

Bolstering EV Charging Ecosystem Infrastructure Resilience and Unraveling Threats - A Comprehensive Study

Khaled Sarieddine

**A Thesis
in
The Concordia Institute
for
Information Systems Engineering**

**Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montréal, Québec, Canada**

April 2024

© Khaled Sarieddine, 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Khaled Sarieddine

Entitled: **Bolstering EV Charging Ecosystem Infrastructure Resilience and
Unraveling Threats - A Comprehensive Study**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

Dr. Ciprian Alecsandru Chair

Dr. Khalil El-Khatib External Examiner

Dr. Abdelwahab Hamou-Lhadj Internal Program Examiner

Dr. Amr Youssef Internal Examiner

Dr. Mohsen Ghafouri Internal Examiner

Dr. Chadi Assi Supervisor

Dr. Sadegh Torabi & Dr. Danial Jafarigiv Co-supervisors

Approved by _____
Dr. Jun Yan, Graduate Program Director

March 27, 2024

Date of Defense

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Bolstering EV Charging Ecosystem Infrastructure Resilience and Unraveling Threats - A Comprehensive Study

Khaled Sarieddine, Ph.D.

Concordia University, 2024

The adoption of electric vehicles (EVs) has seen a significant rise in recent years, driven by the need to reduce greenhouse gas emissions and create greener cities. This has led to the development of a new EV charging ecosystem, composed of both physical and cyber systems. The physical layer consists of high-wattage IoT charging equipment and the power grid, while the cyber layer provides access and flexibility.

As the EV ecosystem has advanced, securing it has become crucial due to its critical role in providing essential services. The inter-connectivity of charging equipment and the lack of standardization make the system an attractive target for cyber attacks, with the potential to disrupt and destabilize the power grid. Khaled Sarieddine's research contributions aim to address these security challenges. The thesis provides a comprehensive analysis of the EV ecosystem, starting with a detailed literature review and the creation of a real-time co-simulation testbed that includes both cyber and physical layer components. The research develops an advanced fingerprinting technique to identify EV charging stations (EVCSs) in the wild and investigates the malware threat landscape, discovering Mirai-infected EVCSs. It also examines mobile applications as a potential attack vector against the power grid, identifying vulnerabilities that could be exploited to initiate unlawful charging sessions. Furthermore, the research assesses the security of OCPP backends worldwide, uncovering

6 zero-day vulnerabilities in each of 16 vendors studied. These vulnerabilities impact the infrastructure's confidentiality, integrity, and availability (CIA triad). To mitigate the limitations of centralized detection algorithms, the research develops an edge-based detection mechanism to identify oscillatory load attacks that leverage both physical and cyber layer features. By addressing these security challenges, this research contributes to the development of a more secure and resilient EV charging ecosystem, ensuring the reliable and safe provision of essential services to individuals and businesses.

Acknowledgments

Special appreciation goes to my Ph.D. supervisor, Dr. Chadi Assi, and my co-supervisors, Dr. Sadegh Torabi, and Dr. Danial Jafarijiv for their continuous support and help throughout this journey. I would like to thank Dr. Haidar Safa for his support and encouragement which has left an enduring mark on my academic and personal growth.

Special thanks go to my friend and colleague Dr. Mohammad Ali Sayed for all the stimulating discussions, the sleepless nights working together, and all the fun that we have had in the last 3 years.

A special note of appreciation goes to Dr. Ali AlSabeh and Dr. Hani Sami for their enduring friendship, understanding, and encouragement during the challenging moments. Your camaraderie made the journey more enjoyable and memorable. I would also like to extend my thanks to Dr. Nathalie Wehbe for her friendship and support.

My deepest thanks go to my beloved Mother, Mouna Sarieddine for her unwavering and boundless support throughout my Ph.D. journey. Without her endless encouragement, this accomplishment would not have been possible.

I also want to express my sincere appreciation to Raphaëlle Akhras for her exceptional support during my Ph.D. Her understanding, and belief in my capabilities played a pivotal role in navigating this journey. Thank you for being a constant source of strength and inspiration throughout this significant chapter.

Contents

List of Figures	x
List of Tables	xiii
1 Introduction	1
1.1 Motivation and Problem Statement	1
1.2 Contributions	5
1.3 Thesis Organization	6
2 Background Information and Literature Review	8
2.1 Background Information	8
2.1.1 Cyber Layer	8
2.1.2 Physical Layer	10
2.2 Literature Review	13
2.2.1 Co-simulation	13
2.2.2 Discovery	14
2.2.3 Attacks	17
2.2.4 Impact and Detection	18
3 A Real-Time Cosimulation Testbed for Electric Vehicle Charging and Smart Grid Security	25

3.1	Methodology	25
3.1.1	EV Fleet and EVCS Aggregation	27
3.1.2	Real-Time Power Grid Simulation	28
3.1.3	Cyber and Cyber-Physical Layer Emulation	28
3.2	Implementation and Demonstration	32
4	EV Charging Infrastructure Discovery to Contextualize its Deployment Security	38
4.1	Methodology	38
4.1.1	Device Discovery	39
4.1.2	Deployment Security	46
4.1.3	Malware Analysis	49
4.2	Experimental Results	51
4.2.1	EVCS Discovery	51
4.2.2	Remote Compromise	54
4.2.3	EVCS Malware Investigation	57
4.2.4	Recommendations	62
5	Investigating the security of ev charging mobile applications as an attack surface	64
5.1	Threat Model	64
5.2	Methodology	65
5.2.1	Mobile Application Look Up	65
5.2.2	Static Analysis	67
5.2.3	Dynamic Analysis	68
5.3	Results	70
5.3.1	Inferred Interactions	71

5.4	Identified Vulnerabilities	74
5.4.1	Attack Scenarios	76
5.4.2	Attack Feasibility	79
5.4.3	Attack Demonstration and Verification	81
5.5	Attack Implications	83
5.5.1	Attack Implications on the Power Grid	83
6	Uncovering Covert Attacks on EV Charging Infrastructure: How OCPP Backend Vulnerabilities Could Compromise Your System	92
6.1	Threat Model and Analysis Methodology	92
6.1.1	Threat Model	93
6.1.2	OCPP Backend Vulnerabilities	99
6.1.3	Ethical Consideration and Responsible Disclosure	103
6.2	Analysis Results	104
6.2.1	OCPP Backend Vulnerabilities Analysis	104
6.2.2	Attack Workflow	107
6.2.3	Attack Scenarios	110
6.3	Discussion	114
7	Edge-based detection and localization of adversarial oscillatory load attacks orchestrated by compromised EV charging stations	119
7.1	Threat Model	119
7.2	Methodology and System Model	122
7.2.1	System Model	122
7.2.2	Distributed Detection Mechanism Methodology	128
7.2.3	Distributed Mitigation Methodology	140
7.3	Experimental Results	141

7.3.1	Distributed Detection Mechanism Results	142
7.3.2	Distributed Mitigation Results	150
7.4	Evaluation, Comparison, and Discussion	155
7.4.1	Discussion of Obtained Results	155
7.4.2	Comparison with Existing Detectors	159
7.4.3	Robustness and Limitations	161
8	Conclusion and Future Directions	163
	Bibliography	166

List of Figures

Figure 2.1	Overview of the EV charging ecosystem and its interactions.	9
Figure 3.1	Overall co-simulation real-time testbed system model	26
Figure 3.2	Total EV load in 24 hours	34
Figure 3.3	Grid frequency response due to EV attack	37
Figure 4.1	Overall Advanced Discovery Methodology.	40
Figure 4.2	Google dork snippet.	43
Figure 4.3	Overall Deployment Security Analysis Framework.	46
Figure 4.4	Distribution of hosts per country and vendor.	53
Figure 4.5	Distribution among the discovered EVCS hosts	59
Figure 4.6	Distribution of discovered security issues and open services among infected hosts	60
Figure 4.7	Number of hosts discovered in 2022 and 2023	62
Figure 5.1	The overall mobile application lookup and vulnerability analysis methodology.	65
Figure 5.2	Overview of the static and dynamic analysis methodologies.	69
Figure 5.3	High-Level control flow graphs for the three interacting components within the ecosystem.	73
Figure 5.4	Sequence diagram depicting remote charging/discharging scenario.	77
Figure 5.5	EVCS device registration with the CMS using our real-time co- simulation test-bed.	81

Figure 5.6	Session confirmation showing the success of our attack by hijacking the charging of an idle vehicle.	81
Figure 5.7	Overview of the (a) Glover book 7-bus grid and (b) the impact of the line tripping attack scenario.	85
Figure 5.8	Incurred (a) transmission losses and (b) costs due to various attack scenarios.	87
Figure 5.9	Frequency behavior over time (a) without load shedding, and (b) with load shedding.	88
Figure 6.1	An overview of the EV charging ecosystem’s cyber/physical layers. The phantom CMS is added only to show a unique and advanced attack scenario (AS5).	93
Figure 6.2	OCPP backend discovered on Zoomeye [1].	95
Figure 6.3	EVCS discovered using Zoomeye [1] that publicly exposes the critical information, namely the EVCS ID.	96
Figure 6.4	EVCS commissioning lifecycle.	98
Figure 6.5	HTTP and WebSocket Handshake.	99
Figure 6.6	Access control and authentication vulnerability analysis.	101
Figure 6.7	TCP flow of OCPP backend acceptance of the connection from the phantom EVCS (test date: Nov. 11, 2022).	102
Figure 6.8	EVCS substitution and OCPP connection hijacking attack PoC workflows.	109
Figure 6.9	An overview of the two-stage persistent covert attacks (AS5).	113
Figure 6.10	WSCC 9-bus grid.	115
Figure 6.11	Frequency output under AS5.5 with 3,750 compromised EVCSs.	117
Figure 7.1	Overview of the covert attack.	120

Figure 7.2 Illustrate the different blocks used to simulate the cyber-attack scenarios. 123

Figure 7.3 EVCS log showing the different features that could be extracted from the charging station logs. 125

Figure 7.4 Flow chart describing the detection mechanism. 125

Figure 7.5 Normal charging behavior of two different charging stations. 131

Figure 7.6 Normal charging behavior of two different charging stations. 133

Figure 7.7 Co-simulation architecture [2] 140

Figure 7.8 Structure of the Long-Short Term Memory Model. 144

Figure 7.9 Structure of the Convolutional Long-Short Term Memory Model. . . 147

Figure 7.10 The variation of the generator’s speed as a result of an oscillatory load attack. 151

Figure 7.11 Load profile after mitigation. 153

Figure 7.12 The variation of the generator’s speed as a result of an oscillatory load attack followed by mitigation. 154

List of Tables

Table 2.1	Literature systematization of knowledge and comparison to previous work.	19
Table 3.1	Implemented OCPP functions and description.	29
Table 3.2	Specifications of the real-time co-simulation testbed.	33
Table 4.1	Overall results for security flaws in EVCS management systems labeled following the threat model: ◐ On-path attacker; ● Remote attacker, blank: no flaw found.	54
Table 5.1	Types of EV charging mobile applications based on their abilities. ^a Indicates the mobile application operators that possess Flaw 1 (Unverified Ownership). ^b Indicates the mobile application operators that possess Flaw 2 (Improper authorization for a critical function). ^c Indicates the mobile application operators that possess Flaw 1 and 2 but mitigate them by requiring information only found physically on the EVCS HMI.	67
Table 5.2	Attack Scenario description and impact.	89
Table 6.1	Vulnerabilities discovered in each of the 16 live CMS operators' backends.	106
Table 7.1	Optimized Hyper-Parameters for the Implemented Models	146
Table 7.2	Classifiers Outcomes	147
Table 7.3	Confusion Matrices for LSTM and ConvLSTM	148
Table 7.4	Classifiers Time	148

Table 7.5 Comparison between the decentralized approach and the centralized approach proposed in [3]. 161

Table 8.1 Contributions during the Ph.D. Program 164

Table 8.2 Other Co-authorships during the Ph.D. program 165

Chapter 1

Introduction

1.1 Motivation and Problem Statement

In recent years the demand for Electric Vehicles (EVs) has been increasing exponentially. The main driver behind this demand is the governmental policies that have been put in due to environmental concerns [4, 5, 6]. For example, 195 countries signed the Paris Climate Agreement to reduce Green House Gases (GHGs). To this end, governments are offering incentives for EV purchases ranging from rebates to road tax exemptions. Additionally, the rising gas prices also contributed to the rapid adoption of EVs worldwide [7]. In Canada, the transportation sector contributes 27% of the total emissions highlighting the importance of the electrification of the transportation system [8].

Consequently, several public and private entities have invested heavily to accelerate the deployment of supporting EV Charging Stations (EVCSs) in major cities. For instance, the Government of Canada has already invested over \$1 billion to support the increased zero-emission EV adoption, with a \$680 million initiative towards addressing the lack of charging and refueling stations in Canada by 2027 [9]. Moreover, Canada also partnered with leading automakers Volkswagen and Mercedes to help meet the growing demand for clean transportation solutions [10]. Similar governmental investments are also being put

into the clean automotive industry aiming to reach carbon neutrality [8]. For example, to support the anticipated growth in EV numbers, the US government has dedicated 8 billion USD to establishing a sufficient charging infrastructure.

The commercialization of the EVCS ecosystem is crucial to sustaining the development of the charging infrastructure further. Thus, remote control capabilities have been instilled into the ecosystem to ensure flexible management of the distributed infrastructure by the user and the operator. Such capabilities, widely popular in electric vehicle fleet management, car rental, and sharing services such as Uber and others, exposed new attack vectors that could be used by adversaries to compromise vital services [11]. Moreover, the connection of the EVCS ecosystem to the critical infrastructure renders the electric grid dependent on the security of integrated components [12]. Cybersecurity has become a major concern in recent years. As part of the United States of America's budget for the year 2024, the Cybersecurity and Infrastructure Security Agency has been allocated a total of \$3.1 billion, representing a \$145 million increase from the previous year's budget. The funding includes \$98 million for carrying out the Cyber Incident Reporting for Critical Infrastructure Act, as well as \$425 million to strengthen the agency's internal cybersecurity measures and analytical capabilities [13]. This is after the recent cyber-attacks that hit the US crippling the Colonial Pipeline and disrupting gas supplies inducing global hikes in prices [14]. Another major recent cyber-attack is the SolarWinds hack that impacted around 18,000 clients that downloaded the compromised Orion software update. The infected users include high-security profiles such as the US Department of Energy, the US Department of Homeland Security, the Center for Disease Control, and multiple Fortune 500 companies. The list of victims also includes NATO, the European Parliament, and various governments [15].

Cyber-attacks against the power grid and critical infrastructure are increasing with the rise of political tension worldwide. For example, the Stuxnet and the BlackEnergy attack to name a few. The Stuxnet malware attack of 2010 against Iran's nuclear facilities stands

out as a landmark event. This incident was the first major example of state-level attacks against the smart grid, marking a significant turning point in the field of cyber warfare. The Stuxnet malware attack was meticulously planned and executed with great precision, as it involved the use of a worm that was introduced into a Windows machine. The worm was programmed to propagate to its intended target, the Siemens PLC-S7 while maintaining the lowest possible chance of detectability. This ensured that the malware could operate undetected for a prolonged period.

The Siemens PLC-S7 is a critical component of industrial control systems and is used extensively in power plants, water treatment facilities, and other large-scale infrastructure projects. The worm, once it had infected the PLC-S7, was able to access and manipulate the centrifugal pressures, which were vital to the operation of Iran's nuclear program. By doing so, the malware was able to destroy 10% of Iran's centrifuges, causing a significant setback to their nuclear program. Its success was a testament to the capabilities of state-sponsored cyber warfare, and it highlighted the vulnerabilities of critical infrastructure systems to cyber threats. Moreover, the 2015 Ukraine BlackEnergy malware attack is a prime example of the devastating impact such attacks can have on power grids. This incident marked the first confirmed instance of a successful cyber attack that left 230,000 consumers without electricity. Another notable example of successful cyber attacks against industrial control systems is the ShadowPad malware. This malware has been used to target a range of critical infrastructure systems, including the Indian power grid. The ShadowPad attack highlights the ease with which cybercriminals can gain access to and control critical infrastructure systems, putting the safety and security of millions of people at risk. Finally, the Aurora Generator Test is yet another example of the devastating impact that cyber attacks can have on power grids. This test demonstrated how malware comprising just 30 lines of code could destroy the rotors of generators in a power grid, resulting in widespread power outages. This test serves as a stark reminder of the critical importance

of robust cybersecurity measures to protect against cyber threats to critical infrastructure systems. Additionally, in recent studies, a high-wattage botnet of IoT devices such as water heaters, air conditioners, and electric vehicle charging stations are shown to impact the grid immensely and disturb the power grid operation [16, 17].

