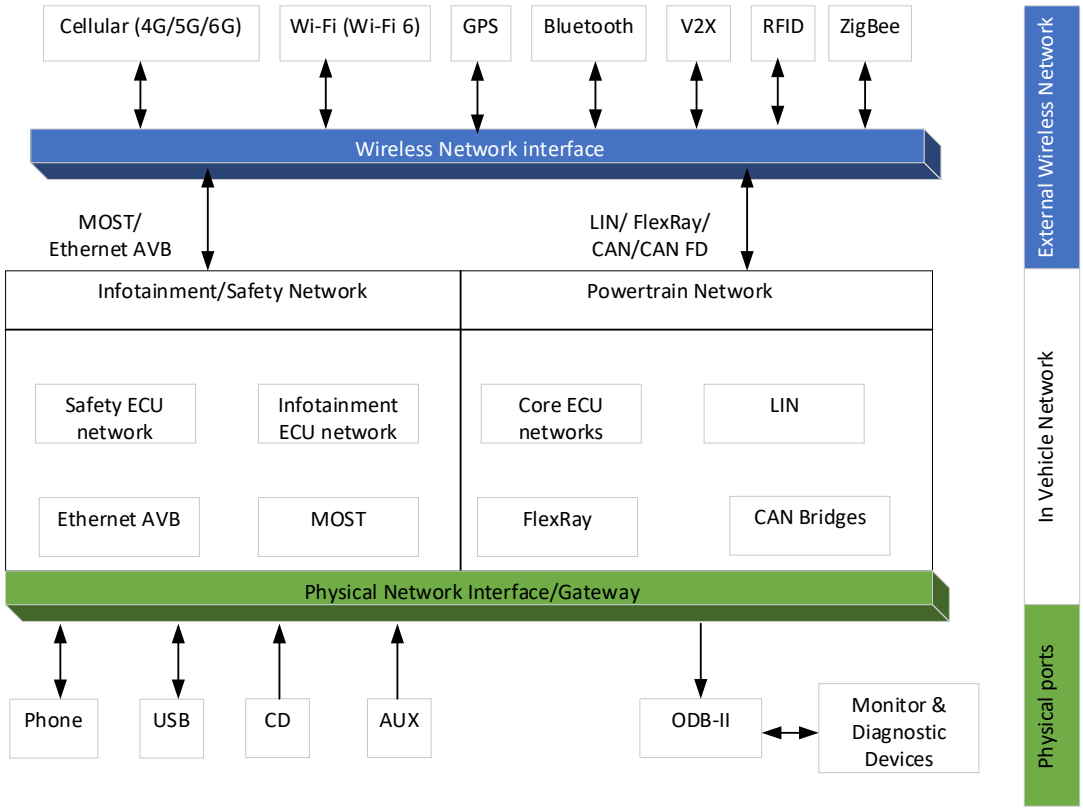


Figure 6.2 Connectivity of the CAV.

6.3 CAV threat landscape and threat intelligence

Threat intelligence begins with identifying the assets and then finding the weighted utility to the assets, i.e., threat landscape. Assets are entities with specific utilities and hence add values to the system. The value comes from the cost of creating it and the competition to make it easily available. Therefore, from a game-theoretic perspective, there is always competition to exploit the utility, i.e., the risk of biased usage, which creates vulnerabilities. An attacker can exploit the

vulnerabilities by using social engineering and reverse engineering. The electromechanical vehicle, while adopting the evolving network architecture and automation—so-called CAV--migrates all the vulnerabilities related to the processes, protocols, supply chains, and software from the incumbent technologies. Furthermore, CAV has vulnerabilities or risks originated from communication, automation, IT, OT, and physical system. Here, we will briefly explain the cyber vulnerabilities of low-level sensors [128] and vehicle control modules.

6.3.1 In-vehicle (low-level sensor) cyber vulnerabilities

GPS: The transparent architecture of GPS, its open standard, and free accessibility are the main reasons for generating spoofing and jamming attacks on GPS.

Inertial measurement units (IMUs): IMUs provide velocity, acceleration, and orientation data using accelerometers and gyroscopes. The gyroscope and inclination sensors measure the road gradient and adjust the speed for safe maneuvering. The spoofed data can generate a false control signal for speed control. Also, the jamming of the sensors may disrupt the vehicle's autonomous speed adjustment.

Engine control sensor: These sensors monitor the dynamics of the engine, such as temperature, airflow, exhaust gas, and engine knock, and are connected to CAN.

Tire Pressure Management System (TPMS): TPMS has not been used in decision-making but is physically accessible to outsiders.

LiDAR sensors: are used to generate the 3D map of the vehicle's environment for localization, obstacle avoidance, and navigation. Laser beams can fool liDAR sensors.

Cameras (stereo- or mono-vision) and infrared systems: These are used for static and dynamic obstacle detection, object recognition, and 360-degree information with other sensor fusion.

Cameras contain the charge-coupled device (CCD) or complementary Metal Oxide Semiconductor (CMOS) sensor that can be partially disabled from a 3- meter distance using low-powered lasers.

6.3.2. Vehicle control modules

All modern vehicles use engine control units (ECU) to acquire, process and control electronic signals. ECUs are roughly categorized into powertrain, safety systems, body control, and data communications. The powertrain is the brain of the ECU that controls transmissions, emissions, charging systems, and control modules. Safety systems are responsible for collision avoidance, airbag deployment, active braking, etc. Body control controls the electric windows, AC, mirrors, immobilizer, and locking. Data communications control the communication between different communication modules. The networking of ECUs can be done through either CAN buses or FlexRay. The key ECUs in CAV in descending orders of importance are as follows [128], [129].

a) Navigation control module (NCM)
b) Engine control module (ECM)
c) Electronic brake control module (EBCM)
d) Transmission control module (TCM)
e) Telematics module with remote commanding
f) Body control module (BCM)
g) Inflatable restraint module (IRM)
h) Vehicle vision system (VVS)
i) Remote door lock receiver
j) Heating, ventilation, and air conditioning (HVAC)
k) Instrument panel module
l) Radio and entertainment center

6.3.3. Security analysis of CAV threats

CAV can have around 100 million lines of code across 50-70 ECUs. As the number of lines of code grows, It's infeasible to perform careful security implications. Some security incidences and their analysis are presented here [122].

a). Remotely control a vehicle: The attacker exploits the vulnerability in the cellular system and lands on the infotainment system. In most vehicles, the infotainment system has a driver with information such as service schedules, tire pressure, etc. The infotainment system connects with the CAN bus that connects all the ECUs. Therefore, it is possible to enter through the infotainment system and inject or spoof malicious signals. E.g., ECUs controlling steering or brake.

b). Disable the vehicle: exploiting flaws in authentication, authorization, and access control in smart devices and apps to activate AC, windows, and windshield to drain the battery.

c). Remotely unlock the vehicle/theft: exploit known vulnerabilities in the keyless entry system using SDR. Hackers unlocked the car door remotely and started the engine in the Mercedes Benz-E class in 2020. The manufacturer generally uses symmetric keys between the key fob, entry system, and ignition keys. An attacker can sniff the radio frequency between the key fob and entry system either by brute force or as a man-in-the-middle attack. Later the symmetric key can be compromised by replay attack or reverse engineering. The problem became humongous when some leading vehicle manufacturers used the same master cryptographic keys along the model line.

d). Safety conditions: Panic attacks such as Mobileye and Tesla X hack fooled the autopilot system to trigger the brakes and steer into an oncoming vehicle.

e). Vehicle tracking/monitoring: extract patterns or fingerprints from the data.

f). Weaponizing the vehicle

g). Malware: bots for crypto-jacking or DDoS.

h). Ransomware could be a huge problem to be dealt with in the future CAV.

i). Distribution of illicit goods.

6.3.4. Attack surfaces

While adopting the evolving network architecture, communications, and AI-powered automation, the orthodox electro-mechanical vehicles migrate all the vulnerabilities of incumbent technologies.

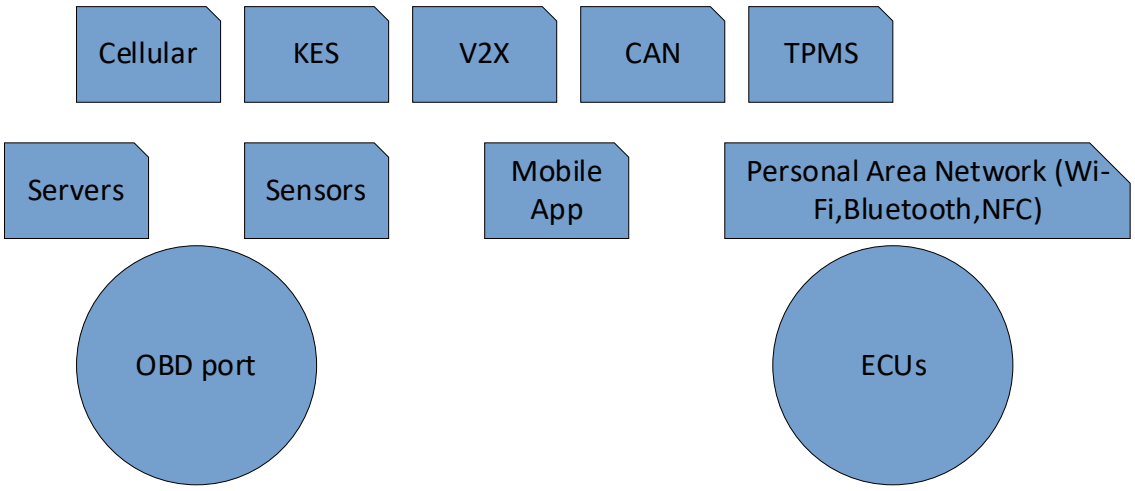


Figure 6.3 Key attack surfaces of CAV.

With the elevated sophistication, CAV also inherits elevated attack surfaces and attack vectors. These attack vectors are the specific methods, paths, or processes through which the CAV can be exploited. Insider threats such as Levandowski trade secret trial between Waymo and Uber [130], Cyberattack into V2X communications [131], Sensor spoofing and exploitation [132], Dumpster diving for data: acting as a honeypot, Supply chain, and third-party risks are some of the prominent threats in the CAV ecosystem. Fig.6.3 enlists the prime attack surfaces.

6.3.5 Organizational risks to the CAV ecosystem

The organizational risks imposed on the CAV ecosystem are well documented in [133]. The convergence of IT, OT, and physical security is a challenging issue in any cyber-physical system, including the CAV ecosystem. The interconnections, interactions, and co-impacts of attack on these eccentric systems should be analyzed and evaluated. Dealing with big data (high volume, high speed, wide variety) and extracting inferences in the CAV ecosystem needs high computational capacity, storage, and processing to deal with the multimodal data from different sensors. Data communication between multiple nodes, servers, and systems impose security and privacy risks. The divergent nature of stakeholders, such as different vendors of CAV, OEM, ITS, V2X, and its data privacy policy, might not allow CAV actors to collaborate in threat detections and mitigations. The cyber-physical security protocols, enterprise policies, and regulations still have to go a long way in the CAV ecosystem.

6.4 CAV threat mitigation: anomaly detection and classification with deep learning

"Deep Learning is building a system by assembling parameterized modules into a (possibly dynamic) computation graph and training it to perform a task by optimizing the parameters using a gradient-based method," as quoted by Yann Le Cunn, ACM Turing awardee and a pioneer in deep learning in AAAI-20 event [134]. Graphs can be defined dynamically by input-dependent programs. Output computation may not necessarily be the feedforward; it might be some minimizing energy functions (inference model) [135]. The designer has complete freedom to choose learning paradigms such as supervised, reinforced, self-supervised/unsupervised, and objective functions such as classification, prediction, and reconstruction. Often limitations of supervised learning are mistakenly seen as limitations of deep learning. If the cake is intelligence, self-supervised learning is the bulk of the cake; supervised learning is the icing on the cake, and

RL is the cherry on the cake. The next revolution in AI won't be supervised nor reinforced [136].

Deep learning has been an exciting paradigm for anomaly detection and classification in various cyber-physical realms, such as industrial control systems, smart grids, SCADA-controlled systems, etc. [137]. Now the specific state-of-the-art applications of deep learning in CAV cybersecurity are summarized. In [138], Generative Adversarial Network-based IDS has been used to detect the anomaly in ECU by analyzing the CAN message frame, specifically the message identifier and frequency. The dataset was recorded from the OBD-II port of an undefined vehicle. The authors modified the firefly algorithm to find the optimal structure of the Generator network. Finally, they claimed the superior accuracy of the proposed model to the PSO- and GA- optimized GAN. However, the paper does not have much information regarding the training time, data size, data samples, computational complexity, etc.

In [126], a Deep learning-based LSTM autoencoder has been implemented to design IDS for CAN and central network gateways using car hacking and UNSW-NB15 datasets, respectively. Statistical features such as total count, mean, and standard deviation are extracted from the network packets. The proposed model claimed to outperform some of the decision tree and SVM-based classifiers. It's unlikely to claim DL model can detect zero-day attacks since the supervised ML model cannot detect and classify the data that have never been trained.

In [139], authors use GAN for designing IDS capable of learning unknown attacks in the in-vehicle network. They extracted the CAN bus data for normal and attack categories using raspberry pi and simple hardware in the OBD-II port. Instead of converting all the CAN data to an image (make real-time detection at stake due to increased processing), only CAN IDs are converted into the image by using one-hot encodings. For training, the first discriminator uses the normal and abnormal CAN images extracted from the actual vehicle, while the second discriminator uses

normal and random noise. The generator and discriminator compete to increase their performance, and the second discriminator can detect fake images similar to real CAN images. The proposed model, however, has used only CAN IDs as the main feature to identify the attack from the non-attack. Converting data into images hinders the real-time detection of IDS. Also, the model can't detect operational flaws from the attack.

In [140], ResNet-inspired DCNN has been used for sequence learning of broadcasted CAN IDs using the same dataset as in [139]. However, they are more interested in finding the pattern in the sequence of IDs than individual IDs. 29 bits IDs are recorded for every 29 consecutive IDs forming a 29x29 grid image ready to go into the DCNN and correspondingly labeled as an attack or no attack. They claimed that the DCNN seems to be more efficient in sequence learning than LSTM for this problem. This model needs high computational power and cannot detect unknown attacks.

Yu [141] proposed a novel self-supervised Bayesian Recurrent Autoencoder to detect adversarial trajectory in Sybil attacks targeting crowdsourced navigation systems. It uses time-series features of vehicle trajectories and embeds the trajectories in a latent distribution space as multivariate random variables using an encoder-reconstructor. This distribution is used to reconstruct the authentic trajectories and compared with the input to evaluate the credibility score. The author claimed that this model improves the baseline model by at least 76.6 %.

6.5 Frontiers in deep learning :Advancement and Future)

The challenges of deep learning: Supervised models need extensive data labeling, while reinforcement learning needs a massive number of interactions. Very slight modifications in fewer pixels and even a small change in rules in the environment can err the model. The inefficacy of the deep learning-based models is rooted in the assumption of "independent and identically

distributed (i.i.d)" data. This assumes that the training data capture all the stochasticity of the real dynamic environment and that observations are independent. The learner model should evolve accordingly to capture the dynamics of changing environments. Deep learning models are data-hungry; future deep learning should be envisioned to learn with fewer samples and fewer trials. For that model should correctly and broadly understand the environment before learning the tasks. Deep learning models are very poor at abstraction, and reasoning needs humongous data to learn a simple task. Symbolic AI has proven to be much better at reasoning and abstraction. Deep learning models are good at providing end-to-end solutions but miserable at breaking them down into interpretable and modifiable subtasks.

Deep learning is said to have achieved system I natural intelligence, i.e., just associative or mapping intelligence. For example, a human driver navigates to the neighborhood with visual cues that have been used a hundred times before without looking up at the direction or map. Also, he could use a map, direction, reasoning, and logic while navigating to the new environment to get to the destination. The first is system I cognition, while the second is system II cognition [142].

The pioneer deep learning scientists pointed out the following roadmaps for the future AI to be more conscious (system II cognition) at NeurIPS 2019:

- Handling the out-of-distribution (o.o.d) nonstationarity data in the environment
- Systematic generalization
- Consciousness prior
- Meta-learning and localized change hypothesis for causal discovery
- Cosmopolitan DL architectures

6.5.1. Meta-learning

When we start learning some new tasks, we merely start from scratch rather we try to use prior experiences ($\theta_i \in \Theta$) from the prior known tasks ($t_j \in T$). Where Θ is the discrete, continuous, or mixed configuration space, T is the set of all known tasks. For example, a human driver can easily drive in a completely new environment. Along the way of learning specific tasks human brain also learns how it is learning. These prior learning experiences from the tasks, if applied to learn new tasks, could bring one step closer to getting the cognitive power of system II. As a result, this new model would learn the new tasks with sparse data in a short time. Meta-learning or learning to learn is the science of transferring learning experiences, metadata, from the broader tasks to learn a new task with the least information in the least possible time [143]. The meta-data embody the prior learning tasks and the learned models in terms of exact algorithm configurations, hyperparameters, network architectures, and the resulting performance metrics such as accuracy, training time, FAR, F1-score, prior weights, and measurable properties of tasks (meta-features). Once meta-data is collected, a machine needs to extract and transfer knowledge of the meta-data to search for the optimal models to solve the new tasks. Paper [143] explains how meta-learners learn from the model evaluations, such as task-independent recommendations, configurations of space design, and its transfer techniques, including surrogate models and warm, started to multitask learning. Instead of the base learner, where the model adapts to the fixed apriori or fixed parameterized biases, meta-learners dynamically choose the correct biases [144].

The meta-learning tends to transfer knowledge learned from different environments to learn a new task with the least training as opposed to data-hungry supervised learning heavily biased due to i.i.d assumption. The evolution of transfer learning may help the machine achieve system II cognition like the human. The working principle of the Meta-learning algorithm is presented

below.

Step 1: Make a set of prior known tasks: $t_j \in T$

Step 2: Make a set of configurations resulting from the learning task t_j such as hyperparameter settings, network architecture, and so on : $\theta_i \in \Theta$

Step 3: Prior evaluation measures (accuracy, FAR, training time, cross-validation) of each configuration θ_i to task t_j: $P_{i,j}(\theta_i, t_j)$

Step 4: Assign a set of all prior scalar evaluations $P_{i,j}(\theta_i, t_j)$ of configuration θ_i on task t_j to P: P= $\{P_{i,j}(\theta_i, t_j)\}$

Step 5: evaluate the performance $P_{i,new}$ on new task t_{new} and assign it to P_{new}: $P_{new} = \{P_{i,new}\}$

Step 6: Now the meta-learner L is trained on P' to predict recommended configurations Θ^*_{new} for new task t_{new}, where $P' = P \cup P_{new}$

Step 7: L is the learning algorithm derived from meta-learning to learn a new task

6.5.2. Federated learning

Federated learning (FL) is a machine learning framework where multiple nodes collaboratively train a model with local data under the orchestration of the centralized service provider/server [145]. Each node does not transfer or share locally stored data; instead transfers the focused updates for immediate aggregation. This way, a learning model can harness the privacy, security, regulatory and economic benefits [146]. In FL, we define N set of CAVs ready to collaborate $\{V_1, \ldots., V_N\}$ from different vendors with corresponding decentralized and isolated data $\{D_1, \ldots., D_N\}$. The conventional ML/DL model pulls up all the data $D = D_1 \cup \ldots. \cup D_N$ and train the learning model L with D. In contrast, FL does not pulls up data D instead it share some model

inference/parameters to train the learning model L.

The future CAV industry is envisioned with numerous CAVs from multiple vendors. With the lack of fully developed protocol standards, cross-vendor trust issues discourage data sharing among competing vendors. The various FL application in the CAV domain can be found in [147]–[150]. Article [147] describes how a piece of falsified information from a single CAV could disrupt the training of the global model. [148] proposed the dynamic federated proximal (DFP) based FL framework for designing the autonomous controller of the CAV. DFP is said to account for the mobility of CAVs, wireless fading channels, and non-iid and unbalanced data. While solving the privacy leakage problem, FL has inherent vulnerabilities such as model inversion, membership inference, etc. [149] proposed Byzantine-fault-tolerant (BFT) decentralized FL with privacy preservation in the CAV environment. Blockchain-based FL for CAV operations is proposed in [150]. Non-iid data distributions among multiple nodes, unbalanced datasets, and communication latency are some of the challenges being solved in FL [145], [146], [151].

6.6 End-to-end deep CNN-LSTM architecture for CAV cyberattack detection

We propose the novel deep CNN-LSTM architecture for attack detection and classification in the CAV environment as per Fig. 6.4. Generally, CNN can learn the high dimensional spatial information of the feature space and may fail to capture distant temporal correlation. LSTM, on the other hand, can capture the temporal correlation by learning the sequence. Thus, the stacked model can learn the Spatio-temporal features of the learning problem. The CNN-LSTM model is expected to perform better by learning hierarchical feature representations and the long-term temporal dependencies of the huge data.

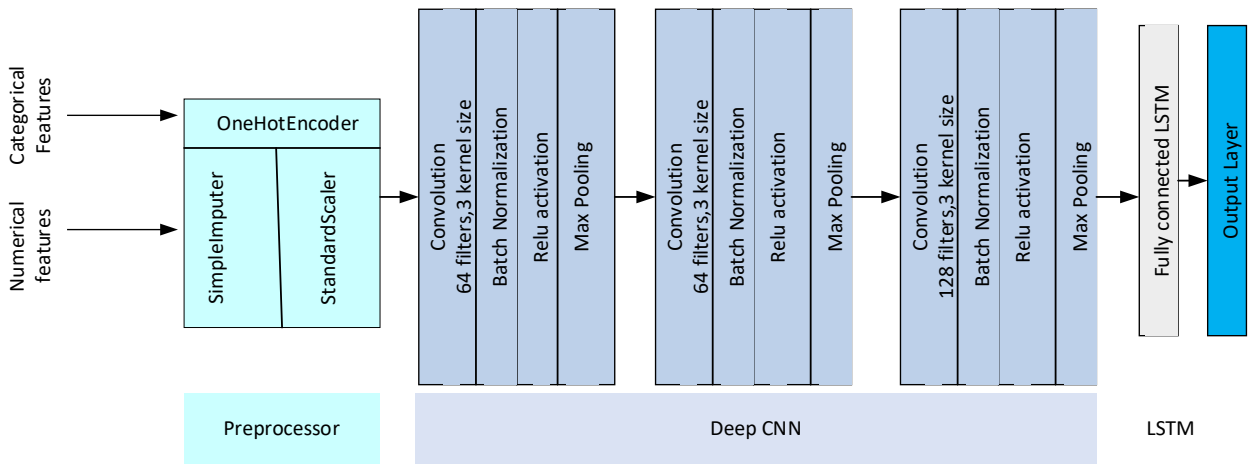


Figure 6.4 Proposed end-to-end deep CNN-LSTM architecture.

The proposed end-to-end Deep CNN-LSTM architecture pipeline has a preprocessor, a deep CNN layer, a fully connected LSTM, and an output layer.

a). Preprocessor: The preprocessor transforms the features into a machine-learning-compatible format. Most datasets contain mixed features such as numerical (integer, float) and categorical (nominal, ordinal) datatypes. Deep learning algorithms perform the computation only in integer or float features. Therefore, all the categorical features should be transformed into numerical forms. OneHotEncoder transforms the nominal features into binary formats. However, the high cardinality features would be encoded with elevated dimensionality. The SimpleImputer transformer deals with the dataset's missing values for numerical features. Moreover, the standard scaler standardizes the features by implementing zero mean and unit variance. Finally, the preprocessor outputs the features ready to fetch to the Deep CNN architecture.

b). Deep CNN architecture: This layer is generally implemented to extract the spatial information using its kernel and present the high-level features. Being able to capture the local patterns, 1D CNN is the popular algorithm for time series classification/regression successfully tested in natural language processing, the audio industry, and anomaly detection [152]. The *1D*

convolution layer creates 64 kernels of size three that convolve with the inputs over a single spatial or temporal dimension. The filters determine the dimensionality of output space, while the kernel determines the length of the 1D convolution window. The *Batch Normalization Layer* normalizes the filter's output using the mean and standard deviations of the current batch of inputs. The activation function used is *ReLU* for solving the exploding and vanishing gradient of its other compatriots, such as sigmoid, tanh [33]. *MaxPolling1D* downsamples the normalized filters' output by taking the maximum value over the spatial window of pool size four that extracts the high-level features. This marks the completion of a single block of deep CNN architecture, and we have added a similar block twice to extract more high-level features.

c). Fully connected LSTM: The 1D CNN generally extracts the local temporal information, and it is hard to capture all the long-term sequential correlations. That is where fully connected *LSTM* comes in handy to capture long-term sequential relations. The details of the LSTM model are explained in our previous journal work [40]. The LSTM layer has 64 LSTM units.

d). Output layer: The output layer has three nodes-belonging to three different classes- to evaluate the probability of the sample belonging to each class. The probability sums to one with the highest probability indicating the predicted class taken care of by the *softmax* [153] activation function. For one-hot encoded output classes, *categorical cross-entropy* [154] is used.