These examples serve as a reminder of the need for robust cybersecurity measures and highlight the devastating impact that cyber attacks can have on critical infrastructure systems. These incidents underscore the urgent need for robust cybersecurity measures to protect against cyber threats and ensure the safety and security of critical infrastructure systems.

Furthermore, recent reports indicate that the EVCS ecosystem is vulnerable to remote cyber attacks, which have real-life impacts. For instance, during the recent Russian-Ukraine conflict, Russian charging stations were exploited by Ukrainian hackers and used a backdoor to display anti-war messages and render the charging stations unavailable [18]. Similar attack vectors could be exploited in a more invasive manner which could impact the power grid as shown in [17] by utilizing such a backdoor to inject malware into the ecosystem. Moreover, various other vulnerabilities have been reported in the system that provides various attack vectors into the ecosystem by targeting the EVCS firmware, the management system, and the communication links/protocols [19, 20, 21, 22]. Previous studies focused on the security of the EV ecosystem by exploring the security of the firmware, the installed management systems, and the communication link [19, 20, 21, 23, 24]. For instance, Nasr et al. [19] studied the security of the EVCS firmware and management systems and discovered 13 severe vulnerabilities. Whereas Alcaraz et al. [20] studied the security posture of the OCPP protocol, which is the main protocol used to control EVCSs, and discovered its susceptibility to Man-In-the-Middle attacks.

1.2 Contributions

Despite such efforts to explore the security of various components within the EV charging ecosystem, there is a lack of understanding about the current security posture of the EV ecosystem. Different platforms were created to simulate individual components of the EV ecosystem cyber or physical. Thus, there is a need for a co-simulation testbed that allows us and other researchers to realistically study the cyber security of the EV ecosystem and its impact on the power grid during attacks and normal behavior.

To assess the security of the ecosystem we aim to discover, and catalog charging stations at scale and assess their security, especially focusing on the malware threat landscape. Previous work has created a fingerprinting mechanism to identify EVCSs but their results were limited to EVCSs that possess keywords related to the EVCS ecosystem. Consequently, we create an advanced fingerprinting technique that is used to identify EVCSs with local management systems. We then assess their security and the spread of malware in the ecosystem as the EVCS is said to be infected with malware with little to no knowledge of it.

Moreover, we identify a lack of knowledge of the mobile applications in the EV ecosystem. To improve user experience and increase system flexibility, mobile applications have been incorporated into the EV charging ecosystem. EV charging mobile applications allow consumers to remotely trigger actions on charging stations and use functionalities such as start/stop charging sessions, pay for usage, and locate charging stations, to name a few. The lack of understanding about the extended attacker capabilities and attack implications when leveraging vulnerabilities across widely used mobile applications to perform large-scale coordinated attacks against various stakeholders and components within the EV charging ecosystem is studied.

We also study the security of OCPP backends which is an understudied component. EVCSs connect to the OCPP backend using OCPP. Thus, enabling remote control of the EVCSs using cloud management systems and mobile applications. Consequently, we study

the OCPP backend for injection and access control vulnerabilities. Access control vulnerabilities that allow spoofing provide adversaries with a new attack vector to launch covert attacks against the power grid while mitigating being discovered by operators.

Finally, we devise an edge-based detection mechanism for oscillatory load attacks. Centralized approaches fail to provide resiliency and are considered a single point of failure. Operator-centric approaches fail to mitigate multi-operator and Man-in-the-Middle (MitM) oscillatory load attacks against the power grid. Additionally, the created test bed is leveraged to evaluate a distributed mitigation technique, which can be deployed on public/private charging stations to average out the impact of oscillatory load attacks while allowing the power system to recover smoothly within 1 second with minimal overhead.

1.3 Thesis Organization

Chapter 2 provides detailed background on the cyber and physical components of the EV ecosystem and their interaction. The cyber layer consists of mobile applications, a cloud management system, an OCPP backend, OCPP communication protocol. The physical counterpart consists of the charging station hardware which is connected to the power grid.

In Chapter 3, we create a real-time co-simulation test-bed to realistically study the cyber security of the EV ecosystem and its impact on the power grid during attacks and normal behavior. We demonstrate the impact of a cyber attack using the EV ecosystem on the power grid.

In Chapter 4, we devise an advanced fingerprinting mechanism to identify EVCSs with local management systems. The identification of EVCS takes into account features and key indicators in their exposed web pages. It does not only consider keywords related to the EVCS ecosystem but also utilizes Google dorking technique and translation to increase the number of discovered EVCSs in the wild. Consequently, we assess their security using

non-invasive techniques and assesses the malware threat landscape of the EVCS ecosystem which has been widely publicized but barely studied.

In Chapter 5, we study the EV charging mobile application as a new attack vector that could be leveraged by adversaries along with other high-wattage IoTs to inflict harm on the power grid. We show that an adversary could extend their capabilities to start un-authorized charging sessions remotely to idly-connected EVCSs which has been demonstrated on two mobile applications.

In Chapter 6, we study the OCPP backend for injection and access control vulnerabilities. We devise a security assessment methodology that could be used to identify vulnerabilities in different EV ecosystems and is adaptable to other cyber-physical systems. We identify 6 vulnerabilities in each of the 16 different operators studied and discover the ability to launch covert attacks against the power grid.

In Chapter 7, we devise an edge-based detection mechanism that overcomes centralized detection mechanism limitations. Our approach mitigates the different attack vectors that can not be detected using operator-centric mechanisms. Additionally, we develop and test a distributed mitigation mechanism that deprives the attacker of the ability to synchronize large electric loads.

Finally, we summarize the thesis contributions in Chapter 8 and highlight the existing research gap that requires further consideration by the research community.

Chapter 2

Background Information and Literature Review

2.1 Background Information

The EV charging ecosystem is a cyber-physical system, composed of interacting hardware and software components. In what follows, we provide details about these components and their interactions. The EVCS ecosystem incorporates multiple entities that collaborate and interact to provide a vital service to the customers (individuals and businesses). It is the main enabler for EVs that have been spreading rapidly due to governmental policies that have driven their adoption. The EVCS ecosystem consists of a cyber and a physical layer, as shown in Figure [2.1](#).

2.1.1 Cyber Layer

The Cyber Layer is composed of multiple software components coupled with the hardware/physical counterpart. The mobile applications are publicly available and distributed through application stores (Google Play and Play Store) These applications are needed by

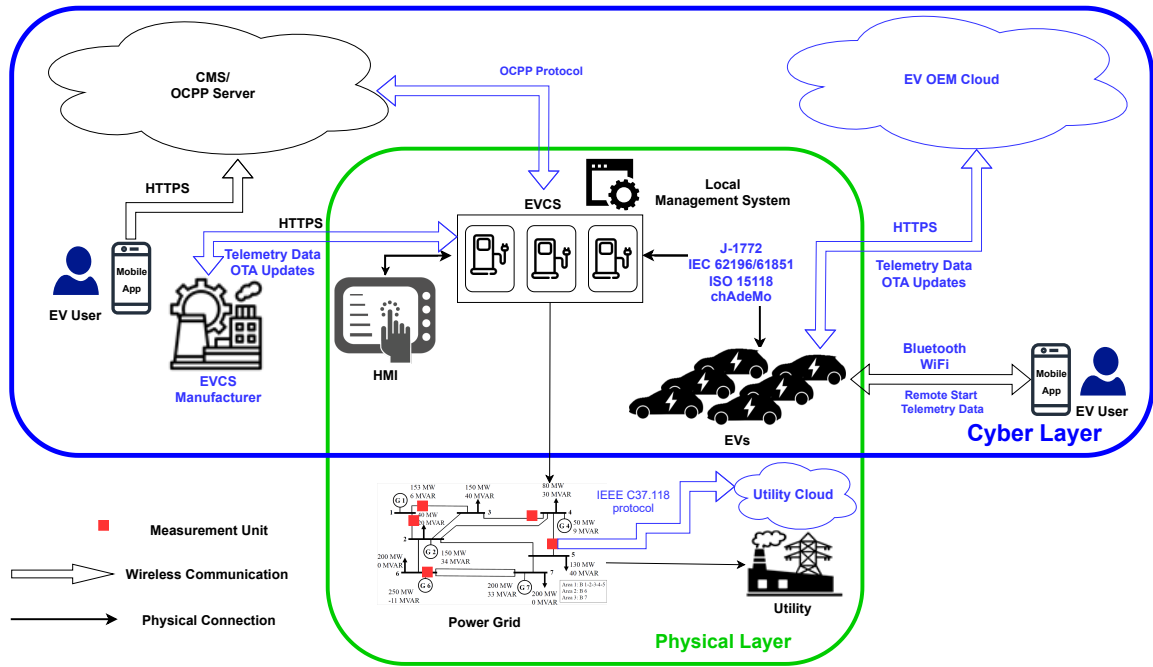


Figure 2.1: Overview of the EV charging ecosystem and its interactions.

users to control EVCSs remotely and view EVCSs' status through their communication with the CMS. The mobile applications could either be operator-specific (manage EVCSs belonging to one operator) or multi-operator (manage EVCSs of multiple operators). The multi-operator mobile applications were introduced to simplify the charging process and enable EV roaming among different operators without the need for operator-specific subscriptions. Consequently, based on the above distinction the operator-specific mobile application communicates with the operator's CMS, whereas the multi-operator mobile application communicates with its owner's backend which in turn forwards the requests to the respective operator's CMS using the Open Charge Point Interface (OCPI) [25].

The CMS plays an equally important role in the ecosystem since it provides API endpoints for the mobile application to communicate with the EVCS. Each operator has their own CMS that is responsible for reservation, scheduling, payments, management, monitoring, etc. The CMS is the most computationally capable component and it is considered the main driver of the ecosystem. However, to control EVCSs, the CMS communicates with

the EVCS using the Open Charge Point Protocol (OCPP).

The OCPP protocol is the de facto standard that is utilized to manage EVCS remotely. The OCPP defines two main roles, a lightweight client (EVCS) and a central server (CMS), which utilizes full-duplex communication over a TCP connection. The communication of the OCPP protocol is in the form of transaction functional blocks, where each entity requires a response to the initiated transaction. This standard is maintained and developed by an alliance of multiple companies working in the industry. Moreover, a connection is usually maintained between the EVCS and the original manufacturer which helps in collecting logging information about the performance of the station. We have validated these interactions between the different components and the responsibility of each entity, with our industrial partner Hydro-Quebec, a major North American utility.

Additionally, different cyber components exist however, are not considered as a core component of the EVCS ecosystem and the connection is optional and does not necessarily always exist. For example, EVCSs maintain a connection with the original equipment manufacturer (OEM) to share telemetry data about the health of the device and its operational efficiency which can be used later on to identify and enhance the product. Moreover, this connection is also used to push over-the-air (OTA) updates to ensure up-to-date software and the delivery of security patches in a timely manner. Moreover, the smart vehicles also are connected to their OEM cloud which is used to share telemetry data and OTAs as well. Finally, there are mobile applications that are used by smart vehicle owners to start/stop the engine of a vehicle, turn the conditioning, etc.

2.1.2 Physical Layer

The Physical Layer is represented by different entities. Namely, the EVCS hardware includes the human-machine interface that is used by the users to interact physically with the

EVCS. After an EVCS is manufactured and bought by an operator, the manufacturer maintains a connection to push firmware updates remotely or can make the updates available online for the operator to manage the process. Moreover, the EVs have multiple hardware and software components including remotely accessible components such as an On-board Diagnostic Port and a CAN bus. The EV charging ecosystem was established to match the demand of EVs and their need to charge. Two types of EVs are dependent on the EV charging ecosystem, which are the main foci when studying the security of the EV charging ecosystem: Plug-in Hybrid EVs (PHEVs) and Battery EVs [23, 26]. Other types of EVs such as Hybrid Electric Vehicles and Fuel Cell Electric Vehicles do not require external charging [27]. Consequently, EVs connect to the charging stations using various standards (SAE J-1772/J-2293/J-2847/J-2836, IEC 62196/61851, ISO/IEC 15118, and chAdeMO) [17, 19, 23] which are a part of the efforts to standardize communication in the EV charging ecosystem. Moreover, the IEEE 2030.5-2018 standard defines the application layer in the context of TCP/IP and facilitates the management of various utility functions related to end-user energy environments. These functions include demand response, load control, time-of-day pricing, distributed generation management, and electric vehicle integration. While this standard primarily focuses on the application protocol and its direct interaction, it also specifies the mechanisms for exchanging application messages, the specific content of those messages (including error messages), and the security features employed to safeguard the application messages. The application profile defined in this standard draws elements from various existing standards such as IEC 61968 and IEC 61850. Additionally, the CCS (Combined Charging System) is a communication protocol and charging standard used in EVs. It enables high-power charging by providing a single connector and communication interface for both AC and DC charging. It is widely adopted, CCS allows EVs to negotiate charging parameters and access a network of charging stations, making it a dominant standard in regions like Europe and North America. Moreover, ISO/IEC 15118

provides a standardized and secure interface for exchanging information during the charging process. The standard covers both wired and wireless communication interfaces, ensuring interoperability between different EVs and charging stations. It includes the Charging Communication Controller as a key component, which is responsible for handling communication between the EV and the charging infrastructure. ISO/IEC 15118 also introduces the concept of "Plug and Charge," allowing for automatic authentication and authorization of the charging session without the need for additional user interactions. Additionally, SAE J-1772, also known as the SAE Electric Vehicle Conductive Charge Coupler, is a standard that defines the physical and electrical interface between electric vehicles (EVs) and charging stations. It specifies the connector and signaling requirements for Level 1 and Level 2 charging, which are the common AC charging levels for EVs. The standard ensures interoperability and safety by providing a standardized connector design for charging infrastructure. It includes provisions for communication between the vehicle and the charging station, enabling features like power delivery control and safety interlocks. SAE J-1772 has been widely adopted in North America and is a crucial component of EV charging infrastructure in the region.

Moreover, there are several EVCS classifications. Level 1 chargers (slow chargers) are being replaced by Level 2 chargers, which are mostly used commercially as public EVCSs. Moreover, Level 3 chargers (providing a higher charging rate) are being introduced to improve the user experience and decrease charging times [17]. We focus on public EVCSs deployed by companies, governmental entities (e.g., Circuit Electric and ChargePoint), or private EVCSs that are made publicly available by the owner to earn extra income. It is worth highlighting that the EVCSs are also connected to the power grid (critical infrastructure) to draw the needed power for EVs to charge. Measurement units are distributed over the infrastructure and share information with the utility cloud. The utility utilizes this information to ensure visibility over the power grid.

2.2 Literature Review

2.2.1 Co-simulation

Previous works have implemented digital platforms for evaluation of cyber-physical systems performance. We focus mainly on power grid-connected technologies. One such example is the work in [28] where the authors survey different digital twin applications in the field of energy storage. They demonstrated how these models can accurately capture the behavior of battery systems, and digital models and couple it with digital data to create a digital twin. All these implementations however are only meant to simulate an individual aspect of their respective applications.

The work in [29] on the other hand, creates a digital twin for the integration of blockchain into the field of photovoltaic-connected microgrids. Their work models the power flow and the blockchain using their detailed mathematical models and evaluates the performance of their digital twin in terms of required computing power, energy cost, and grid voltage deviation.

Other studies have attempted to build an EV ecosystem simulator, however, their implementations were restricted to specific components such as the EV, or the scheduling of charging. In [30], the authors propose a design for a real-time simulator of the EV only. The testbed simulates the EV powertrain such as the energy consumption monitor, control units, mechanical transmission system, etc. Moreover, details like tire-road and aerodynamic information were taken into consideration. However, their work only focuses on EVs' internal components and does not study the ecosystem as a whole. Our testbed on the other hand focuses on the entire EV ecosystem instead of the EV itself.

In [31], the authors proposed a co-simulator to study the security of the OCPP protocol. Their co-simulator relied on ZeroMQ, Protobus, and a publish-subscribe model to achieve

communication between their virtual machines (VMs) hosting their EVCSs, cloud management system (CMS), and the power grid simulator. They used OpenDSS to simulate the distribution power grid. However, this raises the question of the scalability of the used grid, as the performance of OpenDSS relies on the computing power of the machine it is running on. Furthermore, their method only focuses on OCPP and disregards the other essential components of the EV ecosystem. On the contrary, our co-simulator includes all the main cyber components of the EV ecosystem and can simulate the power grid's transient and steady-state stability in real time.