6.6.1 performance analysis

6.6.1.1 Dataset

The CAV-KDD dataset is adapted from the KDD99 [155] dataset, a well-known benchmark for intrusion detection. KDD99 dataset includes normal connections data and simulated attack data in a military network environment. Authors of [123] adapted using 10% of KDD99 train data and 10% of KDD99 test data to form the CAV-KDD train and test dataset. The train data and test data

are mutually exclusive, meaning the model has never experienced the test data during model fitting. There are three kinds of data, normal data refers to the normal packets, Neptune and Smurf refer to the simulated DoS attacks. The reason for choosing only these three types is that deep learning models are data-hungry and need massive sample data to capture the distribution of the dataset. Further, we preprocess and refine the dataset that will be implementable in a deep learning environment. Table 6.1. represents the distribution of the dataset while training and testing the CNN-LSTM. 30 % of training data is held to validate the model for the hyperparameter tuning. Table 6.1 indicates the class imbalance in the CAV train—20% belonging to the Normal category, 22% belonging to the Neptune attack, and 58% belonging to the Smurf attack – that inherently induces data biases in the learning model. Fig. 6.5 presents the variance captured by the singular values over the 20 samples, which is interpreted as the information captured by the prominent features. The four singular values, i.e., four prime features, can contribute to 92.90% of data variance. Singular value decomposition (SVD) is the popular dimensionality reduction technique that projects the m-dimensional data (m-columns/features) into a subspace with m or fewer dimensions without losing the essence of the original data [155].

Principal component analysis (PCA) is the dimensionality reduction technique that uses the SVD to project data from hyperspace to lower dimensional space and extracts the dominant patterns in the matrix [156]. Fig. 6.6 presents the PCA with two principal components over the 90,000 data samples showing tightly overlapped subspaces. In this notion, it's hard for any linear classifier to draw the non-linear boundaries between different classes. Therefore, a non-linear classifier such as deep learning could be handy. Deep learning is handy when one expects minimal or no feature selections since it can make good decisions with hyperdimensional feature space.

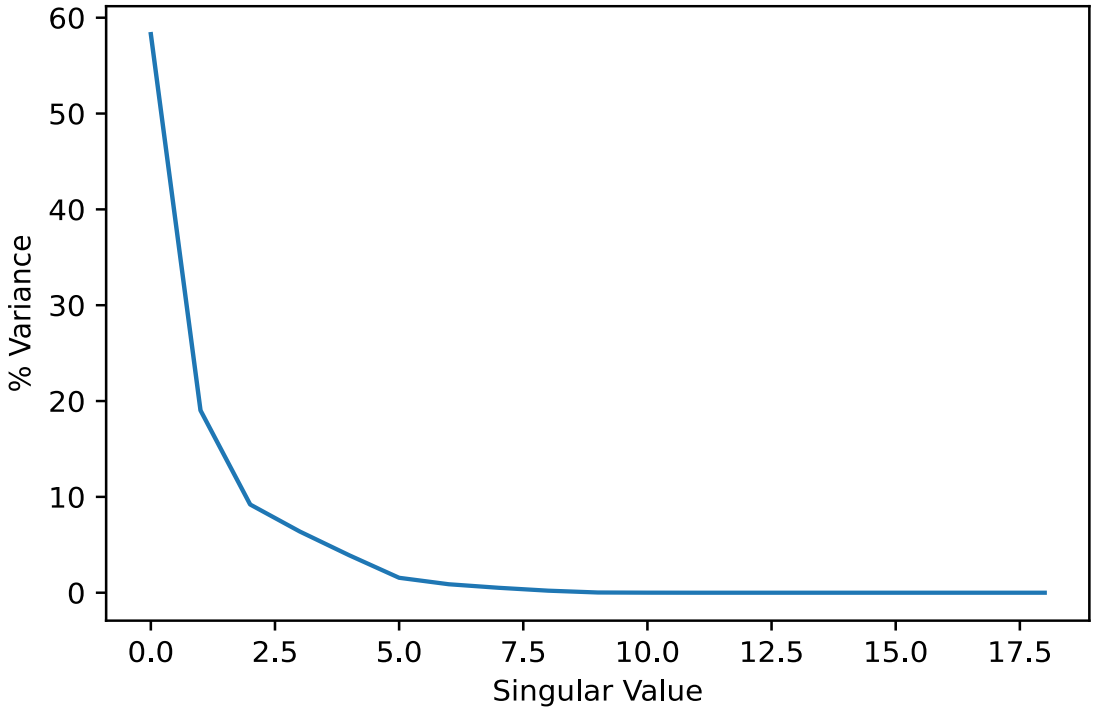


Figure 6.5 Variance captured by the Singular values.

Table 6.1 Data distribution for CAV

Data/Attack Label	CAV Train	CAV Test
Normal	97,262	60,590
Neptune	107,201	58,001
Smurf	280,790	164,091

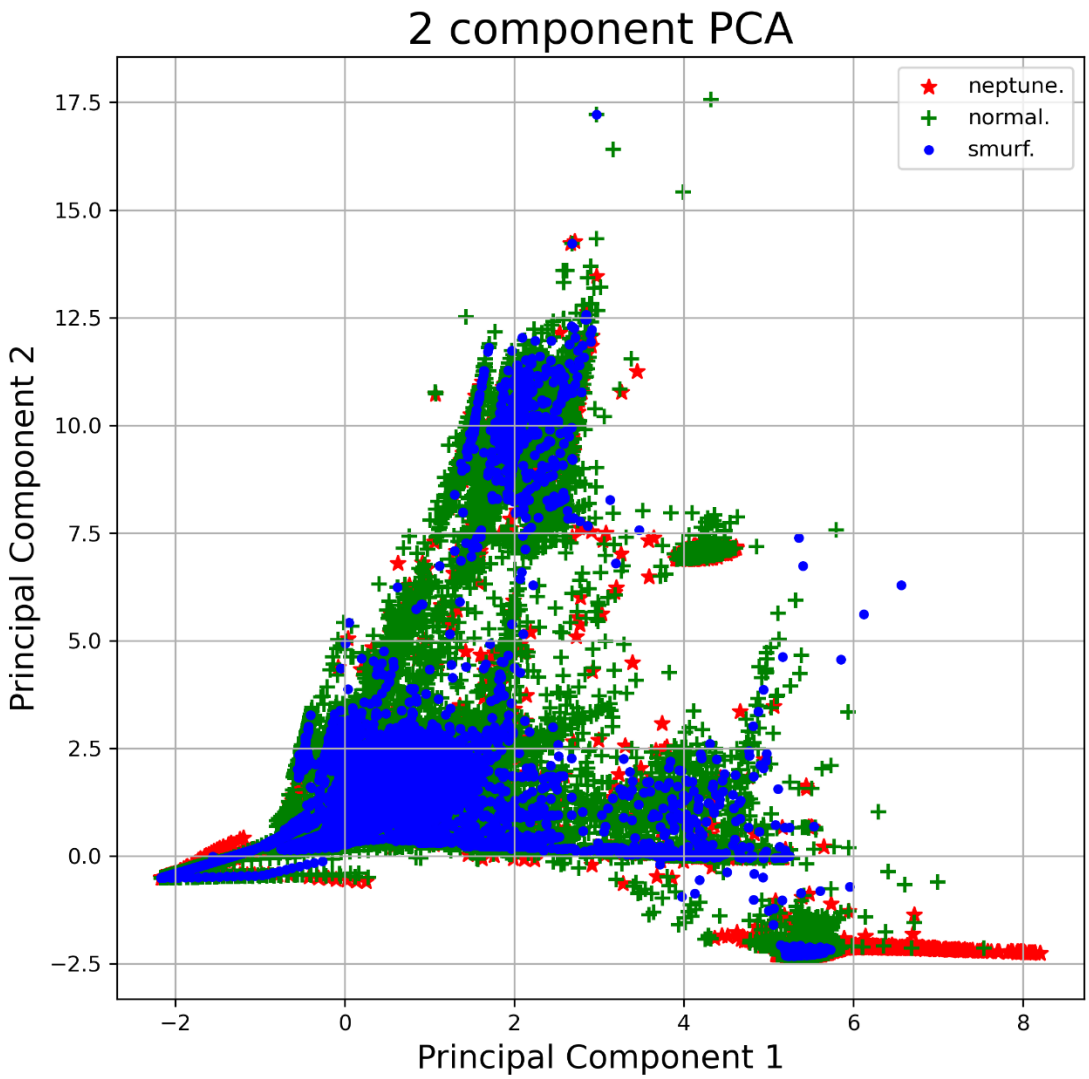


Figure 6.6 Principal Component Analysis.

6.6.1.2 Evaluation metrics

The end-to-end deep CNN-LSTM architecture uses accuracy, precision, recall, and F1-score to assess the performance of the classifier model. In this work, to quantify the performance of the proposed detection method, some performance metrics have been considered, such as accuracy, precision, recall, and F-1 score (defined below) from the confusion matrix. The confusion matrix generally reflects how efficiently a particular machine/algorithm classifies the actual data. It is the most ubiquitous matrix for the performance

evaluation of the classifier, which is discussed in chapter 2.6.

6.7 Results and discussions

The experiments have been carried out in Intel® Core ™ i7 2.6 GHz CPU with 16 GB RAM computer. The Anaconda Navigator 2.0.4 hosts JupyterLab 3.0.14, where algorithms are written in Python 3.8.8 of notebook 6.3.0. Our end-to-end deep CNN-LSTM model architecture clearly explains the number and size of convolution filters, kernel size, numbers and size of pooling layers, batch normalization layers, fully connected LSTM layer, and output layer, as in section 6.6. For the comparison purpose, along with the CNN-LSTM, we created Deep Neural Network (DNN), Convolution Neural Network (CNN), and LSTM. The models' performance metrics are evaluated under similar constraints, such as the same train and test data, hyperparameters, batch size, and so on. Apart from that, all our deep learning models run for 10 epochs taking a batch of size 500 while training and all the models are tested with a batch size of 20. 30 % of the train data has been held out for validation so that one can tune the hyperparameters. The trained model has never experienced the features from the test data during the training, so the models don't over-parameterize and memorize.

Fig. 6.7. presents the progression of training and validation accuracies and training and validation losses along the epochs. The training was so smooth that within 680 steps of the first epoch, our proposed model achieved successive training and validation accuracies of 91.80% and 99.98% with successive losses of 0.3231 and 0.0052. The average training time per epoch was 95.1 seconds. The best model from the training has achieved 99.99% accuracy on validation data. This model can be trained up to two epochs to get more than 99% training and validation accuracies.

Table 6.2 compares the precision, recall, F1-score, AUC, and testing accuracy of different

deep learning algorithms against the proposed CNN-LSTM. The proposed CNN-LSTM model has achieved the highest precision, recall, F1-score, AUC, and testing accuracy, i.e., more than 99% in each metric among the class of other implemented algorithms. 1D CNN algorithm has achieved almost similar AUC and accuracy as the proposed CNN-LSTM algorithm. For similar setups with two hidden layers and 64 hidden nodes, LSTM is the most inferior in terms of all other performance metrics except the AUC. All the performance metrics for DNN, CNN, and CNN-LSTM were found to perform excellently with more than 99% evaluation metrics. This justifies the superior performance of our proposed deep CNN-LSTM algorithm. Similarly, Table 6.3 presents the proposed model's classwise performance, where the model exhibits an almost 100% precision score for the samples from all three classes. The resulting recall and f1-score is almost 100% for Smurf and Neptune, with 99% for samples from a normal class.

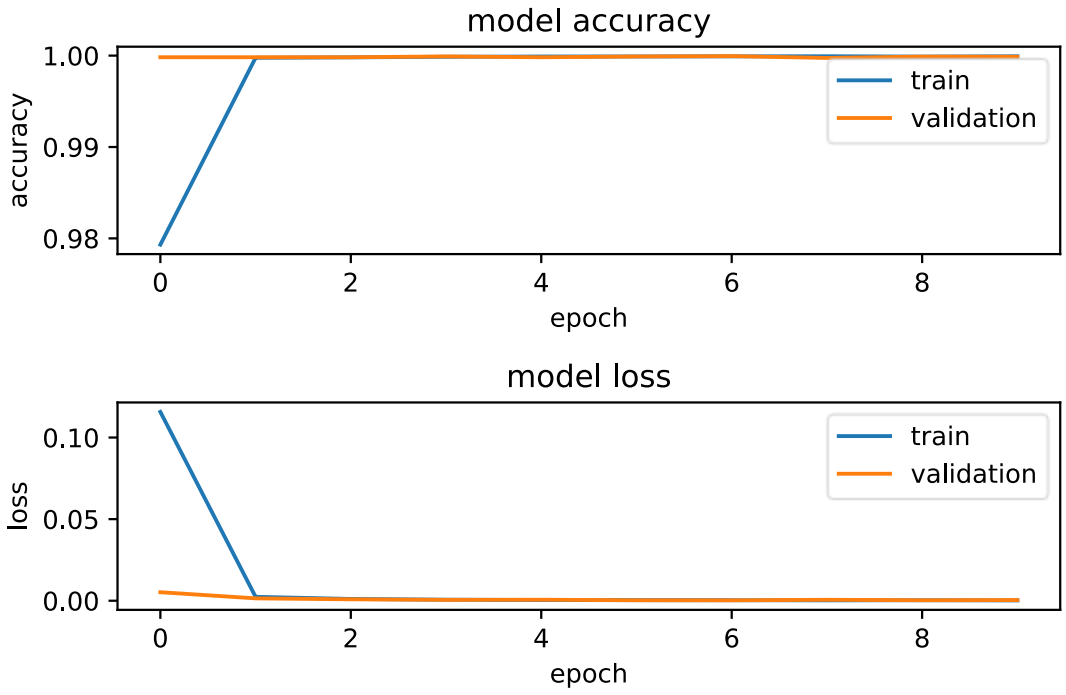


Figure 6.7 Training and Validation progression of deep CNN-LSTM.

Table 6.2 Performance metrics of DCNN-LSTM

Algorithms	Precision	Recall	F1-score	AUC	Accuracy
DNN	99.70%	99.50%	99.60%	99.82%	99.64%
CNN	99.84%	99.62%	99.73%	99.99%	99.75%
LSTM	92.41%	96.29%	93.85%	99.95%	93.89%
CNN-LSTM	99.85%	99.73%	99.74%	99.99%	99.75%

Table 6.3 Classwise performance evaluation of the proposed DCNN-LSTM model

Label	Precision	Recall	F1-score	Support
Smurf	1.00	1.00	1.00	164091
Normal	1.00	0.99	0.99	60590
Neptune	1.00	1.00	1.00	58001

Fig. 6.8 presents the multiclass confusion matrix where each block has the number of samples with a percentage belonging to that block. The last row indicates the actual samples belonging to that class, while the last column represents the predicted samples using the proposed algorithm. The numbers and percentages in red are misclassified samples. The highest misclassification rate of the proposed is 0.23% for the Smurf attack, which is 640 out of 164,091 samples. This implicit misclassification, i.e., bias, came from the data distribution because smurf got almost 58.04% of total samples for the testing model got similar bias because of similar data distribution in training. This bias in classifying Smurf resulted in a false alarm of normal class, i.e., there is a Smurf attack, but the model will predict it as a normal event. However, this error is less than 0.23% which is very small. But for high-sensitivity CAV attack detection, upsampling and downsampling can help to get equal data distribution while training the model. Overall, the proposed model has outstanding performance metrics, almost close to 100%.

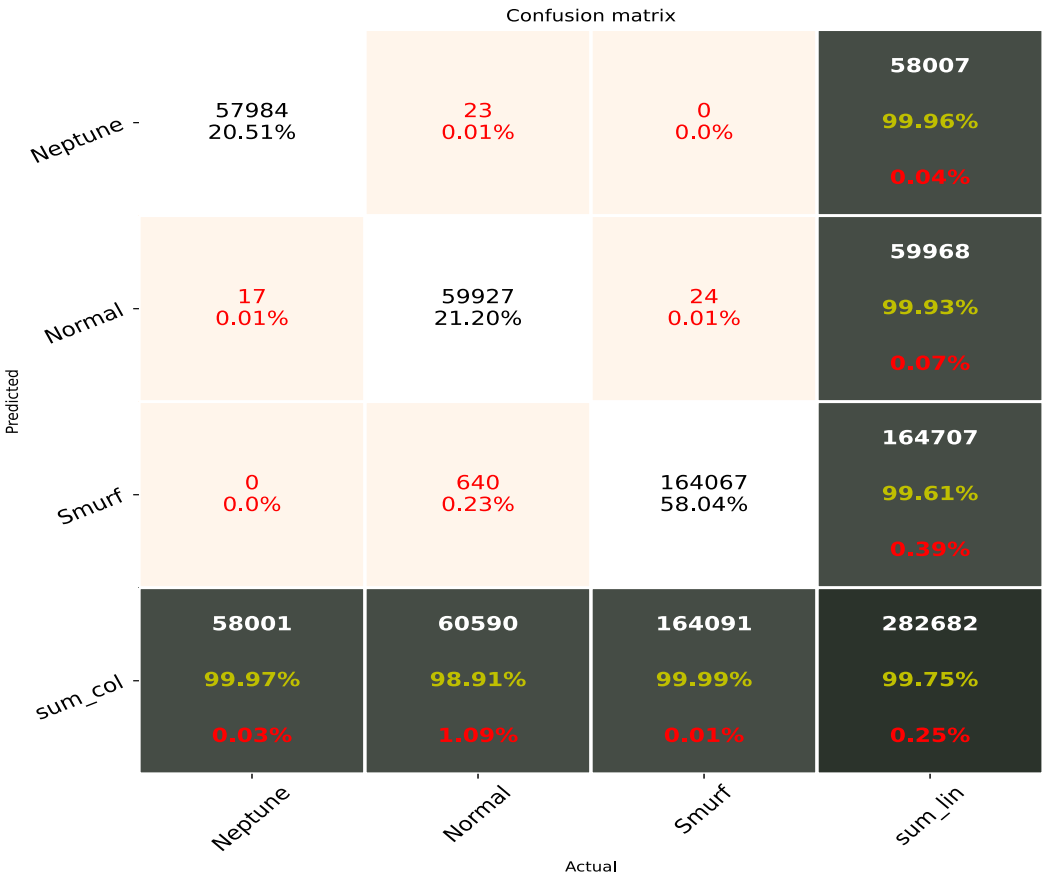


Figure 6.8 Confusion Matrix of deep CNN-LSTM

6.8 Chapter Conclusion

Along with the luxury of automation and connectivity, CAV inherits most of the cyber-physical vulnerabilities of incumbent technologies that primarily include evolving network architectures, wireless communications, and AI-assisted automation. This chapter sheds light on cyber-physical vulnerabilities and risks that originated in IT, OT, and the physical domains of the CAV ecosystem, eclectic threat landscapes, and threat intelligence. To deal with the security threats embedded in high-speed, high dimensional, multimodal data and assets of eccentric stakeholders of the CAV ecosystem, this chapter presents and analyzes some of the state-of-the-art deep learning-based threat intelligence for attack detection. Since deep learning has been evolving to attain superior cognition and intelligence, it would also directly impact threat

intelligence. The collaborative learning platform of deep learning, federated learning, can share threat intelligence without sharing the data between divergent stakeholders of the CAV ecosystem. Also, deep learning for CAV has still to work on Meta-learning for robust and swift generalization under the dynamic environment and out of distribution data context. The frontiers in deep learning and the challenges have been included in the chapter. We have proposed, trained, and tested the deep CNN-LSTM model for CAV threat intelligence; we assessed and compared the proposed model's performance against other deep learning algorithms such as DNN, CNN, and LSTM. Our results indicate the superiority of the proposed model, although DNN and 1d-CNN also achieved more than 99% of accuracy, precision, recall, f1-score, and AUC on the CAV-KDD dataset. The performance of deep CNN-LSTM comes with increased model complexity and cumbersome hyperparameter tuning. Still, there are open challenges to deep learning adoption in the CAV cybersecurity paradigm due to costlier implementations and training, lack of properly developed protocols and policies, poorly defined privileges between stakeholders, adversarial threats to the deep learning model, and poor generalizability of the model under out of data distributions.

The next chapter briefly studies 5G capabilities, Slicing, and their potential application to the internet of EVs and EVCSs.

Chapter 7 Analysis of 5G Slicing Approach to Electric Vehicle Charging Station

7.1 Introduction

The intelligent and extensive deployment of EVCS forces heterogeneous stakeholders to coordinate and communicate. The heterogeneous stakeholders mainly refer to i) the intelligent transportation system (ITS) infrastructures such as roadside sensors, connected automated vehicles (CAV), and EVs, ii) Electric grid infrastructures such as utility, generation, transmission, distribution, sensors, protection and relays so on, and iii) financial institution such as credit card companies for the management of transactions. The extent of administrative privilege for coordinating these eccentric stakeholders to/from the EVCS is still a conflict of interest due to the lack of clearly developed standards for proper interoperability and a fully matured, trustworthy environment. Therefore, the EVCS needs robust, secure, and reliable communication with its stakeholders. In such a scenario, the communication between these multiple nodes may need stringent requirements in terms of latency, bandwidth, and the number of connections. The enabling technology capable of providing ultra-reliable low-latency communication (uRLLC), extended mobile broadband (eMBB), and massive machine-type communication (mMTC) is 5G. This chapter attempts to explore the implementation of 5G slicing on EVCS.

7.2 Benefits 5G Slicing

5G slicing is a tenant-oriented virtual network that acts as a network as a service (NaaS) to handle specific service requirements, meets differentiated service level agreements (SLAs), and automatically builds isolated network instances on demand. It provides secure service isolation, end-to-end assurance for SLAs, customizable on-demand network function, and automation.

- End users can have guaranteed SLAs.

- Tenants can reduce the cost by network resource sharing, isolation, on-demand deployment, on-demand function customization
- Operators have maximized the use of the network supporting vertical industries and fast service rollout.

New Energy: The growing demand for energy-constrained to zero carbon emission involves random and intermittent power generation, which challenges the grids' operation and management.

New user: There are EVs, and EVCSs has the potential to change power consumption dynamics. The massive integration of EVs to the grid needs efficient control and smart management communication.

New requirements: Some high-tech digital devices require zero interruption and better asset utilization efficiency.

7.3 Key Performance Index (KPI) requirements for EVCS

5G slicing for smart grid-enabled EV ecosystem needs uRLLC, high reliability, isolation, and a huge number of connections, as shown in Fig. 7.1 The latency of 4G is 40 ms, and all services running on the same network, resulting in no isolation, same network function for all services do not guarantee the SLAs) [157]. 5G slicing has the following potential.

- Diversification of services: new user, new energy, and new requirements
- Security Isolation
- High-performance requirements
- Millions of nodes

5G offers 1ms latency and 10 million connections per square km, perfect for CAV and EV

charging services. It provides security and isolation as much as costly fiber optics. 5G MEC offers local traffic processing and logical computing that saves BW and latency.

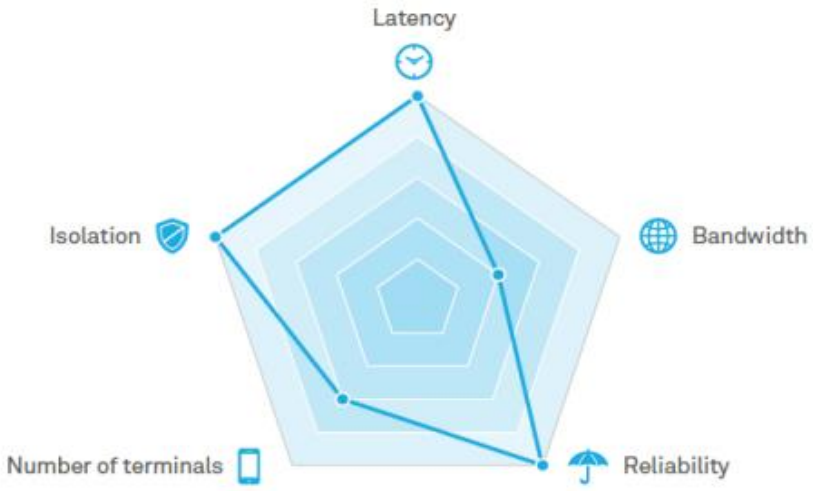


Figure 7.1 key performance indicator (KPI)requirement for low latency EVCS

7.4 5G Slicing model for EV infrastructure

This chapter deals with the potential implementation of 5G slicing for EV infrastructure. Paper [158] implemented LSTM-based EV charging prediction using 26,000 charging records of 318 users collected over a year in Los Angels in a 5G smart grid based on network slicing and edge computing concept. However, the paper lacks the implementation of 5G for the work, although they provided the theoretical background. An author in [159] filed a patent by adding a wireless edge computing module and 5G cellular antenna module in EVCS. The edge computing module connects to an IP network for a cellular network's mobile/internet gateway, and the cellular base station connects to the antenna system, as shown in Fig. 7.2.

Unlike its predecessor, 5G has some key enabling techniques such as edge computing,

network function virtualization, and network slicing. 5G network slicing is designed to handle specific service requirements, meets differentiated service level agreements (SLA), and automatically builds the isolated network instance on demand. 5G network slicing provides end-to-end network assurance for SLA, service isolation, customizable on-demand network function, and automation. It enables communication service operators to allocate the network resources dynamically and provide the network as a service [160]. The orchestrated network for the EVCS slice should have ultra-low latency (<1 ms), very high reliability, and very high isolation, as shown in Fig. 7.2.