Another EV co-simulator was developed in [32] to test a control scheme used to reduce the frequency fluctuations caused by the intermittency of renewable energy resources. The authors develop a control mechanism to store/inject power from the EV batteries into the grid based on frequency fluctuations. Their co-simulator consists of a real-time power grid simulator connected to a power amplifier, which is connected to 2 EVCSs. Although they were able to demonstrate their method using actual EVCSs, their co-simulation was intended to study a single aspect of EV charging and could not be used for security studies since they did not model the actual ecosystem and any of its cyber components and communication channels.

2.2.2 Discovery

In this section, we survey and discuss previous work that tackled IoT and cyber-physical system device discovery mechanisms and provide a detailed security assessment of cyber-physical systems.

Different commercial search engines exist that are used to discover, catalog, and annotate Internet-connected devices by scanning the entire IP address space. For example, Shodan [33] and Censys [34] are two commercial services that are used to discover devices. These device search engines gather information about all devices directly connected

to the Internet. Search engines query devices for various publicly available information. The bulk of the data is taken from banners, which are metadata about software that's running on a device. While these search engines provide access to structured data, they still lack the ability to label the devices due to the wide variety of IoT devices that are connected [19, 35].

Nasr et al. [19, 35], created an EVCS management system discovery mechanism that leverages passive scanning device search engines. Their approach mainly identifies charging stations that possess EVCS-related keywords and login forms in their web interface/banners. They were able to discover 44 EVCS charging vendors accumulating to 27,439 EVCS hosts, where the majority of the discovered hosts are cloud management systems. The authors utilized Shodan, Censys, and Zoomeye, however, the authors note that they were able to discover more than 90% of the EVCS hosts using Zoomeye whereas the others were only able to discover around 5000 hosts only. Moreover, it is worth highlighting that the authors disregarded the presence of EVCS hosts that do not embed EVCS keywords or do not provide a login form thus, limiting their discovery technique. Some charging station vendors do not provide a login form as the web interface is only used to display the status of the charging station and might provide different services to manage the charging station remotely such as SSH. Moreover, the authors did not take into consideration the need for translation to identify EVCSs in the wild and expand the knowledge of the ecosystem. EVCS fingerprinting is essential as it can provide utilities and attackers with a comprehensive view of the ecosystem. Finally, the authors utilized penetration testing techniques to identify vulnerabilities induced by the manufacturer/vendor such as SQL injection, XSS, etc.

In [36], the authors created an Acquisitional Rule-based Engine (ARE) for discovering IoT devices in the wild. ARE is an engine that creates association rules used to identify the discovered generic IoT devices (routers, IP cameras, etc.), that leverages the Apriori

algorithm to dynamically identify IoT devices. They extract product names that follow the observation that a general IoT device product name is a combination of letters and numbers (perhaps containing "-"). Moreover, they utilize device entity recognition that requires access to a predefined list of vendors and product names. ARE engine generates rules that are used to identify IoT devices in a fine-grained manner as compared to other existing tools. However, due to the lack of standardization in the EVCS ecosystem, such a mechanism fails to identify EVCSs as they do not follow a standardized naming convention and hence a comprehensive list of vendors and their respective products does not exist. Moreover, in [37] the authors fingerprint industrial control system management devices by actively scanning mobile communication networks in Japan and the United States of America and manually inspecting web pages. They were able to discover 21 device models accumulating to 890 hosts. They further their study by performing penetration testing techniques on 3 device models and identified 13 0-day vulnerabilities. Moreover, they developed and deployed honeypots that imitate remote ICS devices and monitored attackers' behavior to study the imminent threat that these devices are facing. However, their work only focused on attacker behavior disregarding the malware threat landscape. Similarly, in [38], the authors work on discovering Internet-connected vehicles while developing an approach that is similar to the approach proposed in [19], nasrchargeprint, and discovered 733 hosts belonging to 12 vendors and then further studied the usage of vulnerable service and identified that 91.6% of the vendors are running vulnerable services rendering the Internet-connected vehicles exposed to cyber-attacks. Moreover, Costin et al. [39] utilized supervised machine learning to classify firmware images and correlate them to the WebUI interface. Whereas, Wang et al. [40] proposed an engine for identifying IoT devices by utilizing the similarity between the response data of different IoT devices of the same vendor or product based on the structure and style of the response data. Additionally, Yu et al. [41] proposed a

firmware identification method by analyzing web page content. In contrast to other device types, EVCS has limited and non-trivial banners where most EVCMS products are closed-sourced, in addition to the lack of banner rules for identifying them [35]. Furthermore, EVCMS's lack of standardization among developers and vendors resulting makes it unfeasible to use existing approaches to fingerprint EVCSs [35].

2.2.3 Attacks

In this section, we survey and discuss previous work that tackled the security of the EV charging ecosystem's components. The security was analyzed from various perspectives, one of which discussed the security software component and the communication protocols, and the implications of the security vulnerabilities on the infrastructure.

Nasr et al. [19], studied and examined the security posture of the EVCS and their management systems. They managed to find vulnerabilities across 13 severe vulnerability classes in firmware and management systems (mobile applications and websites). It is worth mentioning that in [19], mobile applications were analyzed using only static analysis, whereas our analysis utilizes both aspects to understand the interaction of the different components without taking into consideration the interactions of the components and design flaws. Moreover, outside of academia Kaspersky Lab's team [42] analyzed the security of ChargePoint home charging station and found significant vulnerabilities in its firmware and mobile management application.

In [19, 35], the authors highlight multiple vulnerabilities in the EVCS firmware that could be exploited by adversaries to impact the power grid such as XSS, CSRF, SQL Injection, etc. The authors of [20, 43, 44] investigated the OCPP protocol from a theoretical point of view and discovered that OCPP 1.6 and 2.0.1, which is widely adopted by the industry, are vulnerable to MitM attacks if the adversary was able to break the TLS. Thus, the authors in [20, 43, 44] assume attackers can compromise confidentiality and integrity

by hijacking the communication during the early stages of the TLS handshake. Whereas, in our work we do not assume any previous vulnerability or advanced adversarial capability. However, the adversary can not interfere with the communication at later stages of the deployment and operation of the EVCS. In 2020, a more secure OCPP 2.0.1 was launched. Nevertheless, it is not backward compatible, making it difficult to go forward without making significant modifications to its products.

Furthermore, several studies were conducted to analyze the security of ISO 15118, which is used for communication between the EVCS and the EV. For instance, in [45, 46], the authors implemented a wireless attack to eavesdrop on the communication between the vehicle and the EVCS to extort sensitive information using electromagnetic side-channel attacks. Whereas in [47], the authors present attacks that impact the charging process. In [48] on the other hand, the authors proposed and demonstrated the first EV relay attack that allows the adversary to steal energy from other users by exploiting the ISO 15118. A system assessment based on the CIA triad was performed in [24]. Similarly, in [49] the ecosystem is studied against known attacks such as network and physical attacks.

To this end, while previous work has tackled the security of different components within the EV charging ecosystem, the OCPP backend systems and mobile applications remain unchecked for vulnerabilities.

2.2.4 Impact and Detection

In [51], the authors exploited publicly available data of EV chargers of the Manhattan, New York, power grid to design a novel data-driven cyberattack strategy using state-feedback-based partial eigenvalue relocation, which targets frequency stability of the power grid. The current number of EVs is not adequate to create sizable impacts, however, with the increased adoption of EVs and deployment of charging stations to match the demand, the grid will face such attacks and impacts.

Table 2.1: Literature systematization of knowledge and comparison to previous work.

Reference	Analyzed Component(s)	Theoretical Analysis	Theoretical Co-simulation Platform	White-box	Gray-Box	Manual	Automated	Known Vuln. (N-days)	Unknown Vuln. (0-days)	Covert Attacks	Impact on the Grid
Our Work	OCPP Backend (CMS)		✓	✓	✓	✓	✓		✓	✓	✓
Sarieddine et al. [50]	Mobile Application			✓	✓	✓			✓		✓
Nasr et al. [35]	EVCS			✓	✓	✓			✓		✓
Nasr et al. [19]	EVCS			✓	✓	✓			✓		✓
Antoun et al. [24]	Ecosystem Assessment	✓						✓			
Gottumukkala et al. [49]	Ecosystem Assessment	✓						✓			
Alcaraz et al. [20]	OCPP Protocol	✓		✓		✓			✓		✓
Alcaraz et al. [43]	OCPP Protocol	✓		✓		✓			✓		✓
Garofalaki et al. [44]	OCPP Protocol	✓		✓		✓			✓		✓
Baker et al. [45]	ISO 15118					✓					
Bao et al. [47]	ISO 15118	✓						✓			
Kohler et al. [46]	ISO 15118					✓					
Conti et al. [48]	ISO 15118					✓					

To initiate an oscillatory load attack from the EVCS surface, several EVCSs have to alter their charging behavior to follow a repeated on-off behavior within a very short period. The oscillatory attacks are characterized by the EV load, duration of the attack, and the instant of switching. These characteristics differ based on the power grid and its loaded conditions. Two variations of the attack exist, the charging oscillatory attack, relies on starting and stopping several charging stations. Whereas, the discharging oscillatory attack relies on charging and discharging connected EVs through several charging stations.

Different combinations of the two variations can also be included; however, this work focuses on the charging oscillatory attacks, whereas future studies could include the discharging paradigm, vehicle-to-grid (V2G), as it gets rolled out to the public. The oscillatory load attack takes advantage of load manipulation and alternates between a surge in demand which causes a frequency drop on the power grid and when the system starts its recovery and the generators start speeding up again the attacker would switch off the EVCS initiated in the first step and cause a frequency increase. This could be amplified by using discharging oscillatory load attacks, which would cause the generators to speed up due to the mismatch between the demand and extra generation [17].

Different types of oscillatory load attacks can be curated and are summarized as follows:

- Switching attacks:
 - Square wave: synchronizing the compromised load and switching them between on and off [50, 52]. This attack can be made stealthier by distributing the switching behavior on multiple EVCSs to reduce the number of events per EVCS.
 - Alternating sine wave: synchronizing only small portions of the compromised load every time step T [3, 53] (stealthier than square wave attacks and detecting them is not straightforward).

- Dynamic attacks: the size and the trajectory of the compromised load is determined by the attacker based on the grid behavior to achieve and maximize the impact on the grid instability [54].

From a grid perspective, oscillatory EV loads can be manipulated to have lower power factors [55], thus entailing a larger impact compared to residential loads [17]. Oscillatory load attacks do not require huge loads or injections to cause abnormal behavior on the power grid [50]. Even when the load is not large enough to cause generator tripping, a sustained switching attack can cause frequency and voltage oscillations, which in turn damages the turbines due to the constant acceleration and deceleration [17]. Moreover, it is worth mentioning that a variation of these attacks might target inter-area frequency as discussed in [3], these attacks are stealthy and may not be distinguished from the load variations of the grid [3, 52] which makes oscillatory load attacks initiated by the EVCS ecosystem a serious concern. Furthermore, other oscillatory load attacks can be used to force different types of oscillations, such as exciting sub-synchronous resonance [56]. Finally, in the dynamic attack scenario, the adversary induces forced oscillation without the need to excite a specific unstable mode present in the power grid [54].

It is worth noting that, the existence of various operators and the wide distribution of the charging stations, created stealthy attack vectors (adversarial) that might exploit the charging stations of different operators to create the same impact on the power grid and hinder the utilities' ability to detect and localize due to the increased complexity in monitoring the consumer loads. Consequently, to locate an attack a utility might depend on PMU measurements and other artifacts however, they do not reach the granularity of identifying the exact location of the charging station that was exploited to initiate the attack due to the wide distribution of the charging stations and the presence of multiple operators. Granular localization information is necessary for the utility to provide adequate countermeasures and create plans to secure its system.

Mohammad et al. [17] demonstrated the impact of compromising EVCSs on the power grid and launched attacks against it. Then discussed the non-linear nature of the EV charging load and simulated multiple attacks that can be launched against the power grid using these EVs. While the grid was able to recover after a 48 MW attack utilizing traditional residential loads, a smaller 30 MW EV load attack can completely destabilize the grid. Moreover, in [57], the authors study how a botnet of compromised EVs and fast-charging direct current stations can be utilized to launch cyber attacks on the power grid and its implications on the transmission and distribution networks. Additionally, in [58, 59, 60, 61, 62] they study the implications of EV charging on the grid and discuss some mitigation techniques. Moreover, in [63] they discuss the use of SMS phishing as a social-based attack. Where an attacker can send spoofed text messages to the users advertising a discount (20% off when the users charge their vehicle at noon). Consequently, they studied the impact of such an attack on the grid. However, mobile phishing attacks require knowledge of the mobile phone numbers of EV users in a certain target area, which affects the feasibility of acquiring such information. Furthermore, in [63] the attack depends on the susceptibility of users to the demand and response phishing attack.

In [3], the authors studied oscillatory load attacks and devised a centralized detection mechanism using a backpropagation neural network. The deep learning model developed can be deployed on a central management system and targets switching attacks that are initiated by the public charging station. Moreover, the proposed approach in [3] resulted in a 30% false negative for swift 20 seconds attacks which translates to 30% of the attacks being classified as normal, the uncertainty in the results motivates the need for an efficient detection mechanism. It is worth mentioning that such operator-centric mechanisms (mechanisms that are deployed on the CMS of each operator) fail to detect multi-operator oscillatory load attacks due to their ability to view the activity of other operators.

Different detection mechanisms have been proposed in the literature to identify physical layer attacks, such as False Data Injection [64], using recurrent neural networks [65], or using Bad Data Detection algorithms that rely on measurement residuals [66]. Moreover, other techniques, such as AdaBoost, random forest, and common path mining, have been studied [67, 68, 69, 70]. Additionally, a hybrid deep learning-based Dynamic Line Rating forecasting approach is used to detect cyber-attacks based on the increase in the least square errors [71]. These schemes, however, are aimed at detecting attacks that target the measurements and cyber layer of the grid without delving into attacks against the actual power consumption of the loads. The detection of oscillatory load attacks, to the best of our knowledge, hasn't been widely studied in previous work. It is worth mentioning that previous work also lacks localization methods that link attacks to particular physical locations. In this approach, because of the portability and the deployment location (EVCS), the operator can identify and localize attacks to the granularity of a charging station which allows the grid operator to create better defenses against attacks. Furthermore, the detection mechanisms that depend on state estimations and knowledge about the power grid using different devices such as PMUs may fail under attacks that target the physical and cyber layers simultaneously [72]. These types of attacks include steps to mislead the control center similar to the Ukrainian power grid attack.

Moreover, in [73] the authors created an LSTM deep learning model to detect DDoS attacks that can violate the availability of EVCSs by targeting the management system. They studied different types of DDoS attacks that will affect the availability of resources. In this work, it is not assumed that the attack has changed the attributes of network packets. Also, no access to network packets is provided before, throughout, and after the attack unlike [73]. EVCS logs are utilized to deploy a distributed detection mechanism on the charging station. Consequently, the authors in [73] furthered their study to create an IDS to detect FDI and DDoS attacks on photovoltaic controllers [74] whereas in this work oscillatory

load attacks on the cyber-layer of the EV ecosystem. Since these attack scenarios require the attacker to send a single OCPP message to the EVCS every attack period (in the order of seconds), the network behavior of such an attack does not resemble that of a DDoS which requires flooding the target with requests. As a result, such a detector cannot be utilized to detect oscillatory load attacks. Moreover, in [75] the authors devised a ransomware detection mechanism while assuming that the ransomware can initiate DDoS and FDI attacks that might alter the state of charge thresholds. The detection mechanism is based on assembly instructions that are generated after the ransomware starts executing, whereas in [76], the authors proposed an early detection mechanism based on pre-attack (paranoiac) activity that the ransomware performs before executing. In [75], the authors utilized 561 ransomware samples to train and test their deep-learning model. However, there are various classes/families, wherein [76] the authors collected about 3000 ransomware samples, which makes the data set created in [75] unrepresentative. Furthermore, both [75] and [76] would fail to detect oscillatory load attacks since these attacks are not a result of any malware/ransomware activity on the EVCS.

Chapter 3

A Real-Time Cosimulation Testbed for Electric Vehicle Charging and Smart Grid Security

3.1 Methodology

This section describes the architecture of our proposed real-time EV co-simulation testbed. Our co-simulator consists of simulated cyber and physical layer components as well as actual communication channels as illustrated in the top-view representation in Figure 3.1. This testbed can be used for various types of studies related to EV penetration levels, sizing of EVCSs, impact of EV charging on power grid stability, and most importantly evaluating the security of the EV ecosystem and implications of cyber-attacks on the power grid.