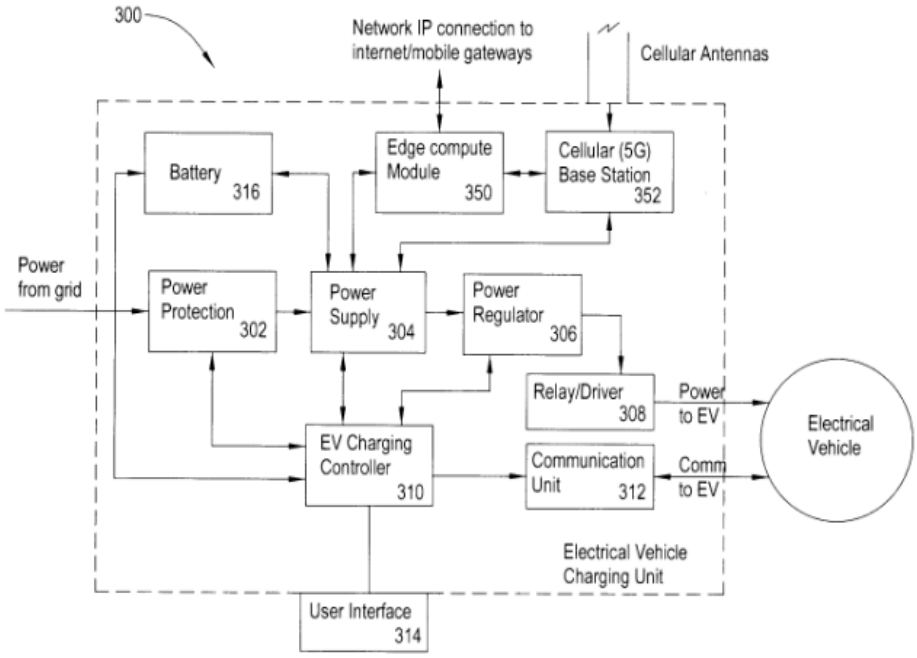


Figure 7.2 5G-enabled EVCS architecture

7.5 5G network Slicing Architecture

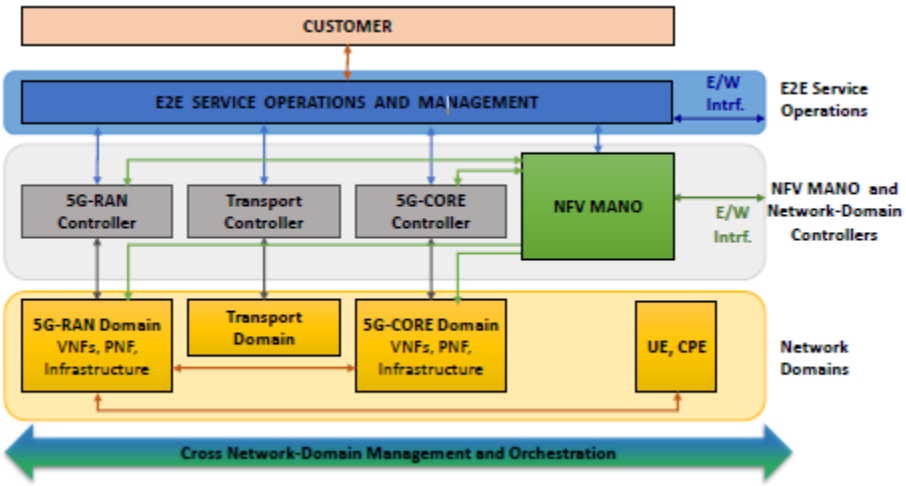


Figure 7.3 Fig. 5G reference Architecture [161]

Fig.7.3 presents the end-to-end architecture spanning all network domains, RAN, CN, and transport network. RAN and CN domains provide the Network Slice Subnet Instances (NSSIs) that can be combined to define the NSIs. TN provides virtual links between the components building the NSIs. In the middle layer, there's NFV-Management and Orchestration and the controllers for each domain. MANO focuses on virtualization-specific tasks, while the domain controller focuses on non-virtualization-related operations. MANO is responsible for managing VNFs. RAN and Core domain controllers manage different NFs at the application level and control all the non-virtualized tasks. The transport controllers have SDN controllers. The end-to-end service operations and management level coordinates and controls all the domain controllers and network services to harmonize RAN, Core, and transport.

7.6 Isolation and Security of NSIs

The SLA's key performance indicators (KPIs) must always be met in a particular NSI regardless of congestion and load surge in the remaining NSIs [161]. The isolation concept of NSI

must ensure security to the NSI regardless of breaches and malfunctions in other NSIs. That implies there should be no unauthorized read or write access to the NSI-specific configurations. Also, faults or attacks in an NSI must be confined and should not propagate to other NSIs. The isolation guarantee can be met in two ways: the first is to partition the resources at the infrastructure level. The second way is to introduce isolation among NSIs at the management level by employing policy-based orchestration or multi-tenancy support defined in the participant management block. There are three ways of achieving isolation in NSIs: the first is physical isolation with all the physical resources given to the tenant, and it is the best way of isolation. The second way is physical resource splitting, e.g., frequency band split into sub-carriers. The third is logical isolation, which can be achieved by logical-capacity delimitation, logical -prioritization, and simple logical isolation. The problem with the splitting is it has access to all the shared resources, which can be the prime security threat. The tenant may demand isolation in the different levels of the radio domain, such as radio access technologies (RAT), antennas, frequency, or carrier bands. Fig. 7.4 shows the isolation concept in 5G slices.

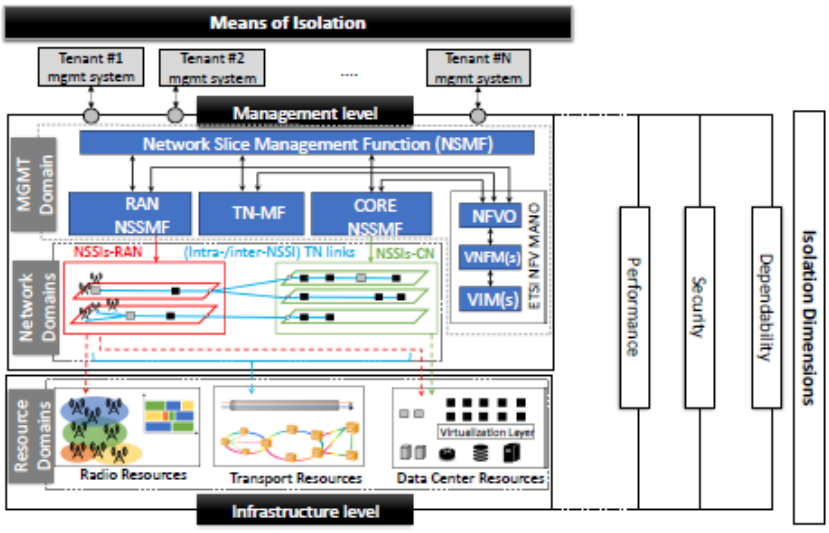


Figure 7.4 Isolation dimension in Network Slicing [161]

7.7 Proposed 5G slicing architecture for EVCS

Based on the above background, we propose the 5G slicing architecture for holistic EVCS with aided isolation as shown in Fig. 7.5. Each 5G slice can be implemented horizontally for the layered architecture; e.g., the SCADA, utilities governing power network, EVCSs, and CAEV each can have a slice as shown in fig below. Also, the detection and mitigation framework (DMF) can be included in the slice. The vertical implementation of 5G slices is not recommended since SCADA, Power network, EVCS, and CAEV have heterogeneous network resource requirements and KPIs. Another potential vertical implementation issue is the privilege conflict to slice management and operation between different service providers in the ecosystem. The proposed architecture can share the threat information with the help of DMF and DMF management and synchronization.

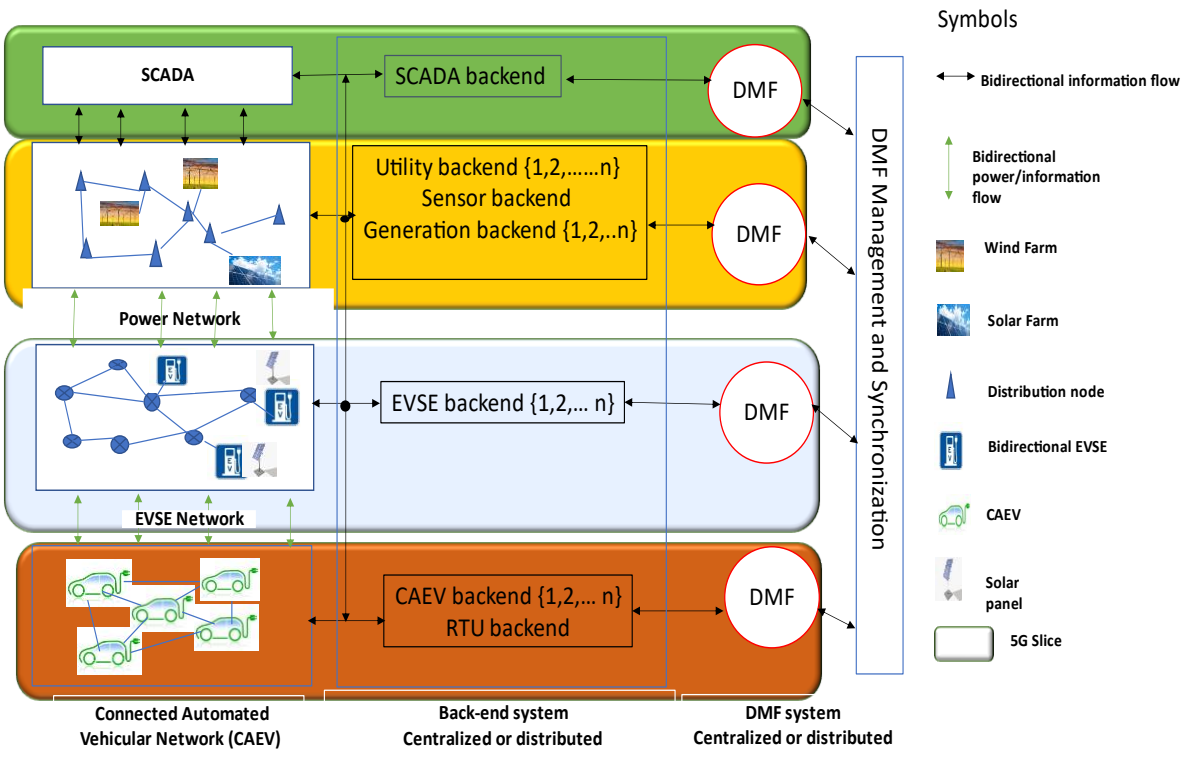


Figure 7.5 5G slicing implementation for secure EVCS.

7.8 Discussions

The reference [162] proposed the idea of slice leaders for clusters of Evs using K-Means clustering to tackle the problem of network traffic congestion caused by multiple vehicles sending requests to EVCS through the roadside unit (RSUs). Instead of sending the request from each vehicle in the slice, the leader will communicate with EVCS and then relay that information to the clustered vehicles. It has improved the throughput and data rates with better resource utilization than the first come, first serve algorithm. However, unsupervised clustering may not produce accurate results for the vast platoons of EVs. Also, the number of slices exponentially increases as there might be highly dynamic clusters, and allocating numerous slices might stress the 5G resources. As far as the author's knowledge, there has been no real implementation of 5G slicing for EV infrastructures. Ref [158] hypothesized the generic smart grid architecture based on 5G network slicing but missed important details about the slicing architecture and technical specifications. Ref. [163] proposed the secure blockchain-based 5G mobility framework to enhance the security of IoEV data but still lacks the implementation details of 5G. On a similar note, ref. [164] analyzes the KPI of the power grid, and the high-level architecture of 5G slicing has been presented. Ref. [165] highlights the security threats and recommendations in 5G network slicing in terms of intra-slice security, inter-slice security, and life cycle security. Based on the above discussion, there has been no research so far with actual 5G slice implementation in smart grids, seldom the EV infrastructure. However, there's growing interest as the technology becoming more matured. As a result, most of the researcher came up with their own network architecture that basically lacks the technical details about the slicing and therefore implementation can surely face challenges.

7.9 Chapter Conclusion

Ideally, 5G slices are logically isolated independent NSIs with the capability to confine the faults and attacks within the NSIs. This feature could safeguard the communication between critical controllers and SCADA/EMS in the EVCS. Also, the EVs and EMS communication for charge schedules, authentication, and authorization.

The next chapter summarizes our conclusion, contributions, and future works.

Chapter 8 Conclusion, Contributions, and Future Work

8.1 Conclusion and Contributions

This dissertation presented several studies conducted in deep learning-powered computational intelligence to detect and mitigate the adverse impacts of cyberattacks on 5G-enabled EVCS. In terms of the field's level or type of research, one study focused on devising a novel double-layered cyberattack detection framework capable of detecting attacks in the network and physical layer in the 5G-enabled EVCS infrastructure and the rest concentrated on designing mitigation and defense of the cyberattacks. The research approach was based on analyzing the existing methods, identifying the gaps or drawbacks, and finding the right solutions. The methodologies were validated through computer simulations. The following findings are briefly concluded:

- This research proposed, designed, and tested a NIDS for EVCS employing DNN and LSTM that could detect and categorize stealthy DDoS attacks on the EVCS network with nearly 99 % performance metrics.

- The performance of the proposed NIDS is further improved by implementing a semi-supervised generative model called WC-GAN-based external classifier that improves the detection performance by resolving the problem of a low sample class, i.e., class imbalance.

- The resultant impact on EVCS operation caused by the DDoS attacks and FDI attacks on 5G core infrastructure has been studied and found to be detrimental to the physical controllers and the controlled components of EVCS.

- The attacks make the EVCS system oscillate or shift the DC operating point. The frequency of oscillation, damping, and the system's resiliency is related to the attacks' intensity and the target

controller.

- A novel, air-gapped stacked LSTM-based HIDS has been proposed, designed, and tested that can detect bypassed cyberattacks on the physical controllers of EVCS with nearly 100% performance metrics.

- A novel ransomware detection framework employing DNN, CNN, and LSTM has been proposed, designed, and tested for the smart grid environment. The performance of different DL algorithms is compared and recommended for deployment.

- Also, the dissertation proposed and tested the deep CNN-LSTM architecture for CAV threat intelligence, assessed and compared the performance of the proposed models with other DL models, and found a superior performance.

- The dissertation introduced, designed, and tested a novel, data-driven clone-based cyberattack mitigation and defense in critical EVCS controllers. The proposed TD3-based software clones are capable of taking over the legacy controllers under APT attacks or even under anomalous behavior. The TD3-based clones are superior to DDPG-based clones in terms of convergence, stability, hyperparameter sensitivity, and mitigation actions.

- The dissertation proposed and tested the generic Bruteforce mitigation and controller clone-based mitigation approaches to deal with the APT attacks at standalone EVCS. The performance measures indicate effective mitigation results by both the proposed models. The results also indicate the superiority of Controller clone-based mitigation in terms of adaptation to system dynamics compared to the less agile but simple Bruteforce method.

- The potential application of 5G slicing to enhance the security and isolation of smart grid environments with Power delivery, EVs, CAEVs, EVCS, and SCADA/EMS networks has been

envisioned, and a novel multi-slice architecture has been proposed. 5G slices are logically isolated independent NSIs with the capability to confine the faults and attacks within the NSIs. This feature could safeguard the communication between critical controllers and SCADA/EMS in the EVCS. Also, the EVs and EMS communication for charge schedules, authentication, and authorization.

It is to note here that all models in this study have been simulated in Python, MATLAB/Simulink, and NetSim environments. The obtained results closely match realistic situations, as the software developer has numerically validated the accuracy of basic model blocks and function blocks found in the software library and toolboxes.

8.2 Future Work

The researcher is interested in pursuing this research or recommends extending this research in the following directions:

- The EVCS integration into the grids can bring many opportunities, such as bidirectional energy transactions from the grid to EVs and vice-versa through EVCS. It will enable consumers to be prosumers and adopt renewable energy sources such as PV, Wind, etc.

- Current industry practices for EVCS, such as AC charging to extreme DC fast charging, and its architecture and efficiency could be enhanced by improving the power, control, and communication circuitries.

- Dynamic wireless charging on the roads can reduce the wait queues and congestion at EVCS and stress on the grids.

- The periodic security assessments and hardening of the current standards and protocols for EVCS, such as IEC 15118, IEC 61851-1, IEEE 2030.5, OCPP, OCHP, OCPI, OSCP, CCS, is

worth exploring.

- The 5G-enabled proposed EVCS can be integrated and tested under the various DERs and power grids to seamlessly adopt EVs. The prediction, detection, and mitigation of cyber-physical attacks propagated to the grids and other DERs should be explored.

- Blockchain can be adopted for more democratized and secure energy transactions between EVCS, grid, EVs, and DERs.

- Actual cyberattack data for EVCS has not been shared in the research community because of the trust, reputation, and fear of further exploitation. Academia and industry should unite to share threat intelligence and tighten security standards.

- The 5G-enabled EVCS prototype in a Simulink environment can be built, and the proposed algorithms for detection and mitigation can be validated and tested on more realistic and sophisticated environments.

- Deep learning-powered detection algorithms in NIDS and HIDS of EVCS can be extended to detect diverse attacks as they can be trained with the new attack types. The highly parameterized DL algorithms should be optimized and tested before deployment for the best performance.

- The future research direction would be developing more efficient and sophisticated algorithms for detecting and mitigating cyberattacks in EV infrastructure.

- The quest for developing and testing lightweight and more visible algorithms with heuristics and rules should always be open along with the forays for developing high-performance DL architecture.

- If there were enough data and testbeds, the DL-powered computational intelligence for

cyberattack detection in EVCS should have been at its peak. However, cyberattack mitigation and defense have a long way to go.

- Cyber-physical threats in 5G-enabled EVCS are the continuously evolving paradigm as it migrates the inherent vulnerabilities of incumbent technologies such as NGN, DL-based computational intelligence, Control, and Optimization. Therefore, security by design and defense in depth should also evolve to tackle the problems.

Bibliography

[1] "Alternative Fuels Data Center: Electric Vehicle Charging Infrastructure Trends." https://afdc.energy.gov/fuels/electricity_infrastructure_trends.html (accessed Feb. 22, 2022).

[2] "FACT SHEET: Biden Administration Advances Electric Vehicle Charging Infrastructure," *The White House*, Apr. 22, 2021. https://www.whitehouse.gov/briefing-room/statements-releases/2021/04/22/fact-sheet-biden-administration-advances-electric-vehicle-charging-infrastructure/ (accessed Feb. 22, 2022).

[3] "President Biden, USDOT and USDOE Announce $5 Billion over Five Years for National EV Charging Network, Made Possible by Bipartisan Infrastructure Law | FHWA." https://highways.dot.gov/newsroom/president-biden-usdot-and-usdoe-announce-5-billion-over-five-years-national-ev-charging (accessed Feb. 22, 2022).

[4] S. Acharya, Y. Dvorkin, H. Pandžić, and R. Karri, "Cybersecurity of Smart Electric Vehicle Charging: A Power Grid Perspective," *IEEE Access*, vol. 8, pp. 214434–214453, 2020, doi: 10.1109/ACCESS.2020.3041074.

[5] B. Anderson and J. Johnson, *Securing Vehicle Charging Infrastructure Against Cybersecurity Threats*. 2020. doi: 10.13140/RG.2.2.28243.12329.

[6] Y. Park, O. C. Onar, and B. Ozpineci, "Potential Cybersecurity Issues of Fast Charging Stations with Quantitative Severity Analysis," in *2019 IEEE CyberPELS (CyberPELS)*, Apr. 2019, pp. 1–7. doi: 10.1109/CyberPELS.2019.8925069.

[7] M. Basnet and M. H. Ali, "Deep Learning-based Intrusion Detection System for Electric Vehicle Charging Station," in *2020 2nd International Conference on Smart Power Internet Energy Systems (SPIES)*, Sep. 2020, pp. 408–413. doi: 10.1109/SPIES48661.2020.9243152.

[8] J. Johnson, *DER Cybersecurity Stakeholder Engagement, Standards Development, and EV Charger Penetration Testing*. 2021.

[9] "UTAustin (2021) EventsFebruary2021TexasBlackout 20210714.pdf." Accessed: Feb. 22, 2022. [Online]. Available: https://energy.utexas.edu/sites/default/files/UTAustin%20%282021%29%20EventsFebruary2021TexasBlackout%2020210714.pdf

[10] "Hackers Breached Colonial Pipeline Using Compromised Password," *Bloomberg.com*, Jun. 04, 2021. Accessed: Feb. 22, 2022. [Online]. Available: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password

[11] C. Osborne, "Colonial Pipeline attack: Everything you need to know," *ZDNet*. https://www.zdnet.com/article/colonial-pipeline-ransomware-attack-everything-you-need-to-know/ (accessed Feb. 22, 2022).

[12] "JBS Ransomware Attack Started in March," *SecurityScorecard*. https://securityscorecard.com/blog/jbs-ransomware-attack-started-in-march (accessed Feb. 22, 2022).

[13] D. F. on February 26 and 2019, "Oil & Gas Cybersecurity and Process Safety Converge Thanks to TRITON," *Security Boulevard*, Feb. 26, 2019. https://securityboulevard.com/2019/02/oil-gas-cybersecurity-and-process-safety-converge-thanks-to-triton/ (accessed Feb. 22, 2022).

[14] "Attackers Deploy New ICS Attack Framework 'TRITON' and Cause Operational Disruption to Critical Infrastructure | Mandiant." https://www.mandiant.com/resources/attackers-deploy-new-ics-attack-framework-triton (accessed Feb. 22, 2022).

[15] "Triton Malware Spearheads Latest Attacks on Industrial Systems," *McAfee Blog*, Nov. 08, 2018. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/triton-malware-spearheads-latest-generation-of-attacks-on-industrial-systems/ (accessed Feb. 22, 2022).

[16] "SANS Industrial Control Systems Security Blog | Confirmation of a Coordinated Attack on the Ukrainian Power Grid | SANS Institute." https://www.sans.org/blog/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid/ (accessed Feb. 22, 2022).

[17] K. Zetter, “Inside the Cunning, Unprecedented Hack of Ukraine’s Power Grid,” *Wired*. Accessed: Feb. 23, 2022. [Online]. Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
[18] “The Real Story of Stuxnet,” *IEEE Spectrum*, Feb. 26, 2013. https://spectrum.ieee.org/the-real-story-of-stuxnet (accessed Feb. 23, 2022).
[19] G. Wyss, P. Sholander, J. Darby, and J. Phelan, “Identifying and Defeating Blended Cyber-Physical Security Threats,” p. 6.
[20] R. Gottumukkala, R. Merchant, A. Tauzin, K. Leon, A. Roche, and P. Darby, “Cyber-physical System Security of Vehicle Charging Stations,” in *2019 IEEE Green Technologies Conference(GreenTech)*, Lafayette, LA, USA, Apr. 2019, pp. 1–5. doi: 10.1109/GreenTech.2019.8767141.
[21] I. S. Bayram and I. Papapanagiotou, “A survey on communication technologies and requirements for internet of electric vehicles,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2014, no. 1, p. 223, Dec. 2014, doi: 10.1186/1687-1499-2014-223.
[22] K. Harnett, B. Harris, D. Chin, and G. Watson, “DOE/DHS/DOT Volpe Technical Meeting on Electric Vehicle and Charging Station Cybersecurity Report,” p. 44.
[23] A. Huseinovic, S. Mrdovic, K. Bicakci, and S. Uludag, “A Taxonomy of the Emerging Denial-of-Service Attacks in the Smart Grid and Countermeasures,” in *2018 26th Telecommunications Forum (TELFOR)*, Belgrade, Nov. 2018, pp. 1–4. doi: 10.1109/TELFOR.2018.8611847.
[24] “Securing Vehicle Charging Infrastructure APR,” *Cyber Secur.*, p. 6, 2019.
[25] S. Vitturi, C. Zunino, and T. Sauter, “Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G,” *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, Jun. 2019, doi: 10.1109/JPROC.2019.2913443.
[26] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, “5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View,” *IEEE Access*, vol. 6, pp. 55765–55779, 2018, doi: 10.1109/ACCESS.2018.2872781.
[27] G. Naik, B. Choudhury, and J. Park, “IEEE 802.11bd 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications,” *IEEE Access*, vol. 7, pp. 70169–70184, 2019, doi: 10.1109/ACCESS.2019.2919489.
[28] M. Polese *et al.*, “Integrated Access and Backhaul in 5G mmWave Networks: Potential and Challenges,” *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 62–68, Mar. 2020, doi: 10.1109/MCOM.001.1900346.
[29] S. Lagen *et al.*, “New Radio Beam-Based Access to Unlicensed Spectrum: Design Challenges and Solutions,” *IEEE Commun. Surv. Tutor.*, vol. 22, no. 1, pp. 8–37, Firstquarter 2020, doi: 10.1109/COMST.2019.2949145.
[30] “Level 1 and Level 2 Electric Vehicle Service Equipment (EVSE) Reference Design,” p. 36, 2016.
[31] S. Hu, X. Chen, W. Ni, X. Wang, and E. Hossain, “Modeling and Analysis of Energy Harvesting and Smart Grid-Powered Wireless Communication Networks: A Contemporary Survey,” *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 461–496, Jun. 2020, doi: 10.1109/TGCN.2020.2988270.
[32] R. Heartfield, G. Loukas, and D. Gan, “You Are Probably Not the Weakest Link: Towards Practical Prediction of Susceptibility to Semantic Social Engineering Attacks,” *IEEE Access*, vol. 4, pp. 6910–6928, 2016, doi: 10.1109/ACCESS.2016.2616285.
[33] M. Basnet and M. H. Ali, “Deep Learning-based Intrusion Detection System for Electric Vehicle Charging Station,” in *2020 2nd International Conference on Smart Power Internet Energy Systems (SPIES)*, Sep. 2020, pp. 408–413. doi: 10.1109/SPIES48661.2020.9243152.
[34] S. Mousavian, M. Erol-Kantarci, L. Wu, and T. Ortmeyer, “A Risk-Based Optimization Model for Electric Vehicle Infrastructure Response to Cyber Attacks,” *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6160–6169, Nov. 2018, doi: 10.1109/TSG.2017.2705188.
[35] D. Reeh, F. C. Tapia, Y. Chung, B. Khaki, C. Chu, and R. Gadh, “Vulnerability Analysis and Risk Assessment of EV Charging System under Cyber-Physical Threats,” in *2019 IEEE Transportation*