The first step of developing our testbed would be the implementation of the 2 main elements of the ecosystem which are the CMS and the EVCS and establishing communication between them through the de facto OCPP standard. We rely on the basic OCPP setup in

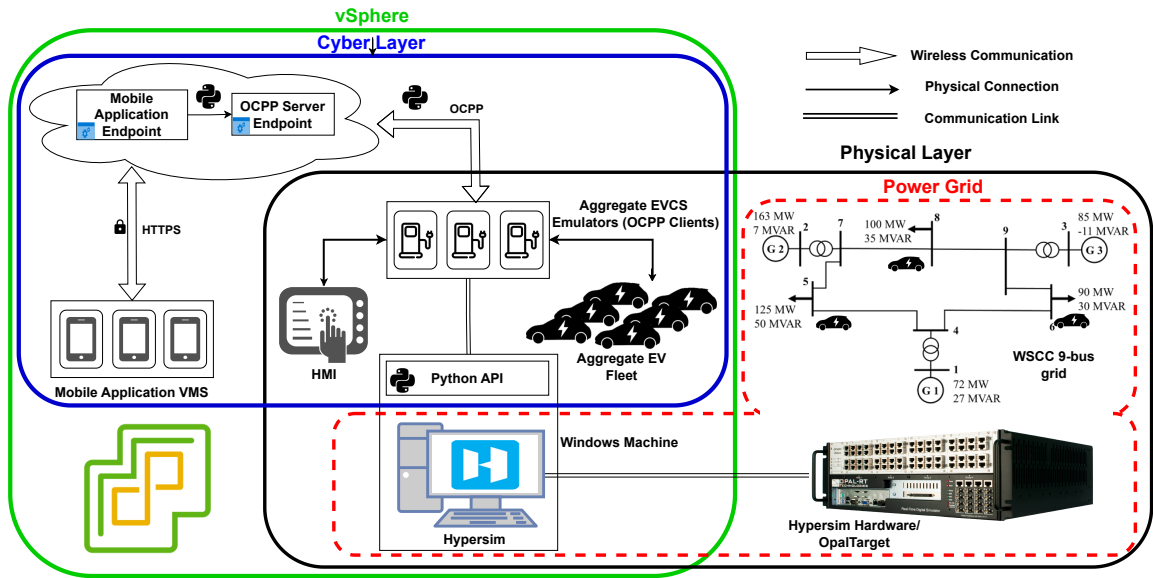


Figure 3.1: Overall co-simulation real-time testbed system model

[77] and follow the actual OCPP standard [25] to implement the protocol's functions and communication methods. We also implement the phone apps used by users to control their charging sessions and their communication with the CMS. To manage these emulated components and ensure the scalability of our testbed we leverage vSphere which is a VMware cloud computing virtualization platform deployed on a server. vSphere allows us to easily manage extremely large numbers of VMs on which we deploy our CMS, EVCSs, mobile apps, and all other emulated components on Linux-based VMs.

Given the connection of the EVCSs to the power grid, it is of the utmost importance to include the grid in our co-simulation testbed. The EVCSs need to be split among the two layers to be incorporated into our co-simulation testbed. The first layer is the EVCS firmware simulated as part of the cyber layer and the second is the power grid EVCS load.

To simulate the power grid in real-time, we utilize Hypersim which is a power grid simulator by Opal-RT that runs on a windows VM. To achieve real-time simulations, Hypersim runs on dedicated multi-processor hardware (OpalTagert) and is connected to the Hypersim VM over a Local Area Network. We also chose Hypersim for its ability to interact with our

emulated components as well as integrated Python scripting abilities.

3.1.1 EV Fleet and EVCS Aggregation

To simulate thousands of users, EVs and EVCSs, we utilized an aggregation mechanism based on the geographic location of the EVCSs. Our aggregation approach is adaptable and could be easily modified based on the studied scenarios and grids. For the sake of demonstration, we aggregate the EVCSs connected at each load bus in our grid into one VM. As for the number of vehicles in our grid, we utilize actual data based on the grid load profile and the number of cars. The specific details are later discussed in the implementation section. While we acknowledge that the current number of EVs is not sufficient to cause large impact on the power grid, the exponentially increasing trend in EV adoption will add a huge EV charging load into the grid as demonstrated in the Experimental Setup section. To this end, after scaling the total number of vehicles to our grid, we perform our study based on a future level of 50% EV adoption. We then determine the number of EVCSs based on the current global EV-to-EVCS ratio and average charging rates. We also create a data-driven model for the arrival and charging times of the EVs. To create a realistic EV load profile we independently simulate a Poisson arrival process of EVs to each EVCS. The charging time of these EVs is then simulated as a truncated Gaussian distribution. The parameters of these models of the arrival and charging time are specified for 1-hour windows for a 24-hour period. These parameters are tuned based on a real dataset containing 5 years of records for 7,000 EVCSs. This dataset was obtained from Hydro-Quebec as part of a legal agreement and research collaboration. Hydro-Quebec owns and operates, through a subsidiary, the public EVCSs in Quebec.

3.1.2 Real-Time Power Grid Simulation

For the real-time modeling of the power grid, we utilize Hypersim which allows us to simulate the power grid with all its static and dynamic behavior. Hypersim provides a flexible and scalable architecture along with high-speed parallel processing to enable real-time and realistic tests to meet the rapidly evolving requirements of the energy sector. Hypersim also allows us to observe, analyze and evaluate the impact of EVCSs on the grid in real-time. Hypersim also includes realistic models of power grid protection mechanisms and the ability to interact with hardware that is connected to the OpalTarget. After we build our power grid model in Hypersim, it is incorporated within our testbed to study the interactions between the grid and the cyber-physical layers of the EV ecosystem. Hypersim also allows us to model any new functionality and control logic we need either by building it directly in Hypersim or importing it from MATLAB Simulink.

To enable the EVCSs VMs to interact with their physical power load on Hypersim, we utilize a Python API that allows the EVCS VMs to control aggregate EV load models on the power grid buses. The delay introduced by the Python script is below $5\mu\text{s}$ which is negligible. In an actual EVCS, this communication happens over a short wire since the processor sits on top of the hardware making the delays also negligible. To simulate the electric behavior of an EVCS which is a battery charger, we use the PQ-dynamic-load model available in Hypersim. This model allows us to set the loads' voltage sensitivity, frequency sensitivity, harmonics, and all properties of a battery charger.

3.1.3 Cyber and Cyber-Physical Layer Emulation

To create an accurate representation of the EV ecosystem, we first need to emulate the ecosystem's components. As mentioned above, the cyber components are emulated on vSphere. We have chosen vSphere as our platform keeping in mind the need for future scalability as we increase the number of emulated hosts and possibly include different power

Function	From	To	Description
BootNotification	EVCS	CMS	After start-up, a request is sent with information about the EVCS (e.g. version, vendor, serial number, etc.). The receiver then responds to indicate whether it will accept the connection.
HeartBeat	EVCS	CMS	A message that ensures availability of the EVCS.
Get Configuration	CMS	EVCS	Retrieve the value of configuration settings of the EVCS to understand the available functionalities such as remote charging, smart charging, etc.
Authorize	EVCS	CMS	Before the owner of an EV can start or stop charging through the mobile application, the EVCS has to authorize the operation and confirm to the CMS that an EV is connected.
Remote Start/Stop Transaction	CMS	EVCS	After authorization, a remote start/stop transactions command can be triggered to initiate the charging session and draw power from the grid (Hypersim in our testbed).
Firmware Update and Status	CMS	EVCS	A function that is used to command the EVCS to retrieve an updated firmware from a remote entity whose location is specified via a URL.
Charging Profile	CMS	EVCS	a request that contains information about the charging profile including the charging schedule, charger cable limits, etc.
Set/Get/Clear Display Message	CMS	EVCS	A set of functions that control the display screen on the HMI and set, retrieve, toggle or clear the messages on the screen.
Meter Values	EVCS	CMS	A function that is utilized to send periodic updates to the CMS by collecting readings such as the power consumption, voltage, current, etc. Using these meter values, the CMS gathers granular information about the operation of the EVCSs.

Table 3.1: Implemented OCPP functions and description.

simulators operating in tandem. vSphere provides us with great flexibility in allocating resources in an efficient manner. The emulated cyber components of the ecosystem include the mobile applications and CMS used to provide remote capabilities and management of the EVCSs, the EVCS firmware and its Human-Machine-Interface (HMI), as well as realistic models of the EV fleet connection to the chargers and their total power drawn from the power grid. Finally, all these components are integrated together by implementing the actual communication protocols they use.

MOBILE APPLICATION The mobile application is an essential component for the commercialization of the EV ecosystem. It provides various remote capabilities such as starting, stopping, and scheduling charging. These mobile applications provide a new attack vector that could be used to impact the power grid [50] by exploiting the lack of end-end authentication between the users and their vehicles. Thus, to study the system comprehensively, we create an emulated version of the mobile application and its functionalities. For the current implementation, we focus on the starting and stopping of charging sessions which we use to demonstrate the attack scenario described below. We create three mobile

application VMs that represent the aggregate EV load connected to each bus in the chosen grid. To ensure realistic emulation of the communication initiated by the mobile apps, we use HTTPS communication between the mobile application and the CMS to emulate the real-world communication forced by Android and iOS operating systems. Specifically, we utilize HTTP Post requests to send information to the cloud backend and the HTTP Get request to retrieve information and display it on the mobile app VM.

CENTRAL MANAGEMENT SYSTEM The CMS provides two vital services that are used to enable communication between the mobile app and the EVCS. The CMS possesses a huge amount of computing power and is usually hosted on cloud computing platforms such as AWS, Google Cloud, and Azure. The first service is the mobile application endpoint that is responsible to send and receive requests from the mobile application as discussed above. The CMS will receive start and stop requests to trigger subsequent actions in the cloud where each mobile application VM sends the ID of the EVCS it is targeting. Since we are using aggregated EVs/EVCSs, each of our mobile app VMs also sends the number of EVCSs it is controlling.

Furthermore, the CMS hosts an OCPP server service and has an established communication channel with the EVCSs. The CMS translates the actions triggered by the mobile application to OCPP which is used to manage the EVCSs. The CMS OCPP server is implemented over Python 3.9 following the official standard release [25] and utilized the basic OCPP library [77].

EV CHARGING STATIONS The EVCS is the central cyber-physical component of the EV ecosystem. The EVCS would receive requests from the CMS over the OCPP protocol and manages its hardware accordingly. An EVCS is made up of an OCPP client and charging hardware that are common to all EVCSs and firmware that is specific to different manufacturers, as well as an onboard HMI. We implement the OCPP client service on the EVCS to emulate the behavior of a real EVCS and deploy it on a Linux-based VM to

represent an aggregate number of EVs/EVCSs following the aggregation logic presented above. The EVCS OCPP client initiates a connection to the CMS to establish a persistent connection that is utilized for all subsequent requests. The OCPP client also keeps track of internal information in a lightweight database, such as EVCS variables that show the status of EVCS (available, busy, error), transaction IDs, etc. To emulate the physical connection of the EVCS to the power grid using the Python API discussed in the Power Grid Simulation Section.

One important function of the EVCS is to verify that an EV is connected and inform the CMS that it can initiate a charging process. Since we are aggregating our EV charging load, the EVCS VM needs to check with the external service hosting the EV fleet model we discussed above for the number of connected EVs at the given bus. Then it reports this aggregate number to the CMS by utilizing the Authorize function implemented over OCPP and described in Table 3.1. For this work, the EVCS firmware utilizes the Pandas library available in Python to read minute-by-minute EV load values from a CSV file generated as discussed in our EV fleet aggregation section. This will be extended into a digital twin of the EV fleet that changes dynamically in real-time within our testbed.

Finally, we create a representation of the HMI available at the EVCSs that allows local authorization and payment for the charging sessions. This is emulated simply by having a script that can locally initiate a charging session. We also simulate the HMI's display screen by creating the functionality to receive, write and display messages. We aim in our future work to create a visual interface for this display instead of displaying the messages in the console.

COMMUNICATION PROTOCOLS IMPLEMENTATION The OCPP protocol is the de facto standard and the main communication protocol used between the CMS and EVCS. OCPP defines two main roles, a lightweight (client/EVCS) and a central server (server/CMS).

OCPP utilizes WebSockets that provides full-duplex communication over a TCP connection. The communication is initiated by the EVCS client when it connects to the CMS server and provides a persistent channel. We have implemented the OCPP protocol and its functions following the official documentation [25]. We implemented the most commonly used version which is OCPP v1.6 and the latest version 2.0.1. It is worth noting that, we validated our OCPP client and server implementations with production-grade EVCSs and the CMS backend provided by Hydro-Quebec as part of a legal agreement and research collaboration.

The communication of the OCPP protocol is in the form of transaction functional blocks, where each entity can initiate a transaction which requires a response from the receiving entity. Initially, when the EVCS is connected to the power outlet, it will send a boot notification to declare its presence to the CMS. The CMS in return replies with the current time, heartbeat interval, and notification if the connection was accepted or rejected. The current CMS time is used to synchronize the clock of the EVCS, whereas the heartbeat interval is used to set the heartbeat frequency of the EVCS which is used by the CMS to validate that the EVCSs are still online. It is worth noting that because of the usage of WebSockets, the OCPP standard mentions that the heartbeat time interval can be as low as once every 24 hours. However, we observed in current practices, heartbeats every 180s. We describe the OCPP functionalities we have implemented in Table 3.1.

3.2 Implementation and Demonstration

The goal of this co-simulation testbed is to generate realistic EV ecosystem behavior for cyber security and power grid stability studies. This testbed can be used to evaluate the security of the EV ecosystem based on actual communication channels, and realistic implementations of the main components within this ecosystem. Our testbed can also be used to collect realistic and real-time data on the power grid's reaction to EV charging load

Table 3.2: Specifications of the real-time co-simulation testbed.

Technology	Specification
vSphere ESXi	Version 6.0.0
Hypersim	Version 2022.1
Hypersim Simulation Step Size	25 μ s
OpalTarget	OP5707XG - RCP/HIL Virtex-7
EVCS VM	1GB 1 CPU
CMS VM	1GB 1 CPU
Mobile app VM	1GB 1 CPU
Python Interface	Python 3.7

during normal operation and during EV attacks against the grid. Different types of data and communication traffic can be monitored and studied using our testbed ranging from OCPP traffic, mobile applications to CMS communication, EVCS interaction with the power grid, EVCS logs, CMS logs, etc. Using the Hypersim functionalities related to power grid real-time monitoring, we can monitor and record measurements such as voltage and current values, power flows, frequency fluctuations, transformer loading, etc. These measurements can be collected and logged in CSV files to be analyzed later. These measurements can also be incorporated into grid protection mechanisms to add resilience to the power grid whether these mechanisms exist in Hypersim or are implemented by us.

EXPERIMENTAL SETUP AND PARAMETERS:

After preparing our testbed, with the previously mentioned emulated components, we generate multiple instances of the EVCSs and mobile apps to scale our ecosystem as required following the specification in 3.2. Given our utilization of vSphere, scaling this environment would be rather easy to achieve.

To demonstrate the power grid portion of our co-simulation testbed, we chose to build the WSCC 9-bus grid and implement its detailed and realistic generator models and control mechanisms in Hypersim. The WSCC grid is a simplified abstraction of the Western Interconnect in the United States and Canada. This grid has 9 buses, 3 generators, and 3 loads totaling 315MW. For EV fleet aggregation, we consider each of the load buses as a

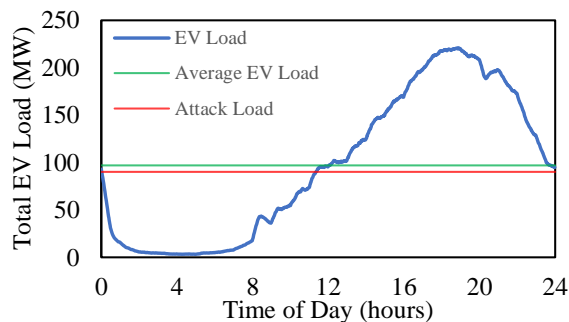


Figure 3.2: Total EV load in 24 hours

geographical area and aggregate its EVCSs giving us 3 emulated EVCSs and 3 emulated mobile applications. In case of multiple operators, each will have a CMS, 3 EVCSs, and 3 mobile apps. When the setup is complete we scale our EV fleet and our grid loads based on the New South Wales (NSW) grid. To this end, we scale the 9-bus grid based on the NSW load profile [78] and achieve a 24-hour load profile for our testbed. The minimum, average and peak loads in the NSW grid are 5,897MW, 6,968MW, and 8,214MW respectively.