Electrification Conference and Expo (ITEC), Jun. 2019, pp. 1–6. doi: 10.1109/ITEC.2019.8790593.
[36] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/j.jnca.2012.09.004.
[37] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.
[38] D. Niyato, D. T. Hoang, P. Wang, and Z. Han, "Cyber Insurance for Plug-In Electric Vehicle Charging in Vehicle-to-Grid Systems," *IEEE Netw.*, vol. 31, no. 2, pp. 38–46, Mar. 2017, doi: 10.1109/MNET.2017.1600321NM.
[39] J. Antoun, M. E. Kabir, B. Moussa, R. Atallah, and C. Assi, "A Detailed Security Assessment of the EV Charging Ecosystem," *IEEE Netw.*, vol. 34, no. 3, pp. 200–207, May 2020, doi: 10.1109/MNET.001.1900348.
[40] M. Basnet and Mohd. H. Ali, "Exploring cybersecurity issues in 5G enabled electric vehicle charging station with deep learning," *IET Gener. Transm. Distrib.*, p. gtd2.12275, Aug. 2021, doi: 10.1049/gtd2.12275.
[41] M. Basnet, S. Poudyal, Mohd. H. Ali, and D. Dasgupta, "Ransomware Detection Using Deep Learning in the SCADA System of Electric Vehicle Charging Station," in *2021 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, Sep. 2021, pp. 1–5. doi: 10.1109/ISGTLatinAmerica52371.2021.9543031.
[42] E. Gumrukcu *et al.*, "Impact of Cyber-attacks on EV Charging Coordination: The Case of Single Point of Failure," in *2022 4th Global Power, Energy and Communication Conference (GPECOM)*, Jun. 2022, pp. 506–511. doi: 10.1109/GPECOM55404.2022.9815727.
[43] T. Nasr, S. Torabi, E. Bou-Harb, C. Fachkha, and C. Assi, "Power jacking your station: In-depth security analysis of electric vehicle charging station management systems," *Comput. Secur.*, vol. 112, p. 102511, Jan. 2022, doi: 10.1016/j.cose.2021.102511.
[44] Y. Shen, W. Fang, F. Ye, and M. Kadoch, "EV Charging Behavior Analysis Using Hybrid Intelligence for 5G Smart Grid," *Electronics*, vol. 9, no. 1, Art. no. 1, Jan. 2020, doi: 10.3390/electronics9010080.
[45] S. Dey and M. Khanra, "Cybersecurity of Plug-In Electric Vehicles: Cyberattack Detection During Charging," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 478–487, Jan. 2021, doi: 10.1109/TIE.2020.2965497.
[46] M. Girdhar, J. Hong, H. Lee, and T. Song, "Hidden Markov Models based Anomaly Correlations for the Cyber-Physical Security of EV Charging Stations," *IEEE Trans. Smart Grid*, pp. 1–1, 2021, doi: 10.1109/TSG.2021.3122106.
[47] S. Acharya, R. Mieth, C. Konstantinou, R. Karri, and Y. Dvorkin, "Cyber Insurance Against Cyberattacks on Electric Vehicle Charging Stations," *IEEE Trans. Smart Grid*, vol. 13, no. 2, pp. 1529–1541, Mar. 2022, doi: 10.1109/TSG.2021.3133536.
[48] A. Sanghvi, T. Markel, S. Granda, A. Nagarajan, and M. Jun, "Identification and Testing of Electric Vehicle Fast Charger Cybersecurity Mitigations," National Renewable Energy Lab. (NREL), Golden, CO (United States), NREL/TP-5R00-80799, Nov. 2021. doi: 10.2172/1832208.
[49] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks - SASN '03*, Fairfax, Virginia, 2003, p. 125. doi: 10.1145/986858.986876.
[50] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.
[51] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi:

10.1016/j.jnca.2012.09.004.
[52] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, “Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems,” *IEEE Access*, vol. 7, pp. 46595–46620, 2019, doi: 10.1109/ACCESS.2019.2909807.
[53] S. M. Kasongo and Y. Sun, “A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System,” *IEEE Access*, vol. 7, pp. 38597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
[54] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System,” *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
[55] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, “MSML: A Novel Multilevel Semi-Supervised Machine Learning Framework for Intrusion Detection System,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1949–1959, Apr. 2019, doi: 10.1109/JIOT.2018.2873125.
[56] Q. Tian, J. Li, and H. Liu, “A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning,” *IEEE Access*, vol. 7, pp. 38688–38695, 2019, doi: 10.1109/ACCESS.2019.2905754.
[57] H. Yang, G. Qin, and L. Ye, “Combined Wireless Network Intrusion Detection Model Based on Deep Learning,” *IEEE Access*, vol. 7, pp. 82624–82632, 2019, doi: 10.1109/ACCESS.2019.2923814.
[58] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, “An Optimization Method for Intrusion Detection Classification Model Based on Deep Belief Network,” *IEEE Access*, vol. 7, pp. 87593–87605, 2019, doi: 10.1109/ACCESS.2019.2925828.
[59] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, “An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks,” *IEEE Access*, vol. 7, pp. 42210–42219, 2019, doi: 10.1109/ACCESS.2019.2904620.
[60] S. Park, M. Kim, and S. Lee, “Anomaly Detection for HTTP Using Convolutional Autoencoders,” *IEEE Access*, vol. 6, pp. 70884–70901, 2018, doi: 10.1109/ACCESS.2018.2881003.
[61] K. Wu, Z. Chen, and W. Li, “A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks,” *IEEE Access*, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.
[62] C. Yin, Y. Zhu, J. Fei, and X. He, “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
[63] M. Sheikhan, Z. Jadidi, and A. Farrokhi, “Intrusion detection using reduced-size RNN based on feature grouping,” *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012, doi: 10.1007/s00521-010-0487-0.
[64] M. Sheikhan, Z. Jadidi, and A. Farrokhi, “Intrusion detection using reduced-size RNN based on feature grouping,” *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012, doi: 10.1007/s00521-010-0487-0.
[65] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, “Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection,” in *2016 International Conference on Platform Technology and Service (PlatCon)*, Feb. 2016, pp. 1–5. doi: 10.1109/PlatCon.2016.7456805.
[66] “Numenta White Paper - Science of Anomaly Detection.pdf.”
[67] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, “A Distributed Anomaly Detection System for In-Vehicle Network Using HTM,” *IEEE Access*, vol. 6, pp. 9091–9098, 2018, doi: 10.1109/ACCESS.2018.2799210.
[68] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017, doi: 10.1016/j.neucom.2017.04.070.
[69] J. Johnson, T. Berg, B. Anderson, and B. Wright, “Review of Electric Vehicle Charger Cybersecurity Vulnerabilities, Potential Impacts, and Defenses,” *Energies*, vol. 15, no. 11, Art. no.

11, Jan. 2022, doi: 10.3390/en15113931.
[70] D. Said, M. Elloumi, and L. Khoukhi, “Cyber-Attack on P2P Energy Transaction Between Connected Electric Vehicles: A False Data Injection Detection Based Machine Learning Model,” *IEEE Access*, vol. 10, pp. 63640–63647, 2022, doi: 10.1109/ACCESS.2022.3182689.
[71] M. R. Habibi, H. R. Baghaee, T. Dragicevic, and F. Blaabjerg, “False Data Injection Cyber-Attacks Mitigation in Parallel DC/DC Converters Based on Artificial Neural Networks,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 68, no. 2, pp. 717–721, Feb. 2021, doi: 10.1109/TCSII.2020.3011324.
[72] C. Roberts *et al.*, “Deep Reinforcement Learning for DER Cyber-Attack Mitigation,” in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, Nov. 2020, pp. 1–7. doi: 10.1109/SmartGridComm47815.2020.9302997.
[73] H. A. Kholidy, “Autonomous mitigation of cyber risks in the Cyber–Physical Systems,” *Future Gener. Comput. Syst.*, vol. 115, pp. 171–187, Feb. 2021, doi: 10.1016/j.future.2020.09.002.
[74] S. J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P. K. Chan, “Cost-based modeling for fraud and intrusion detection: results from the JAM project,” in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX’00*, Hilton Head, SC, USA, 1999, vol. 2, pp. 130–144. doi: 10.1109/DISCEX.2000.821515.
[75] R. P. Lippmann *et al.*, “Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation,” in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX’00*, Jan. 2000, vol. 2, pp. 12–26 vol.2. doi: 10.1109/DISCEX.2000.821506.
[76] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.
[77] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.
[78] J. Song, H. Takakura, and Y. Okabe, “Description of Kyoto University Benchmark Data,” p. 3.
[79] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, “WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks,” *J. Sens.*, vol. 2016, pp. 1–16, 2016, doi: 10.1155/2016/4731953.
[80] “IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” https://www.unb.ca/cic/datasets/ids-2017.html (accessed Oct. 20, 2019).
[81] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, “5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15,” *IEEE Access*, vol. 7, pp. 127639–127651, 2019, doi: 10.1109/ACCESS.2019.2939938.
[82] “Updated ENISA 5G Threat Landscape Report to Enhance 5G Security.” https://www.enisa.europa.eu/news/enisa-news/updated-enisa-5g-threat-landscape-report-to-enhance-5g-security (accessed Dec. 22, 2020).
[83] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, “A Technique for Improving P O MPPT Performances of Double-Stage Grid-Connected Photovoltaic Systems,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4473–4482, Nov. 2009, doi: 10.1109/TIE.2009.2029589.
[84] O. Rabiaa, B. H. Mouna, S. Lassaad, F. Aymen, and A. Aicha, “Cascade Control Loop of DC-DC Boost Converter Using PI Controller,” in *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Nov. 2018, pp. 1–5. doi: 10.1109/ISAECT.2018.8618859.
[85] N. Femia, G. Petrone, G. Spagnuolo, and M. Vitelli, “Optimizing duty-cycle perturbation of P O MPPT technique,” in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, Jun. 2004, vol. 3, pp. 1939-1944 Vol.3. doi: 10.1109/PESC.2004.1355414.
[86] K. J. Sauer and T. Roessler, “Systematic approaches to ensure correct representation of measured

multi-irradiance module performance in PV system energy production forecasting software programs," in *2012 38th IEEE Photovoltaic Specialists Conference*, Jun. 2012, pp. 000703–000709. doi: 10.1109/PVSC.2012.6317706.
[87] V. Kumar, V. R. Teja, M. Singh, and S. Mishra, "PV Based Off-Grid Charging Station for Electric Vehicle," *IFAC-Pap.*, vol. 52, no. 4, pp. 276–281, Jan. 2019, doi: 10.1016/j.ifacol.2019.08.211.
[88] P. T. Krein and M. A. Fasugba, "Vehicle-to-grid power system services with electric and plug-in vehicles based on flexibility in unidirectional charging," vol. 1, no. 1, p. 11, 2017.
[89] T. Liu and T. Shu, "Adversarial FDI Attack against AC State Estimation with ANN," *ArXiv190611328 Cs Stat*, Jun. 2019, Accessed: Apr. 07, 2021. [Online]. Available: http://arxiv.org/abs/1906.11328
[90] P.-Y. Chen, S. Yang, J. A. McCann, J. Lin, and X. Yang, "Detection of false data injection attacks in smart-grid systems," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 206–213, Feb. 2015, doi: 10.1109/MCOM.2015.7045410.
[91] M. Bogdanoski, T. Suminoski, and A. Risteski, "Analysis of the SYN Flood DoS Attack," *Int. J. Comput. Netw. Inf. Secur. IJCNIS*, vol. 5, no. 8, Art. no. 8, Jun. 2013.
[92] A. Ingle and M. Awade, "Intrusion detection for TCP-SYNC Flood attack," *Int. J. Adv. Res. Comput. Sci.*, vol. 4, no. 5, May 2013, Accessed: Jan. 05, 2021. [Online]. Available: https://search.proquest.com/docview/1443755249/abstract/F9C268C2959B4E2APQ/1
[93] N. R. Projects, "NetSim-TETCOS/DOS_Attack_in_5G_v12.1." Jul. 06, 2020. Accessed: Jan. 14, 2021. [Online]. Available: https://github.com/NetSim-TETCOS/DOS_Attack_in_5G_v12.1
[94] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
[95] "IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/ids-2018.html (accessed Dec. 05, 2019).
[96] S. M. Kasongo and Y. Sun, "A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System," *IEEE Access*, vol. 7, pp. 38597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
[97] S. Park, M. Kim, and S. Lee, "Anomaly Detection for HTTP Using Convolutional Autoencoders," *IEEE Access*, vol. 6, pp. 70884–70901, 2018, doi: 10.1109/ACCESS.2018.2881003.
[98] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.
[99] "NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/nsl.html (accessed Dec. 05, 2019).
[100] N. Moustafa and J. Slay, "The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems," in *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, Kyoto, Japan, Nov. 2015, pp. 25–31. doi: 10.1109/BADGERS.2015.014.
[101] J. Song, H. Takakura, and Y. Okabe, "Description of Kyoto University Benchmark Data," p. 3.
[102] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks," *J. Sens.*, vol. 2016, pp. 1–16, 2016, doi: 10.1155/2016/4731953.
[103] "IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/ids-2017.html (accessed Dec. 05, 2019).
[104] "Optimizers - Keras Documentation." https://keras.io/optimizers/ (accessed Dec. 13, 2019).
[105] "Regularizers - Keras Documentation." https://keras.io/regularizers/ (accessed Dec. 13, 2019).
[106] Y. Yan, L. Qi, J. Wang, Y. Lin, and L. Chen, "A Network Intrusion Detection Method Based on Stacked Autoencoder and LSTM," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149384.
[107] Y. Guan and T. Plötz, "Ensembles of Deep LSTM Learners for Activity Recognition using

Wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 2, p. 11:1-11:28, Jun. 2017, doi: 10.1145/3090076.
[108] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability Assessment of Cybersecurity for SCADA Systems," *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1836–1846, Nov. 2008, doi: 10.1109/TPWRS.2008.2002298.
[109] A. Sajid, H. Abbas, and K. Saleem, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016, doi: 10.1109/ACCESS.2016.2549047.
[110] "Adversary: Wizard Spider - Threat Actor," *Crowdstrike Adversary Universe*. https://adversary.crowdstrike.com/en-US/adversary/wizard-spider/?L=236/ (accessed Aug. 10, 2022).
[111] S. Ghosh and S. Sampalli, "A Survey of Security in SCADA Networks: Current Issues and Future Challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019, doi: 10.1109/ACCESS.2019.2926441.
[112] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, 2014, vol. 27. Accessed: Jul. 11, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html
[113] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, Apr. 2018, pp. 289–293. doi: 10.1109/ISBI.2018.8363576.
[114] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks," *Sci. Rep.*, vol. 9, no. 1, Art. no. 1, Nov. 2019, doi: 10.1038/s41598-019-52737-x.
[115] K. Dahal, "Automatic Detection of Shockable Rhythms in AED from Imbalanced ECG Dataset Using EC-WCGAN," M.S., The University of Memphis, United States -- Tennessee, 2022. Accessed: Jul. 12, 2022. [Online]. Available: https://www.proquest.com/docview/2658177541/abstract/87F22C06EB8E461APQ/1
[116] J. Choi, T. Kim, and C. Kim, "Self-Ensembling With GAN-Based Data Augmentation for Domain Adaptation in Semantic Segmentation," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6830–6840. Accessed: Jul. 12, 2022. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Choi_Self-Ensembling_With_GAN-Based_Data_Augmentation_for_Domain_Adaptation_in_Semantic_ICCV_2019_paper.html
[117] T. Salimans *et al.*, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems*, 2016, vol. 29. Accessed: Jul. 12, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html
[118] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.
[119] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *ArXiv180209477 Cs Stat*, Oct. 2018, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/1802.09477
[120] "Motor Vehicle Deaths in 2020 Estimated to be Highest in 13 Years, Despite Dramatic Drops in Miles Driven - National Safety Council." https://www.nsc.org/newsroom/motor-vehicle-deaths-2020-estimated-to-be-highest (accessed Jul. 21, 2021).
[121] F. Falcini and G. Lami, "Deep Learning in Automotive: Challenges and Opportunities," in *Software Process Improvement and Capability Determination*, vol. 770, A. Mas, A. Mesquida, R. V. O'Connor, T. Rout, and A. Dorling, Eds. Cham: Springer International Publishing, 2017, pp. 279–288. doi: 10.1007/978-3-319-67383-7_21.
[122] "Security Challenges for Connected and Autonomous Vehicles," *BAE Systems | Cyber Security &*

Intelligence. https://www.baesystems.com/en/cybersecurity/feature/security-challenges-for-connected-and-autonomous-vehicles (accessed Sep. 12, 2021).
[123] Q. He, X. Meng, R. Qu, and R. Xi, "Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles," *Mathematics*, vol. 8, no. 8, p. 1311, Aug. 2020, doi: 10.3390/math8081311.
[124] "Autonomous cars generate more than 300 TB of data per year," *Tuxera*, Jul. 02, 2021. https://www.tuxera.com/blog/autonomous-cars-300-tb-of-data-per-year/ (accessed Jul. 21, 2021).
[125] "J3016C: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International." https://www.sae.org/standards/content/j3016_202104/ (accessed Aug. 24, 2021).
[126] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021, doi: 10.1109/TITS.2020.3017882.
[127] "Automated Vehicles for Safety | NHTSA." https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety (accessed Aug. 25, 2021).
[128] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2898–2915, Nov. 2017, doi: 10.1109/TITS.2017.2665968.
[129] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth, and K. Venkatasubramanian, "Security of Autonomous Systems Employing Embedded Computing and Sensors," *IEEE Micro*, vol. 33, no. 1, pp. 80–86, Jan. 2013, doi: 10.1109/MM.2013.18.
[130] N. Statt, "Self-driving car engineer Anthony Levandowski pleads guilty to stealing Google trade secrets," *The Verge*, Mar. 19, 2020. https://www.theverge.com/2020/3/19/21187651/anthony-levandowski-pleads-guilty-google-waymo-uber-trade-secret-theft-lawsuit (accessed Sep. 12, 2021).
[131] Z. El-Rewini, K. Sadatsharan, D. F. Selvaraj, S. J. Plathottam, and P. Ranganathan, "Cybersecurity challenges in vehicular communications," *Veh. Commun.*, vol. 23, p. 100214, Jun. 2020, doi: 10.1016/j.vehcom.2019.100214.
[132] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Demo: Attacking Multi-Sensor Fusion based Localization in High-Level Autonomous Driving," in *2021 IEEE Security and Privacy Workshops (SPW)*, May 2021, pp. 242–242. doi: 10.1109/SPW53761.2021.00039.
[133] "Cybersecurity for Connected and Autonomous Vehicles," p. 36, 2019.
[134] "AAAI 2020 Conference | Thirty-Fourth AAAI Conference on Artificial Intelligence." https://aaai.org/Conferences/AAAI-20/ (accessed Sep. 12, 2021).
[135] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, no. 7, pp. 58–65, Jun. 2021, doi: 10.1145/3448250.
[136] *AAAI 20 / AAAI 2020 Keynotes Turing Award Winners Event / Geoff Hinton, Yann Le Cunn, Yoshua Bengio*, (Feb. 10, 2020). Accessed: Sep. 12, 2021. [Online Video]. Available: https://www.youtube.com/watch?v=UX8OubxsY8w
[137] M. Basnet, S. Poudyal, M. H. Ali, and D. Dasgupta, "Ransomware Detection Using Deep Learning in the SCADA System of Electric Vehicle Charging Station," *ArXiv210407409 Cs Eess*, Apr. 2021, Accessed: Jun. 04, 2021. [Online]. Available: http://arxiv.org/abs/2104.07409
[138] A. Kavousi-Fard, M. Dabbaghjamanesh, T. Jin, W. Su, and M. Roustaei, "An Evolutionary Deep Learning-Based Anomaly Detection Model for Securing Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4478–4486, Jul. 2021, doi: 10.1109/TITS.2020.3015143.
[139] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network," *2018 16th Annu. Conf. Priv. Secur. Trust PST*, pp. 1–6, Aug. 2018, doi: 10.1109/PST.2018.8514157.
[140] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, p. 100198, Jan. 2020, doi:

10.1016/j.vehcom.2019.100198.
[141] J. J. Q. Yu, “Sybil Attack Identification for Crowdsourced Navigation: A Self-Supervised Deep Learning Approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4622–4634, Jul. 2021, doi: 10.1109/TITS.2020.3036085.
[142] D. Kahneman, *Thinking, Fast and Slow*, 1st edition. New York: Farrar, Straus and Giroux, 2013.
[143] J. Vanschoren, “Meta-Learning: A Survey,” *ArXiv181003548 Cs Stat*, Oct. 2018, Accessed: Aug. 30, 2021. [Online]. Available: http://arxiv.org/abs/1810.03548
[144] R. Vilalta and Y. Drissi, “A Perspective View and Survey of Meta-Learning,” p. 20.
[145] P. Kairouz *et al.*, “Advances and Open Problems in Federated Learning,” *ArXiv191204977 Cs Stat*, Mar. 2021, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/1912.04977
[146] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated Learning with Non-IID Data,” *ArXiv180600582 Cs Stat*, Jun. 2018, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/1806.00582
[147] R. A. Mallah, G. Badu-Marfo, and B. Farooq, “Cybersecurity Threats in Connected and Automated Vehicles based Federated Learning Systems,” *ArXiv210213256 Cs*, Jun. 2021, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/2102.13256
[148] T. Zeng, O. Semiari, M. Chen, W. Saad, and M. Bennis, “Federated Learning on the Road: Autonomous Controller Design for Connected and Autonomous Vehicles,” *ArXiv210203401 Cs Eess*, Feb. 2021, Accessed: Aug. 31, 2021. [Online]. Available: http://arxiv.org/abs/2102.03401
[149] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J. Weng, “BDFL: A Byzantine-Fault-Tolerance Decentralized Federated Learning Method for Autonomous Vehicles,” *IEEE Trans. Veh. Technol.*, pp. 1–1, 2021, doi: 10.1109/TVT.2021.3102121.
[150] Y. Fu, F. R. Yu, C. Li, T. H. Luan, and Y. Zhang, “Vehicular Blockchain-Based Collective Learning for Connected and Autonomous Vehicles,” *IEEE Wirel. Commun.*, vol. 27, no. 2, pp. 197–203, Apr. 2020, doi: 10.1109/MNET.001.1900310.
[151] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic Federated Learning,” in *International Conference on Machine Learning*, May 2019, pp. 4615–4625. Accessed: Aug. 31, 2021. [Online]. Available: https://proceedings.mlr.press/v97/mohri19a.html
[152] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, “Rethinking 1D-CNN for Time Series Classification: A Stronger Baseline,” *ArXiv200210061 Cs Stat*, Feb. 2021, Accessed: Sep. 07, 2021. [Online]. Available: http://arxiv.org/abs/2002.10061
[153] A. Martins and R. Astudillo, “From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification,” in *International Conference on Machine Learning*, Jun. 2016, pp. 1614–1623. Accessed: Sep. 08, 2021. [Online]. Available: https://proceedings.mlr.press/v48/martins16.html
[154] Z. Zhang and M. Sabuncu, “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels,” in *Advances in Neural Information Processing Systems*, 2018, vol. 31. Accessed: Sep. 08, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html
[155] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, “Singular Value Decomposition and Principal Component Analysis,” in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, Eds. Boston, MA: Springer US, 2003, pp. 91–109. doi: 10.1007/0-306-47815-3_5.
[156] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1, pp. 37–52, Aug. 1987, doi: 10.1016/0169-7439(87)80084-9.
[157] “5g-network-slicing-enabling-the-smart-grid.pdf.”
[158] D. Sun *et al.*, “Integrated human-machine intelligence for EV charging prediction in 5G smart grid,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2020, no. 1, p. 139, Jul. 2020, doi: 10.1186/s13638-020-01752-y.
[159] “Vehicle charging station with built-in wireless access point, computing and storage - US 2020

207,228 A1 - PatentSwarm.” https://patentswarm.com/patents/US20200207228A1 (accessed Nov. 15, 2020).
[160] “2_Powered-by-SA_Smart-Grid-5G-Network-Slicing_China-Telecom_GSMA_v2.0.pdf.” Accessed: Nov. 15, 2020. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2020/03/2_Powered-by-SA_Smart-Grid-5G-Network-Slicing_China-Telecom_GSMA_v2.0.pdf
[161] A. J. Gonzalez *et al.*, “The Isolation Concept in the 5G Network Slicing,” in *2020 European Conference on Networks and Communications (EuCNC)*, Jun. 2020, pp. 12–16. doi: 10.1109/EuCNC48522.2020.9200939.
[162] R. K. Gupta, A. Choubey, S. Jain, R. R. Greeshma, and R. Misra, “Machine Learning Based Network Slicing and Resource Allocation for Electric Vehicles (EVs),” in *Internet of Things and Connected Technologies*, Cham, 2021, pp. 333–347. doi: 10.1007/978-3-030-76736-5_31.
[163] Md. A. Rahman, M. S. Hossain, M. M. Rashid, S. Barnes, and E. Hassanain, “IoEV-Chain: A 5G-Based Secure Inter-Connected Mobility Framework for the Internet of Electric Vehicles,” *IEEE Netw.*, vol. 34, no. 5, pp. 190–197, Sep. 2020, doi: 10.1109/MNET.001.1900597.
[164] W. Li, Z. Wu, and P. Zhang, “Research on 5G Network Slicing for Digital Power Grid,” in *2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)*, Nov. 2020, pp. 679–682. doi: 10.1109/ICEICT51264.2020.9334327.
[165] R. F. Olimid and G. Nencioni, “5G Network Slicing: A Security Overview,” *IEEE Access*, vol. 8, pp. 99999–100009, 2020, doi: 10.1109/ACCESS.2020.2997702.