The total number of registered vehicles in NSW is 5,892,206 [79]. Scaled down to fit the 9-bus grid, the number of vehicles becomes 266,367. As stated previously, we intend to perform studies on the future impact of EVs on the grid. To this end, we assume a 50% EV penetration level giving us a total of 133,184 EVs in our environment. As per the International Energy Agency (IEA) [80] EVCS operators, utilities, and governments strive to maintain sufficient EVCSs to guarantee the quality of charging services. This has resulted in a global average of 1 public EVCS for every 10 EVs on the road. According to this ratio, our ecosystem will have a total of 13,318 EVCSs distributed proportionally on the load buses. Furthermore, according to the IEA [80], based on the mixture of different charging rates, the global average rate is 24kW per EVCS.

Additionally, we extract the EVCS utilization information from the dataset provided to us by Hydro-Quebec. The average utilization rate of EVCSs is 30-32% with the peak charging demand occurring in the afternoon. By examining the dataset, we extract average

hourly arrival rates and charging times and simulate them as a Poisson process and truncated Gaussian distribution respectively. This results in a data-driven model for our EV load for a realistic implementation of our testbed. These details, however, vary from one place to the other where the charging demand in New York, for example, is larger in the morning hours. From the presented statistics and data-driven EV fleet model, we generate the EV load profile presented in Figure 3.2. Figure 3.2 demonstrates the minute-by-minute change in the EV load, the daily average EV load, and the magnitude of the EV attack load used below.

ATTACKER MODEL We consider a remote adversary that can exploit the vulnerabilities mentioned below to target EVCSs with connected EVs control the EV charging process and coordinate an oscillatory attack against the grid. To demonstrate the operation of our real-time testbed we utilize vulnerabilities discussed in previous work. The EV ecosystem is vulnerable to remote attacks and exploitation by leveraging design flaws [50] and the lack of trust model between the mobile app users and the EVCS they are controlling. The adversary can then create a botnet of genuine mobile applications to be able to utilize it as an entry point to hijack or initiate an unauthorized charging session remotely [50]. Moreover, the adversary can control ongoing charging sessions by initiating man-in-the-middle (MitM) attacks against OCPP communication using the OCPP vulnerabilities discovered in [20]. Using these vulnerabilities, the adversary can leverage compromised charging sessions to perform large-scale, coordinated attacks against the power grid. We also plan to further develop our EVCS model to include attacks that compromise the firmware of the EVCS itself which depends on specific implementations of different manufacturers.

EV ATTACK SCENARIO The EV ecosystem is connected to the power grid, which makes it of the utmost importance to have a realistic testbed to test the EV ecosystem's security and its impact on the power grid. In this paper, we utilize our real-time co-simulation testbed to evaluate the impact of attacks initiated through EV charging loads against the

power grid. However, this testbed can be used to study any other types of EV attacks on the power grid.

Oscillatory load attacks are described by an attacker's manipulation of the power grid's load following a certain oscillatory trajectory. These attacks are used to induce forced oscillations on the power grid's frequency making it deviate from the normal 60Hz operating point. This attack is initiated by increasing the power demand to cause a frequency drop and when the system starts its recovery, the attacker would decrease the load to cause a frequency spike. The attack load profile can follow an on/off behavior having a certain periodicity or it can follow a certain periodic waveform such as a sine wave.

In our attack scenario, we initiate the EV load oscillations through the mobile application and OCPP attack models described above. The attacker forces the compromised EVCS to follow an on/off pattern to cause frequency fluctuation on the grid. Our total EV attack load is equal to 90 MW or 3,750 EV/EVCSs split proportionally on the 3 attacked buses. This means that the adversary will have a 12.67-hour window between 11:20 to 24:00 where enough EVs will be connected to the EVCSs. It is noteworthy that other attacks are plausible with smaller compromised EV loads. A smart attacker would target the grid when it is at its weakest point to cause the most damage. Power grids are at their weakest point when their power demand is at its peak in the afternoon or at its lowest point after midnight. Coincidentally, the afternoon is the same time the EV load is at its highest (2.5 times larger than the needed attack load).

For the attack to remain stealthy and hidden from the utility operating the grid, we chose a stealthy/slow oscillatory attack with a frequency of 1 on/off cycle every 5s. Furthermore, this attack remains hidden from EV users since it does not require any change in user behavior. The attacker will only compromise the EVs that are connected to their respective EVCSs at the instance of attack. Furthermore, since an average EVCS would deliver less than 0.9 miles of charge in 1 min (slightly varies depending on the EV properties), an attack

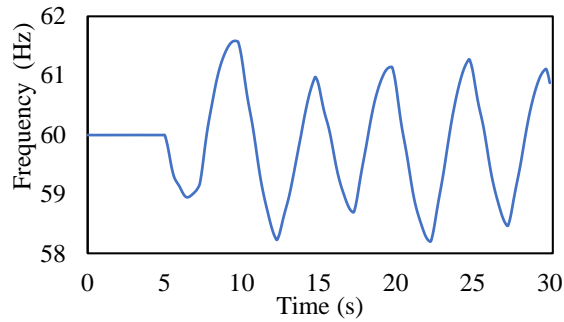


Figure 3.3: Grid frequency response due to EV attack

lasting a few seconds will have an unnoticeable impact on an EV's range.

EV ATTACK IMPACT AND DATA COLLECTION The described attack is initiated and $t=5s$ and will cause the frequency oscillations depicted in Figure 3.3 on all the grid's buses and its generators while simultaneously causing voltage fluctuations. These frequency oscillations and the deviation beyond 61.5Hz would trip the grid's generator protection relays. The entire grid in this case will lose electricity and enter into a state of blackout. We mention that most utilities have a more stringent requirement and would trip their generators when they experience 1Hz deviations. Even when the attackers control a small number of compromised EVs which is not enough to cause major frequency violations, a sustained attack will hinder the grid from returning to normal operation. A sustained oscillatory load attack would damage the generator turbines due to the induced acceleration and deceleration. Other implications include damaging electric appliances such as EVCSs and home appliances by forcing to operate at frequencies and voltages outside their rated limits.

To visualize this attack, Hypersim offers a functionality called ScopeView that allows us to monitor measurable parameters in real-time and display them in a plot format. For more advanced applications, we collect the data directly through a Python API and store them in CSV format then use them for data analytics later on. As such, Figure 3.3 was generated where the instantaneous frequency values were stored in lightweight data storage and used to plot the frequency response of the grid under attack.

Chapter 4

EV Charging Infrastructure Discovery to Contextualize its Deployment Security

4.1 Methodology

To understand the current threat landscape facing the EVCS ecosystem due to deployment (in)security, we describe our overall methodology for device discovery in Figure 5.1. We also illustrate our deployment security analysis, which is among the first attempts in the EVCS ecosystem. First, we analyze the different deployment strategies and create a discovery mechanism that aids in identifying new EVCSs with an accessible web interface to create a robust mechanism and increase the number of discovered hosts that do not necessarily embed EVCS-related keywords. Charging station vendors might create EVCS web interfaces that do not include any of the keywords that were utilized in [19, 35] as a means to create their initial discovery seed (e.g., charging station, EVCS, OCPP, etc.), but rather only include vendor or product names that require domain knowledge. Consequently, we assess the security of these EVCSs in the wild by studying their deployment security namely focusing on OWASP-Top 10 deployment security-related risks such as

security misconfiguration, vulnerable and outdated services, etc. Finally, we provide comprehensive recommendations on how to secure the deployment of the EVCSs which also requires considerable effort from the EVCS manufacturers as well.

4.1.1 Device Discovery

EVCS management systems do not expose unique services that allow their identification unless configured incorrectly. Search engines which utilize internet-wide scans and other protocols such as Modbus do not allow us to discover or uniquely identify EVCSs because these services are not restricted to EVCSs and are not used by all of them. On the other hand, some EVCSs do have a web user interface that could be used to identify them as part of the EV ecosystem or belonging to an EVCS vendor. The challenge arises in distinguishing these devices among the massive number of hosts with web interfaces, noting that in some cases these EVCSs do not have any EVCS-related keywords, especially since the lack of standardization in the ecosystem provides a considerable challenge in identifying these devices.

Our fingerprinting technique is visualized in Figure 5.1. We leverage the observation that device manufacturers embed keywords in their websites that might indicate the manufacturer/vendor and give an indication about the device. However, another challenge exists since there is no consolidated list of manufacturers and their web interfaces that allow us to easily search for EVCS hosts. In this work, we aim at addressing the limitations of [19, 35], by not limiting the search to a subset of hosts that possess EVCS keywords. Consequently, we select networks similar to [37].

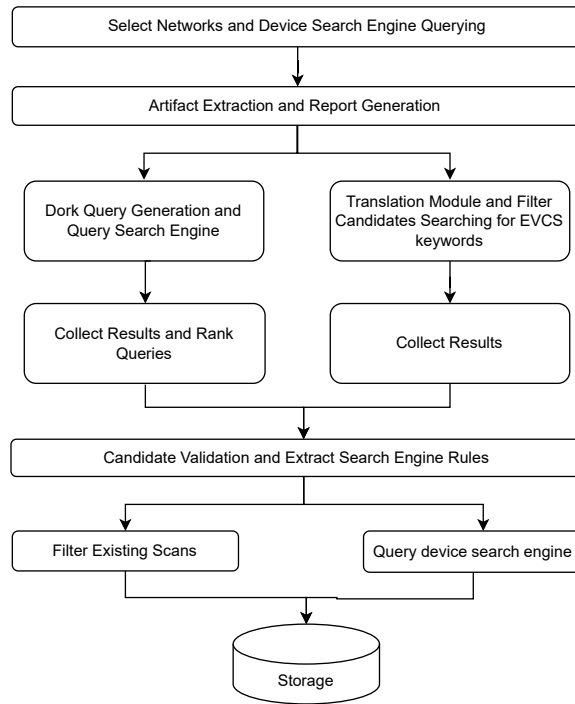


Figure 4.1: Overall Advanced Discovery Methodology.

NETWORK SELECTION AND DEVICE SEARCH ENGINE QUERYING: While internet-wide scans would identify an overwhelming number of WebUIs, we start our process by selecting specific networks where the presence of EVCS is more probable. Similar to [37], which aimed at identifying remote management systems of industrial control system devices, we expect a higher concentration of such hosts in mobile data communication networks which were part of the seed used to identify hosts. Consequently, we select ISPs as a seed for our approach thus, not limiting ourselves to a predefined seed related to EVCS keywords. We collect the WebUIs present in selected networks in Finland, France, Italy, Germany, the United States, and Canada (e.g., Vodaphone Italia). We selected networks in these countries as it has been shown in [19, 35] to have a high concentration of EVCS hosts. We were able to discover new hosts in the same area showing the advantage of our approach. The IP address range of the ISPs is obtained from publicly available AS numbers and IP address assignment information. Consequently, we leverage device search engines

that regularly scan the internet and gather information about these networks. Namely, we utilize Zoomeye [1], as it showed the best performance compared to the other device search engines.

ARTIFACT EXTRACTION AND REPORT GENERATION: Scans will provide us with EVCS banners that exist in a certain network. Consequently, we leverage the fact that EVCSs embed keywords in their WebUIs that could be used to uniquely identify them. It is worth noting that the EVCSs will share highly similar WebUIs, whereas regular websites will have a higher entropy due to the heterogeneity of the information they contain [37]. Moreover, other IoT devices, digital video recorders, and routers will also share similar WebUIs within each family of devices. We define \mathcal{W} as the candidate WebUI and \mathcal{K} as a set of fields that need to be extracted from \mathcal{W} . Namely, we create a report for each \mathcal{W} that contains $\forall k_i \in \mathcal{K}_{\mathcal{W}}$. \mathcal{K} includes the title of the tab, title of the page, headers (h1-h4), file names, paragraph fields, footer, images source link, links href, and URL links in the embedded Javascript. Consequently, each report will include a list of keywords. We further filter our candidates by rigorously filtering based on generic IoT device keywords. Some types of IoT devices, such as IP cameras, might embed keywords in their WebUI that identifies them uniquely and gives us an indication that these are not EVCSs which allows us to filter out candidates. Moreover, we further filter the reports by removing time, date, and header information along with generic stop words using the NLTK [81] python package. NLTK is a natural language toolkit that is used to work in computational linguistics to tokenize and tag text, identify named entities, and remove stop words. These generated reports provide us with a defined list of keywords that are used in our google dork tool. Candidates that do not contain unique words are then discarded as general IoT devices.

DORK QUERY GENERATION, AND SEARCH ENGINE: We leverage the generated reports to identify unique keywords found in WebUI. To distinguish EVCS local management

systems, we leverage the fact the vendors will embed data that would identify the product/vendor in the HTML code. Product names in the EVCS ecosystem do not conform to the naming convention of IoT devices thus, increasing the complexity of identifying EVCSs and rendering the methodology proposed in [36] limited to generic IoT devices. After generating a set of reports \mathcal{R} for the web interfaces, we identify the relevance of that document to the EVCS ecosystem by using Google Dorks. When Google crawls the web to index pages for its search engines, it retrieves terabytes of data. However, whenever a user searches for something on Google, millions of records are retrieved, and following their proprietary ranking algorithm it will show thousands of search results. Consequently, the user will need to go through each and every document to identify how relevant it is to their search goal. Thus, we utilize Google Dorks which is a technique used to help limit the number of retrieved results by directing the search engine to search for these keywords in certain websites or by curating a query that has certain criteria. Instead of searching for the keywords on Google and checking their relevance manually, we use an advanced searching technique that allows us to dynamically find EVCSs. This advanced search technique allows us to find information not readily available on websites. Google Dorking can return information difficult to locate. We utilize two main websites, Chargemap [82] and Plugshare [83], that are continuously updated as new vendors join. They provide a platform for locating EVCSs by the users and also might include news about the EVCS ecosystem. Such platforms continuously reflect the newest charging networks that are joining and provide a comprehensive corpus for the EVCS ecosystem. We curate queries such that we direct our search to specific websites that are related to the EVCSs. Additionally, we also curate queries where we search for keywords extracted from the HTML banners along with two keywords "charging" and "management system" which retrieve results that contain the keywords along with "charging management system". These queries give us very high confidence that the retrieved pages are related to the EVCS ecosystem. Instead

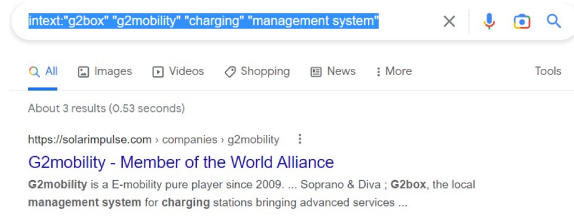


Figure 4.2: Google dork snippet.

of manual search for information on Google and trying to create relevancy between the keywords and the retrieved results, we utilize the Dorking technique to identify information in unstructured data such as Plugshare and Chargemap.

Formally, we can define our query generation using Equation (1) defined below:

Let K be the set of all combinations of keywords in Report R_i

Let \mathbb{K} be the number of keywords in Report R_i

Let k be the number of keywords chosen as input to the Query

$$\text{Select } c = \binom{\mathbb{K}}{k} \text{ where } k \in [1, \mathbb{K}] \quad (1)$$

We utilize the keyword combinations with our queries and retrieve the results. Three different query templates were used as shown below

$$\text{query}_1 = \text{site:} \text{“chargemap.com” intext: “keyword}_c \text{”}$$

$$\text{query}_2 = \text{site:} \text{“plugshare.com” intext: “keyword}_c \text{”}$$

$$\text{query}_3 = \text{intext:} \text{“keyword}_c \text{” “charging management system”}$$

We show in Figure 4.2 a sample query₁. As for the others, we follow a similar mechanism. For example, an example of the query would be intext: “SENEC” “charging” +

“management system”, where we ensure that the used keywords are related to the EVCS ecosystem by leveraging the search algorithm that is provided by Google. The results of the queries help us create a correlation between the keywords discovered and the EVCS ecosystem. Programatically, our search queries can be formatted as “search engine/search?q=site:“chargemap.com”+“g2mobility”+&btnG=Search”, where the mark (?) indicates the end of the URL, and the (&) separates arguments, q is the start of the query, the plus mark (+) represents space, and btnG=Search denotes that the search button is pressed on the web interface [36].

Consequently, after the results are collected, each query can then be ranked based on its relevance using Equation (2):

$$ChargeScore(q) = \sum t f_{EVCS_k} \sum t f_{t,d} i d f_t \quad (2)$$

where tf is term frequency, idf is inverse document frequency, q is the query, t is the term in the query, and d is the results of each query that will get a score and sorted by decreasing ChargeScore. Namely, ChargeScore is the tf-idf weighted by the EVCS keywords term frequency. The charge score takes into account if EVCS ecosystem keywords are found in the search results denoted by $EVCS_k$, showing that it has greater relevance to the EVCS ecosystem. We can then calculate the repetition of query words in the document (tf), thus showing that query keywords are present in our search results. Finally, the relative rarity of a term in the collection of results per query is calculated. This is denoted by the IDF showing the unevenness of term distribution in the corpus. This measures the informativeness of the terms, which will be very low for queries with general terms. The usage of Google-Dorking techniques alongside the ChargeScore allows us to identify accurately which queries are the most relevant to the EVCS ecosystem. Thus, showing that the studied banner of a specific host is actually an EVCS which we later validate. The higher the ChargeScore is, the higher our confidence that these query results might actually be for an EVCS vendor.

TRANSLATION MODULE AND FILTERING RESULTS: In this work, we shed light on the importance of using translation to discover new EVCSs. EVCS vendors might customize WebUIs and keywords based on the country of deployment. Thus, utilizing keywords of one language to search for EVCSs will hinder the discovery of EVCS candidates. Consequently, we translate EVCS-related keywords to different languages, mainly, Italian, French, German, and Spanish (e.g., *Système de gestion des bornes de recharge*, *Management system für Ladestationen*). We filter the WebUIs collected using this list of keywords we generated which allowed us to identify EVCSs that possess EVCS-related keywords in English as well as different languages.

VALIDATION AND SEARCH ENGINE QUERIES GENERATION: Consequently, we validate the candidates by calculating the body hash of the banners to cluster them. This led to the discovery of 28 main banner groups that we manually explore and leverage to create search engine rules. The search engine rules are utilized to scale up our discovery mechanism by leveraging a combination of artifacts that we extract from each report \mathcal{R} that would uniquely identify the candidate such as the title, file names, footer information, HTML attribute, etc. and using them as a search query on Zoomeye [1] device search engine. It is worth mentioning that the queries generated out of the previously mentioned artifacts extracted provide a unique signature that allows us to uniquely identify similar devices with similar banners. Then we utilized the hash of the banners to further validate the similarity. Finally, we filter our previous scans and query device search engines and store the results for future analysis. To scale up the detection of devices using the hosts we identified, we leverage devices search engines to increase our results by utilizing the keywords we determined as EVCS management system. We continuously followed the same approach and identified 28 device signatures accumulating 33,320 EVCS hosts belonging to 22 different vendors.

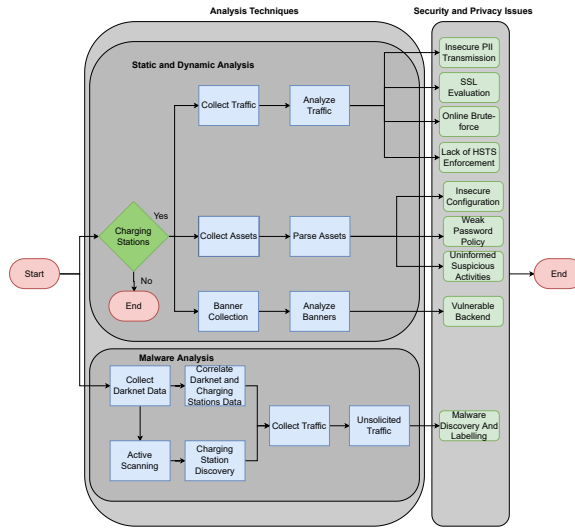


Figure 4.3: Overall Deployment Security Analysis Framework.

4.1.2 Deployment Security

We study the susceptibility of the EVCS ecosystem to remote attacks. We aim to understand the current threat facing the EVCS ecosystem. We evaluate the security, and privacy based on the General Data Protection Regulations (GDPR) password policies compliance among other security concerns that would expose the EVCS ecosystem to a multitude of attacks. Moreover, we study the malware threat landscape by providing a deeper understanding of the current threat facing the EVCS ecosystem. To this end, we propose the framework depicted in Figure 4.3 for analyzing the deployment of EVCSs to assess their deployment strategies and security practices [84, 85, 86] as summarized below.

AUTHENTICATION SECRETS LEAKAGE: We evaluate the communication protocol used by the operators to interact with the charging station management system. Namely, we try to identify the redirection to an encrypted communication channel to secure the interaction with the EVCS. Consequently, we leverage Zoomeye [1] to identify the communication protocol utilized by the charging station operators. We also confirm that by interacting with the EVCSs and transmitting a username and a password (i.e., admin, 123) using their portal we are able to identify authentication secrets transmitted in plaintext by inspecting

the traffic collected using Wireshark [87]. We search for the transmitted username or password that can be leaked via the request URL and requests' payload. **SSLSTRIP ATTACK:** To check for SSLStrip attacks, we check for the lack of HTTP Strict Transport Security (HSTS) enforcement. HSTS is a widely supported standard to protect visitors and ensure that their browsers always connect to a website over HTTPS. HSTS exists to remove the need for the common, insecure practice of redirecting users from http:// to https:// URLs. We connect to the online portal while mimicking common use case scenarios. We then utilize Burpsuite[88] to check for the lack of HSTS. Such misconfiguration means that HTTPS redirects may be putting the operators at risk. This is classified as a medium-risk vulnerability and represents low-hanging fruit for adversaries.

ONLINE PASSWORD BRUTE-FORCE AND RATE-LIMITING: Due to the connectivity of EVCSs to critical infrastructure and the features that this web portal provides (firmware update, change configuration, etc.) protecting the EVCSs from password brute force attacks is imperative. Especially that lack of rate limiting could also lead to Denial of Service. Consequently, we use Burp Suite [88] to test the existence of rate-limiting mechanisms. To keep the load on the server minimal, we test the presence of defensive mechanisms by 50 attempts on the EVCS from a single computer. We continue to monitor the performance of the EVCS to ensure that we do not impact its performance.

INSECURE CONFIGURATION: We investigate the usage of default configurations that are found in the manufacturer's manuals. This investigation is done to analyze the deployment security followed by the operators. Operators have exposed their devices to the Internet without taking security precautions to protect the ecosystem. Consequently, we investigate further deployment security measures of the operator by analyzing the configuration for 10 different vendors. We created an automated tool, that identifies the vendor of the target and tries one pair of login and password from the vendor's manuals, without trying to brute-force other combinations thus, minimizing our impact on the studied systems.

Due to ethical concerns, the tool is specifically designed to return the count of successful logins and the IP hosts, without retrieving any information or any further access to the web interface. We would like to highlight that this exposes the ecosystem to a Mirai-like attack vector (Mirai originally targeted services with the default configuration and brute-forced the login). However, we do not need to utilize brute force since we identified the specific login pair for each vendor accurately following our discovery methodology. The importance of such testing for insecure configuration lies in lowering barriers for the adversary to create an impact on the ecosystem and the connected power grid. The adversary can perform denial of service on the EVCS [19, 35], on the backend [89], can perform oscillatory load attacks which impact power grid stability [17, 19, 35, 50, 54]. Consequently, we reported our results by communicating with the manufacturer or the operator to help raise awareness.

WEAK PASSWORD POLICY AND UNINFORMED SUSPICIOUS ACTIVITIES: EVCS vendors provide the operators with the ability to change the password of their accounts that allow them to access the EVCS web portal. The password policy instilled determines the flexibility of the operator to utilize weak passwords. Consequently, to review the password policies we utilize open source intelligence (e.g., manuals) or through communicating with owners of the charging stations to understand the security controls implemented for each vendor whenever possible. Moreover, we also study the features instilled to report uninformed suspicious activities such as changing passwords.

BACKEND ASSESSMENT: Due to ethical/legal concerns, we refrain from using any invasive vulnerability scanning tools to assess the backend servers. Instead, we look into the backends' software components as disclosed by web servers frameworks in their HTTP response headers. The vulnerable backend utilized by the EVCSs exposes them to a wide range of attacks and vulnerabilities if exploited by an adversary. Consequently, we study the EVCS backend components when possible such as "Server" and "X-Powered-By" to

determine the risks associated with them. We then match these components against the CVE database to detect known vulnerabilities associated with these versions since a considerable number of the CVEs exist with an exploitable proof of concept.

4.1.3 Malware Analysis

Next, we investigate the malware threat landscape in the EVCS ecosystem through the methodology in Figure 4.3. We start by examining the EVCSs' presence on a network telescope and extracting artifacts from their network traffic. The network telescope is a portion of IP address spaces dedicated to observing inbound Internet traffic. The main outcome of the network telescope is to detect and log malicious traffic that originates from malware and viruses [90] that perform scanning actions by sending probes. We utilize the UC San Diego network telescope under CAIDA stewardship. The network is globally routed and accounts for approximately $\frac{1}{256^{th}}$ of all IPv4 Internet addresses that carry almost no legitimate traffic because there are few provider-allocated IP addresses in this prefix. The data is pre-processed and legitimate traffic is discarded from the incoming packets. The remaining data represent a continuous view of anomalous unsolicited traffic (e.g., the scanning of address space by attackers or malware looking for vulnerable targets) [91]. Consequently, we correlate the EVCSs discovered from our fingerprinting methodology with the CAIDA dataset by cross-referencing the two datasets. The detection is based on 3 million IP addresses that are detected on the darknet as scanners after monitoring traffic from February 2022 till October 2022. Namely we collect darknet scans around every two months on the following dates:

- 26, 27, 28 February 2022.
- 07, 08, 09, 10, 11 April 2022.
- 10, 11, 12, 13, 14 July 2022.

- 13, 14 October 2022.
- 15 March 2023 to 13 April 2023 every two days

ACTIVE SCANNING: Moreover, we scale our fingerprinting of EVCSs on the darknet by actively scanning the hosts with inbound traffic (~ 2 million) on 179 ports that we collected from the unique set of ports that are used by different EVCS vendors and operators as a result of our fingerprinting mechanism. We do not limit our scanning to known traditional HTTP and HTTPS ports due to the fact that EVCS manufacturers provide flexibility to operators to assign unusual ports to access their web portals. For example, Schneider EVLink EVCSs provide flexibility to the operator to assign a port between 1 and 9999 for hosting the EVCS web portal. Consequently, we utilize a two-stage approach to scan EVCSs to avoid being detected as malicious and scanning the whole port range. We first send TCP probing requests to determine the open ports based on the received replies. To this end, we utilize Zmap [92] which is a fast single-packet network scanner optimized for Internet-wide network surveys. We then utilize the resulting hosts with their respective ports for an application-layer handshake to retrieve and collect web banners using Zgrab [93] which is a stateful application-layer scanner, written in Go language and supports HTTP/HTTPS protocols. Consequently, after collecting the web banners, we extract artifacts following the proposed methodology discussed in Section 4.1.1. We scale our findings of EVCSs on the darknet by collecting web artifacts and filtering the results similar to the approach discussed above. We utilize active scanning on the darknet hosts to minimize the impact of our scanning on uninfected devices in the wild.

MALWARE FAMILY IDENTIFICATION: While the existence of the EVCS traffic on the darknet is proof of malicious EVCS behavior, we further our analysis of the traffic to identify the signature of the scanners/malware (e.g., Zmap, nmap, Mirai botnet, etc.). Mainly, we focus in this work on the Mirai malware and its variants. We collect the inbound traffic (~ 4 million packets). We note that to identify the Mirai malware and its variants, we

extract artifacts from the sent packets. Mirai and variants have a unique TCP SYN signature where the probes sent by an infected device have a TCP sequence number (normally a random 32-bit integer) equal to the destination IP address [94]. This is used to attribute the scanning to a Mirai or a variant. It is worth highlighting that Mirai traffic originating from an IP address that is associated with an EVCS is an indication that the EVCS is indeed infected[94, 95, 96, 97]. We highlight that it is statistically impossible for legitimate traffic originating from 100s of EVCSs to have a scanning signature identical to Mirai without being infected. Additionally, the data that is retained by CIADA consists entirely of malicious behavior since all legitimate traffic is filtered and discarded. While focusing on network traffic limits our result, we plan in our future work to utilize active artifact extracting tools [95] to get a deeper understanding of the other unlabelled scanners that we discovered in the EVCS ecosystem.

4.2 Experimental Results

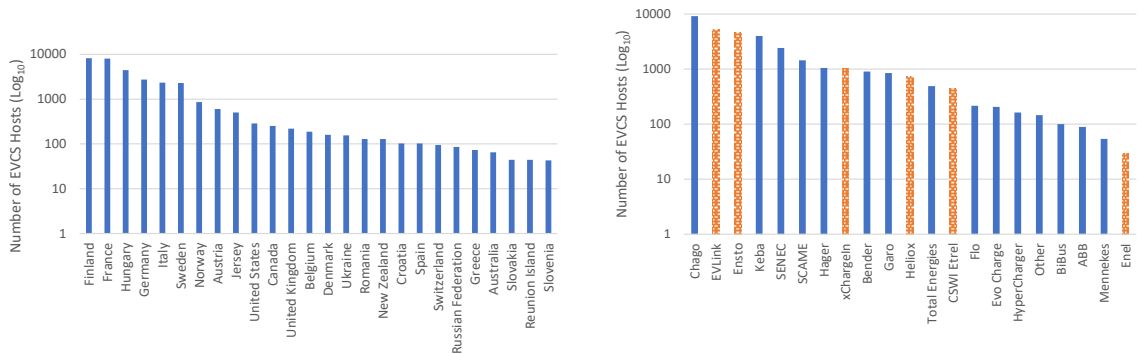
We provide a detailed discussion of our results that show the exposure of EVCSs to the internet, providing a new attack vector for adversaries to exploit. Our discovery shows the lack of proper network layer defenses to protect the charging infrastructure from remote intruders and the lack of proper security practices by the vendors and operators as they are both equally liable for securing this ecosystem.

4.2.1 EVCS Discovery

We illustrate in Figure 4.4a the geographical distribution of the discovered EVCSs. We show that EVCS management systems are mainly concentrated in Europe where Finland, Hungary, and France account for around 61% of all the discovered EVCS management systems. While this is expected because of the chosen scanned networks, we chose other

networks in North America and discovered a low number of EVCSs with exposed management systems. This is attributed to the fact that the EVCS operators and vendors in North America utilize different types of EVCSs that do not deploy management systems per device but rather connect them to the operator’s cloud management systems. Indeed, we examine the EVCS deployment of 6 different vendors in North America and discover that their deployment strategy and choice of EVCSs are keeping them from being discovered using online tools as they do not possess any web interface that might leak information indicating their correlation to the EVCS ecosystem. Partly, we attribute this to the strict government policies and interest in the security of the EVCS ecosystem [98]. However, we managed to identify Flo EVCSs by identifying their communication gateway that is used by the EVCS to communicate with the back-end systems. Flo is a charging station manufacturer that operates in North America. Through our analysis using Open Source Intelligence (OSINT) techniques, where we leverage, collect and analyze publicly shared information by the manufacturer/operator (e.g., commissioning guides, installation manuals, etc.) to get a deeper understanding of the deployment strategy, we identified that Flo charging stations are deployed with a Digi router, namely, Digi Industrial Gateway—Communication network LTE (4G) and HSPA+. Consequently, we leverage these keywords such as Digi to explore the report dataset that we collected using our aforementioned approach. After careful inspection of the retrieved candidates, we were able to identify the communication gateway. While Flo communication gateways do not provide a web interface for configuration, however, they do possess open SSH services that are running outdated versions. Thus, identifying them is important for assessing the security of the infrastructure, especially since they are utilized to route OCPP traffic that is used to manage and configure the charging station remotely.

After identifying EVCS management systems, we group the hosts based on the extracted titles. Consequently, we identified 28 clusters of devices. We notice that Ensto,



(a) EVCSs Distribution Per Country (b) EVCS Discovery Results (Uniquely discovered hosts are highlighted with solid blue).

Figure 4.4: Distribution of hosts per country and vendor.

Chago, Garo, Mennekes, and Bender possess 2 clusters each which shows that there are variations of the same product. After further inspection, we identify that these products are of two different firmware versions. Moreover, we identify two EVLink signatures where the difference between these also accounts for newer firmware being deployed on the EVCS that changes the banner and the interface. For example, older EVLink EVCSs possess “Charging Station” as a title whereas newer ones possess “EVSE Web portal”. We further elaborate on the security concern that arises from finding multiple signatures that could be attributed to running old firmware versions. Consequently, this shows the constant need to update and discover new signatures to identify EVCSs of known or unknown vendors.

For our subsequent study, we utilize the wide range of open HTTP and HTTPS ports that are known to be used to operate an EVCS management system. We note that 53% of the discovered EVCSs operate on known HTTP and HTTPS ports (e.g., 80, 8080, 443, 8443, etc.), whereas the rest operate on unusual ports such as port 30, 10000, etc. We discover 179 unique ports where hosts operate EVCS management systems. This increases the complexity of identifying EVCS management systems. Some EVCS vendors’ discoveries might be more straightforward than others. For example, Etrel EVCSs based on their installation guide recommend operating the EVCS on port 10000 and incrementing by one every time you need to add a new EVCS in the same location. Indeed, 90% of the Etrel

Table 4.1: Overall results for security flaws in EVCS management systems labeled following the threat model: ◐ On-path attacker; ● Remote attacker, blank: no flaw found.

Security Flaw	Attack Vector	# of Vendors	# EVCS Hosts
Insecure Configuration	●	10	5,240
Vulnerable Backend	●	3	9,150
Insecure Authentication	◐	18	28,046
Weak password policy	●	10	21,246
Uninformed Suspicious Activity	●	11	24,519
Online Password Bruteforce	●	12	22,506

EVCSs operate on port 10000 which aids in identifying this vendor in the future. Furthermore, EVLink which is manufactured by Schneider, operates on more than 100 ports with 97% of them operating on unusual ports such as 9100, 2082, etc. Thus, the policies instilled by the manufacturer hinder the discovery of EVCSs and add a layer of complexity in discovering them based on services and ports.

4.2.2 Remote Compromise

Following the methodology in Section 4.1. We analyzed EVCS management systems which include 33,320 EVCS distributed over 22 vendors. We devise a non-invasive security approach that could be used on other cyber-physical systems to assess the risk of remote exploits. Although, these vulnerabilities that we highlight might exist in other IoT devices, however, the EVCS ecosystem is widely distributed over very large geographical areas and connected to a very critical infrastructure. Thus, the existence of such vulnerabilities is concerning. Moreover, businesses are dependent on the service it is providing, thus, providing an attack vector that would have an economic impact in case of disruption of services. Finally, this lowers the barrier for adversaries to attack the ecosystem at scale highlighting the ecosystem’s widespread deployment insecurity. The EVCS management system is commissioned to manage individual EVCSs remotely. Namely, the portal provides the operator with the ability to change EVCS configuration, CMS communication, CMS control over the individual EVCS, reboot, firmware update, logs, and sensitive

user information. The EVCS also provides power-related functionalities such as setting the charging rate and load shedding.

In this work, we focus on studying the ability of an on-path and remote attacker to impact and intrude into EVCSs by assessing the access control measures instilled. Through our investigation, we discovered that the communication between the operator and the management systems occurs over un-encrypted channels rendering them vulnerable to Man-in-the-Middle attacks impacting 28,046 EVCSs belonging to 22 vendors except for Hager and Flo as they do not provide password protection but rather a status update that the EVCS is running. EVCS provides access to their web server over HTTP without enforcing HTTPS and HSTS to redirect the connection to a secure and encrypted one. HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses and to digitally sign those requests and responses. Thus, hindering any on-path adversary from eavesdropping on the communication and conserving the integrity of the data transferred.

Moreover, EVLink EVCSs are running a "mini-httpd 1.19 19dec2003" server, which is an early version of mini-httpd with 3 known CVEs impacting 3971 hosts. We group the hosts based on the server information and we notice the EVLink EVCSs possess two different signatures. Namely, that is because of a software update the vendor introduced. We notice that multiple devices do not provide any information about the backend system showing that some of the operators have updated their firmware. However, a considerable number did not update their firmware and accounts for 76.73% of all the discovered EVLink EVCSs. Moreover, we discover that a considerable number of EVCS are running vulnerable backends. Namely, SCAME that is running light httpd 1.4.28 that has 9 CVEs with 6 out of 9 that are of critical or high severity. This impacted 216 EVCS hosts. However, we notice that some of the EVCS operators provide partial/no information about their backend showing that the majority of the operators updated their EVCS management systems. Finally, Hager is running TwistedWeb 12.2.0 with 2 known CVEs rated as high

severity impacting 963 EVCSs distributed worldwide. We note that the proper security practice is to hide the backend system operating on the EVCS and we note that the majority (94%) of the vendors provided new updates that would hide such sensitive information from adversaries.

Moreover, we study remote attacks on the EVCSs and we discover that 67.5% of the EVCSs with password protection are vulnerable to password brute-force to the management portal that is used to configure the EVCS. Moreover, we continue to study the password policies implemented by the vendor and the presence of intrusion monitoring in case of a password change on the system. The password policy implemented by EVLink, Ensto, Mennekes, Chago, Garo, Bender, EvoCharge, HyperCharge, Etrell, and SCAME is very weak and does not have a minimum requirement of digits allowing the operator to use any password weakening the security of the ecosystem. Whereas, the Keba charging station forces a minimum of 10-character passwords with no two identical characters repeated. Moreover, we note that none of these EVCSs provide a reporting service in case of a password change, which impacts 73.58% of the discovered EVCSs.

Finally, we test these hosts for insecure configuration by testing the default logins. We scrape the manuals of the EVCSs we discovered by searching for default credentials that are utilized during setup. Mainly we test that for 10 vendors EVLink, Ensto, Mennekes, Chago, Garo, Bender, EvoCharge, HyperCharger, Etrell, and SCAME. While other EVCSs are provided with different ways of configuration and setup. For example, Eaton EVCSs provide a default password to each EVCS that is found on a configuration label in the EVCS. Consequently, we utilize our tool that connects to the EVCS management system and attempts to log in using the default credentials that we identified through scraping the configuration guides with no impact on the host, although they are vulnerable to brute-force attacks. Our non-invasive tool showed that 15.7% of the EVCSs discovered are being deployed without proper security measures by the operator. We note that alongside

we discover more than 200 EVCS cloud management systems belonging to Garo that are operating without authentication providing the adversary with access to scheduling, schedules, firmware updates, and EVCS status. Our tool could be used to provide adversaries with a Mirai-like attack vector, noting that the original Mirai malware targeted the Telnet services with default credentials similar to the current situation. This could be used to launch attacks against the EVCS ecosystem to impact the connecting critical infrastructure, confidentiality, integrity, and availability of the ecosystem. The adversary, after connecting to the management portal, will have access to multiple sensitive functionalities such as the firmware update, and configuration, which could be used to hold the operator at ransom and impact the ecosystem. Thus, highlighting the important role of both the operator and the manufacturer's lack of best security practices to secure the ecosystem. We provide a comprehensive recommendation in Section [4.2.4](#).

4.2.3 EVCS Malware Investigation

As part of our investigative study to identify the current imminent threat that is facing the EVCS ecosystem, we focus on identifying whether the EVCS ecosystem is a victim of malware attacks. We then aim to identify the type of malware that is infecting the ecosystem.

We mainly focus on Mirai which utilizes scanning activities (TCP-SYN) to find victims on the Internet. Whenever the scanner receives a reply from a victim device, the malware tries to either brute-force or exploit vulnerabilities in the device. The earliest versions of Mirai started using brute force to login into unprotected telnet services. However, after posting the Mirai-source code online, Mirai variants started to appear targeting different services and customized towards certain vulnerabilities. As part of the cyber kill chain, malware propagation is crucial to increase the number of infected victims. Consequently, scanning activities are initiated by malware to probe IP addresses that are not allocated to

any device but rather belong to CAIDA, thus showing the malicious intent of their activity [95]. We discover 79 EVCSs that were participating in scanning activities on the Internet. We first identify the IP addresses of EVCSs that were collected in the discovery phase, then we investigate their presence in the Darknet. This presence of an IP on the Darknet gives us a clear indication that the associated EVCS is participating in scanning activities. The results are then vetted by checking that the IP address is still connected to the same device with the same banner. Thus, we were able to confirm that the discovered devices are indeed EVCSs. The presence of malware is able to infect the ecosystem shedding light on the importance of securing this ecosystem proactively due to its connection to critical infrastructure.

Consequently, we investigate the type of malware that is infecting the EVCS instances by inspecting the packets it generates. Mirai malware creates packets with a unique signature where each probe has a unique TCP sequence number (normally a 32-bit integer), which is equal to the destination IP address [94]. We note that the probability that the TCP sequence number is equal to the destination IP address is $\frac{1}{2^{32}}$ showing that this is an accurate identification of Mirai variants [94]. Roughly, around 4 million data points were collected between January 2021 and October 2022. Consequently, following the approach suggested in [94, 99] we identify the scans that targeted the IPv4 space at an estimated rate of at least five packets per second. Through this work, we show that the EVCS ecosystem is a victim of traditional malware such as Mirai and its variants and requires extra attention due to its connection to critical infrastructure. While malware numbers might seem small in the EVCS ecosystem, we must keep in mind that the total number of public EVCSs is around 1.7, million which is still a very small number. In comparison, in 2016, when the Mirai Malware first surfaced, there were over 14.8 billion devices and Mirai infected around 600,000, representing a ratio of 40 Mirai infections per million IoT devices. Along the same lines, the ratio of infected EVCSs represents 33 Mirai infections per million EVCSs.

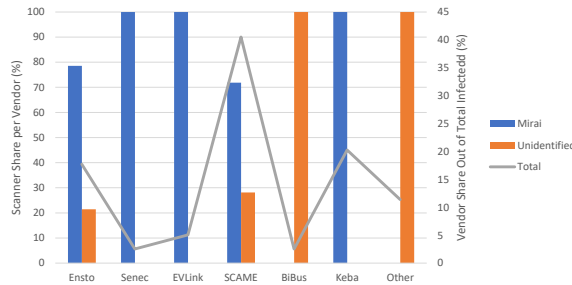


Figure 4.5: Distribution among the discovered EVCS hosts

This is to demonstrate that even though this is a relatively new environment, it is not safe from infection with Mirai malware families, however, it entails a greater risk due to the connection of the EVCS to critical infrastructure.

After further investigation of the infected samples, we categorized their distribution based on the vendor in Figure 4.5. The columns labeled Mirai and unidentified show the percentage of each type of malware among the infected hosts from each vendor. On the other hand, the line labeled Total shows the percentage of infected hosts from each vendor with respect to the total discovered EVCS hosts on the darknet. SCAME EVCSs account for 40% out of the total number of discovered hosts on the darknet followed by Keba and Ensto accounting for 20% and 17% respectively. Moreover, the Mirai-infected EVCSs account for 70% of the infected samples. This high share of Mirai is relatively understandable as new variants have been created and launched after the leakage of the source code. We note that through our analysis of the Mirai EVCSs, they generate probing requests with an average rate of 141 packets per second showing a clear indication of maliciousness in the behavior. Whereas, for the unrecognizable scanner we identify 3 different average probing rates (30.3, 168, and 446). The different probing rates give us a clear indication of maliciousness and the possible presence of 3 different malware types other than Mirai, which require further investigation. We plan in our future work to use a real-time artifact extractor proposed in [95] to identify the type of these scanners. Furthermore, the presence of a low probing rate of 17 packets per second shows that there might be stealthy malware operating

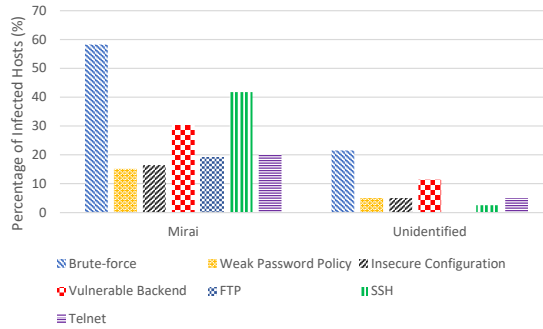


Figure 4.6: Distribution of discovered security issues and open services among infected hosts

on the EVCS ecosystem.

We further investigate the presence of security issues on infected EVCSs. We illustrate the distribution of such issues in Figure 4.6, based on the malware type. We note that these issues could be the probable entry point of the malware to the ecosystem. We note that 30% of the discovered Mirai-infected hosts are running vulnerable backends belonging to EVLink and SCAME. Whereas 16% are running with insecure configuration which provides the adversary with admin privileges over the EVCS management system. Consequently, an adversary can leverage the weak deployment security to inject malware into the EVCS by exploiting the weak access controls implemented and gaining access to different injection points such as the firmware update field [19, 35]. Moreover, the adversary could also modify the configuration of the EVCS and change the backend communication links which would allow them to remotely control the EVCSs using the OCPP protocol. Moreover, we note that 57% of the discovered hosts are vulnerable to brute force attacks whereas 15% possess a weak password policy that could be utilized by malware to get access to EVCS hosts. Moreover, we highlight that Mirai-infected EVCS hosts operate sensitive services that are well known to be used by malware to propagate, especially Mirai. The three main services are FTP, SSH, and Telnet where the majority of the Mirai-infected device operates at least one of these services. Moreover, we note that the malware could be infecting the embedded router of these devices exposing the ecosystem to a wide range

of attacks. The existing vulnerabilities of the OCPP protocol allow adversaries to launch replay attacks [20, 44]. Thus, an infected router could be used to launch replay attacks allowing adversaries to launch oscillatory load attacks, steal electricity, and steal user information (e.g., financial information). We highlight that the responsibility behind such security concerns falls upon the vendor and the operator. Where the vendor is responsible for the policies implemented and the operator is responsible for the security beyond the deployment of EVCSs.

It is worth noting that out of completeness for our malware threat landscape analysis we investigated the presence of EVCS-specific malware by analyzing the IoTPot dataset [100] and VirusTotal. The IoTPot dataset contains 92,056 IoT malware samples collected from 2016 to 2020 and the VirusTotal dataset contains malware samples collected from 2016 to 2022. We then extract strings using the Linux string utility to create a report for each malware binary. We then search for EVCS-related keywords in their binaries. While we did not find any EVCS-specific malware, we expect to see new variants as this system is proving itself to be vulnerable to remote attacks and is already being infected by traditional malware. To ensure the fairness of our methodology due to the originally selected networks, we actively scan the darknet dataset to identify new unseen EVCS hosts from September 2022 till November 2022. We note that the EVCS hosts discovered are mainly found in Europe, where 60% are found in Italy and Sweden. Whereas the rest are distributed all over Europe (Finland, Hungary, France, Germany, Romania, Croatia) and Australia.

In 2023 alone we discovered 455 EVCS participating in unsolicited scanning. Where 57% are identified as Mirai as shown in Figure 4.7. In terms of geographical distribution Italy was by far the country with the largest share of infected EVCS with 40.5% of infected hosts in 2022 and 48.1% in 2023. Whereas Sweden had the second largest share at 20.25% in 2022 and France had the second largest share in 2023 at 25.93%. Additionally, we investigated the EVCS-specific malware by leveraging the updated IoTPot data that provides

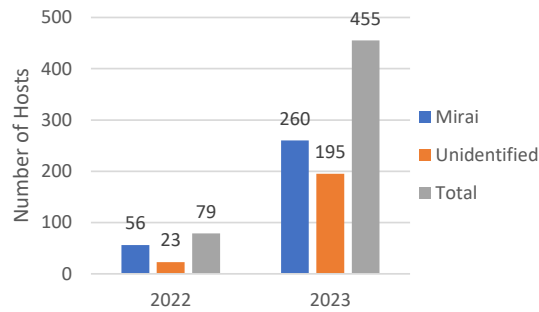


Figure 4.7: Number of hosts discovered in 2022 and 2023

IoT malware samples from 2020 till the end of 2022. Although no specific EVCS malware was discovered. The presence of Mirai on these EVCSs proves that general IoT malware poses an imminent threat to the EVCS ecosystem.

4.2.4 Recommendations

Discovering devices is a double-edged sword. Security analysts and utility operators could use it to identify EVCSs at scale, also adversaries could use it to target the EVCS instances through their vulnerable services. Various techniques could be used to protect the EVCS ecosystem, some of which are described below.

MANUFACTURER RECOMMENDATIONS To ensure that the newest and urgent security patches are implemented the manufacturer should contribute to securing the ecosystem. We recommend that manufacturers implement backward-compatible over-the-air updates that allow them to push the newest updates with minimum interaction from the operator. Current methods utilized allow the operator to install the firmware manually through the configuration portal, or through OCPP, however, we notice that there is a considerable number of operators that are not updating their firmware promptly. Moreover, we recommend that the manufacturer implement a strong password policy that forces the operator to change the password upon setup, making the EVCSs access more secure, or following the

same method utilized by some of the Etrac EVCSs. Moreover, utilizing a notification service to inform the respective operators of security events is recommended. Consequently, we recommend utilizing Two-factor authentication to increase the complexity for the adversary in accessing these web portals.

OPERATOR RECOMMENDATIONS First, EVCS operators need to deploy a middleware that would block untrusted traffic. This is achieved based on a traffic management filter in which the filter rules rely on IP reputation and the abnormal behavior shown by the scanning parties. Operators must prevent unauthorized access to their HTTP webserver which would hinder adversaries from accessing their interface. Such techniques are basic countermeasures to prevent the characterization of EVCSs based on the services and their respective HTTP web server. However, more advanced techniques could be employed such as moving target defense which increases the uncertainty and complexity for attackers by reducing their window of opportunity and increasing the costs of their probing and attack efforts. Thus, changing the mapping of an internal IP address and ports to a random external port would increase the cost of detecting the exposed services by adversaries. Thus, an advanced management technique could be employed, where the EVCS would broadcast regularly to the management system the path needed to access its web portal. The aim here is to make it harder for the adversary to access services and guess information. Moreover, we recommend following the deployment strategy adopted by ABB. Connecting to EVCSs would be through a centralized management system that the manufacturer configures for the operator such as the TerraConfig portal. Continuous patching by the operator is needed with the lack of automated firmware updates by manufacturers. Finally, ensuring communication occurs over encrypted and secure channels is of utmost importance to prevent MitM attacks.

Chapter 5

Investigating the security of ev charging mobile applications as an attack surface

5.1 Threat Model

We consider a remote adversary with access to one or more mobile applications distributed on the Google Play Store and Apple Store. Moreover, we consider the remote adversary is able to create an account on these mobile applications. Similar to [101], we do not assume any forms of software bugs or protocol vulnerabilities. The adversary relies on understanding the interactions between the components by utilizing various analysis methods to identify vulnerabilities and understand the interactions between the mobile application and the CMS. The analysis methods range from reverse engineering and white-box testing to functionality analysis, system fuzzing, and black-box analysis. The attacker aims to utilize mobile applications as an entry point to target EVCSs with connected vehicles. The adversary's goal is to exploit design flaws in the interactions among the different entities (e.g., EVCS, CMS, Mobile application) to hijack or initiate an unauthorized charging session remotely without compromising legitimate user accounts. Moreover, the adversary's ultimate goal is to leverage illegal charging sessions to perform large-scale, coordinated

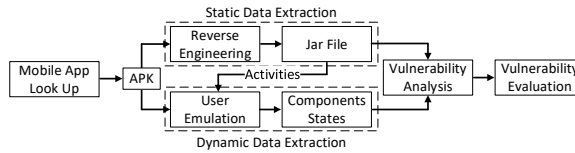


Figure 5.1: The overall mobile application lookup and vulnerability analysis methodology.

botnet attacks against the underlying critical infrastructure (e.g., the power grid) and the EV users.

5.2 Methodology

In this section, we elaborate on the analysis methodology to identify vulnerabilities that allow adversarial accounts to control charging sessions for vehicles they do not own. As shown in Figure 5.1, we combine static (reverse engineering and code review) and dynamic analysis techniques (functionality and traffic) to perform vulnerability analysis and assessment of the identified mobile applications. First, we start by fetching EV charging mobile applications, then for each product, we extract data for analysis by applying reverse engineering techniques on their binaries. Second, we extract network traffic during the functionality analysis while emulating the user behavior of the application; consequently, we analyze the application states and behavioral changes to abstract the system interactions and then evaluate it to find flaws. We provide details on the proposed methodology components below.

5.2.1 Mobile Application Look Up

In this section, we discuss our selection strategy for mobile applications. According to Statista [102], Android maintained its position as the leading mobile operating system for mobile phones with about 73% market share. We look for EV charging mobile applications on the Google Play store, which is the main platform used by Android users to

download applications. Furthermore, we automatically fetch 50 mobile applications from the Google play store. Then, we choose the mobile applications and filter them based on the features they possess by automatically searching the description for keywords such as start/stop charging. After further analysis, we discarded 8 mobile applications that are either EV charging calculators or not related. Our analysis focuses on mobile applications that provide remote control functionalities to control public EVCSs, consequently, we analyze the applications manually to ensure the existence of these functionalities as they pose a real danger to the power grid when compromised at scale [17].

Based on the prior differentiation between the applications according to their capabilities, we identify and classify the applications into three types, as shown in Table 5.1. Type 1 are applications that allow users to have an overview of the ecosystem and control the charging session, while Type 2 applications are used to perform reconnaissance activities and can only show an overview of the system. Whereas Type 3 mobile apps are developed to target home EVCS owners or businesses with private EVCSs, limiting its impact, especially from a power grid perspective. Type 3 apps could possess vulnerabilities, however, it is considered out of the scope of this work as we focus on mobile applications that provide access to public EVCSs thus, increasing the impact of an attack on the power grid. Namely, we focus on Type 1 applications that can be used to perform attacks on the power grid because of their special capability to control the EVCS and its operations. Additionally, Type 1 and 2 can be used by an adversary to prepare for their attacks by analyzing the availability of EVCSs and their usage trends, as discussed in Section 5.4.2.

Finally, we fetch the remaining mobile applications and download/install their APKs. It is crucial to highlight that any security concern discovered in the communication between the mobile application and the CMS applies to both Android applications and iOS since they rely on the same back-end that handles their requests in most cases. Therefore, we assume that our analysis methodology and results can be generalized to both platforms,

Table 5.1: Types of EV charging mobile applications based on their abilities. ^a Indicates the mobile application operators that possess Flaw 1 (Unverified Ownership). ^b Indicates the mobile application operators that possess Flaw 2 (Improper authorization for a critical function). ^c Indicates the mobile application operators that possess Flaw 1 and 2 but mitigate them by requiring information only found physically on the EVCS HMI.

Description	Type	Application Names
Remote control of charging sessions and system overview	1	<p>Remote Start Charging: ChargeHub^{a,b} - Electrify Canada^c - PodPoint^{a,b} - Electrify America^{a,b} - EVDC^{a,b} - Semma Connect^{a,b} - eCharge Network^{a,b} - Tata Power EZ Charge^c - Flo EV Charging^{a,b} - BC Hydro EV^{a,b} - Circuit Électrique^{a,b} - PlugShare^{a,b}</p> <p>Remote Start and Stop Charging: Petro-Canada EV^{a,b} - ChargePoint^{a,b} - vend-electric^{a,b} - Anywhere Charging^{a,b} - Electromaps^{a,b} - Ionity^{a,b} - Volta Charging^{a,b} - Charge Assist^{a,b} - Virta^{a,b} - Global Charge^{a,b} - EV Charging By NewMotion^{a,b} - EV Match^{a,b} - EcoFactor Network^{a,b} - FastNed^{a,b} - EVgo^{a,b} - Greenlots^{a,b} - EVduty^{a,b} - EV Connect^{a,b} - NextCharge^{a,b}</p>
System overview	2	Zap-Map - Charge Map - EVMap - Kazam EV - Chargeway - Charge Finder - Open Charge Map - EV Stations
Home charging optimization	3	EV Energy - OptiWatt - Monta EV Charging

respectively. However, confirming this will be considered for future work.

5.2.2 Static Analysis

We aim at documenting and understanding the functionality of mobile applications and their utilized libraries. We used static analysis to understand the security measures implemented by EV charging mobile application vendors to thwart automated bot attacks by examining libraries and artifacts found in the binary files, along with understanding the structure of the mobile applications (Figure 5.2a) to perform a systematic functionality analysis. Thus, we utilize reverse-engineering of the APK, which is an archive package that contains a manifest file with the package name, activity names, hardware features support, permission, and other configurations. The APK also contains the certificates for the application, a `lib` directory holding compiled libraries used by the application, and a file with compiled application code in the `dex` file format, which can be interpreted by the Dalvik Virtual Machine (DVM) and the Android run-time environment.

We extract all the files using apktool [103], which disassembles all the resources and extracts the application Manifest and the `dex` files. Consequently, we use the extracted

dex files and convert them to a JAR file using dex2jar [104] utility. We then input the file into jd-gui [105] to browse the underlying Java source code. We then analyze the extracted jar files using white-box analysis techniques to check the application resources (e.g., libraries and their functionality, certificate signing techniques used by the application developer, etc.). Further, we extract resources from the generated reports (e.g., the activities used in the mobile application and the flow of activities), which allow us to perform a detailed functional analysis and identify libraries used in EV charging mobile applications by using MobSF [106] and LiteRadar [107]. Consequently, we check the functionality of the libraries used (e.g., Google reCAPTCHA, hCAPTCHA, Anti-location Spoofing). The understanding created by studying the information obtained through static analysis is used to systematically guide the following step which is the dynamic analysis.

5.2.3 Dynamic Analysis

We rely on dynamic analysis (Figure 5.2b) to complement our static analysis method and provide a holistic and comprehensive assessment of all the interactions and functionalities provided by the mobile application. The dynamic analysis is performed through functionality analysis, recording user input, traffic analysis, and monitoring system state changes.

Functionality Analysis: We perform functionality analysis to collect data and identify system states to understand the communication between different entities. Specifically, we seek to answer the question of “How the adversary can utilize the interaction vulnerabilities and weaknesses to connect to a remote EVCS and control its operations?”

Guided by the static analysis that allowed us to understand the structure of each mobile application by mapping its activity flow, we attempt to systematically cover all the possible scenarios to systematically perform a detailed functionality analysis [108, 109]. We

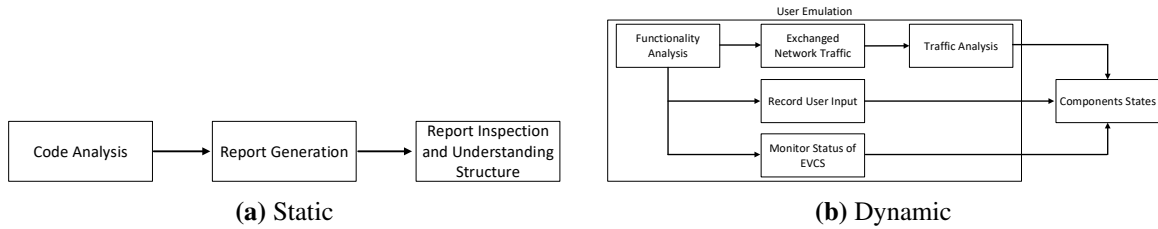


Figure 5.2: Overview of the static and dynamic analysis methodologies.

analyze each mobile application by manually mimicking regular users’ behavior and operations and triggering every functionality possible in the mobile application. Throughout our analysis, we discover that some functionalities are strictly prohibited based on the location of the device. For example, Petro-Canada EV prevents initiating charging requests if the users’ location is not in the vicinity of the EVCS. However, to mitigate that during our functionality analysis, we spoof the location of the device by broadcasting a location close to the EVCS. We utilized GPS JoyStick ADB Shell [110] to spoof the device’s location. It is worth mentioning that other applications (e.g., Electric Circuit) notify the user that they are far away from the location of the charging station however, it does not restrict the user from initiating a charging request. Furthermore, some applications (e.g., electromaps) detect that the user is far from the EVCS even if the location was spoofed. This is attributed to the IP-Geolocation services used to detect the location of the originating IP, which can be circumvented using a VPN that routes our traffic through a private tunnel that appears to originate from the same country as the charging station.

Traffic analysis: While emulating user behavior and performing user actions, we capture the traffic generated by the mobile application to understand the information that is being sent and the interactions between the mobile application and the CMS similar to [101]. We trigger important functionalities of the application such as sign up/in and start/stop charging. However, most applications use trusted Certificate Authorities (CA) to protect user privacy by encrypting the communication between the mobile application and the

CMSs [101]. To decrypt the communication we utilize an un-rooted device with the Android 7 operating system

Moreover, since Android OS with version (≥ 7.0) does not trust user-installed certificates by design. Thus, to run applications on an un-rooted device with user-installed certificates, we create a virtual space on the phone that allows running Android APKs as plugins by utilizing VirtualXposed [111]. Consequently, we perform API hooking to bypass certificate pinning/verification by using Inspeckage Package Inspector [112]. Moreover, in some applications we bypass certificate verification, by reverse engineering the application using APKTool [103], followed by injecting code into the application. We then repack and sign the application before installing it. Consequently, we utilize Burpsuite [88], which operates as a proxy server between the mobile application and the target server to intercept, inspect, and modify raw traffic passing in both directions. Our analysis was not intrusive, however, it allowed us to unravel and understand the communication between the mobile application and the CMS.

5.3 Results

We utilize different methods to infer and analyze the interaction of the main components within the EV charging ecosystem during user and device registration and when initiating charging requests. We extract the capabilities instilled in each mobile application. The capabilities recorded for each mobile application include remote start charging and remote stop charging. Moreover, we record the remote control restrictions that are implemented by the mobile application vendors to hinder the illegal or abnormal usage of a charging station. The vendors limit the flexibility for the user to initiate charging requests based on: (i) location proximity of the users (i.e., must be near the target charging station to initiate requests), (ii) IP Geolocation info (i.e., charging requests should originate from the same country/area as the target charging station), and (iii) user entered charging station ID that

is found on the target device.

Our preliminary analysis shows that all applications provide their users with a remote start charging service. Moreover, only 13 (e.g., Flo EV Charging, Plugshare, etc.) mobile applications do not provide stop-charging functionality forcing users to remove their cars when they finish charging. It is important to note that only two applications (Petro Canada EV and Electromaps) check the integrity of the users by validating the location of the device, whereas only one application (Electromaps) checks the locations of the originating IP of the charging request. Finally, two applications force the user to input a station ID or scan a QR code to initiate charging (Tata Power EZ Charge and Electrify Canada).

5.3.1 Inferred Interactions

As described in Section 5.2.3, we leveraged dynamic traffic analysis to capture and infer the interactions between the mobile application and the CMS. As an adversary, we consider the communication between the EVCS and the CMS as a black box. However, the communication between the other entities can be inferred from the different states that the mobile application goes through while performing different actions. Moreover, while previous work presented in [20, 25, 45, 113] complement our analysis of the communication between the EV, EVCS, and the CMS, they provide us with some additional insights about such communication and interactions. Specifically, it has been shown that the communication between the EV and the EVCS lacks proper security measures, which renders the underlying equipment vulnerable to remote attacks. For instance, Baker et al. [45] were able to eavesdrop on the Pulse-Width Modulation (PWM) communication, which is utilized by IEC 61851 [114, 115] for safety-related signaling mechanism between EVs and EVCSs. Furthermore, despite the added security features into the ISO/IEC 15118 (e.g., Signal-Level Attenuation Characterization and TLS encryption), the works in [45, 114, 115] highlighted the improper deployment of these features in practice. Moreover, as highlighted in [17],

most of these security features are optional and are commonly ignored by manufacturers, thus, rendering devices vulnerable in real life.

User and EVCS Registration. To this end, we analyzed the EVCS charging applications listed in Table 5.1 of Type 1 to infer the interaction between the different entities upon user registration, EVCS registration, and upon sending a charging request. We identify the main interactions with the CMS during the registration of a new user and an EV charging station (EVCS). During user registration, each user is assigned a unique identifier, which allows the user to log into the platform and use existing functionalities. There are several options for the user identifiers such as email/password combination or authentication tokens to name a few. The unique user identifier is transferred to the CMS upon user registration on a given platform and then used later for authentication purposes. On the other hand, when an EVCS is installed and made available for the public, the operator needs to register it with the CMS by sending/registering its unique identifier. This information gets saved in the CMS and used by the mobile application to identify a charging station.

Initiating Charging Requests. After the user connects the vehicle to the charging station using one of the available connectors, the user must initiate a charging request using the mobile application by selecting the desired charging station (e.g., using a map view). The application embeds the unique user identifier along with the selected station's ID in the message sent to the CMS, respectively. The CMS then sends a start charge request to the charging station with the respective ID/info. Consequently, the EVCS checks for any connected vehicle before initiating the charging session. Note that in case no vehicle was connected at the time when the user initiates a charging request, the EVCS will wait for a grace period (e.g., 5 minutes) to provide the EV owner with sufficient time to plug the charger into the vehicle. Otherwise, if a car was found to be connected to the EVCS, it will initiate the charging session by sending a confirmation message to the CMS, this is inferred by monitoring the changes on the mobile application user interface. Once the CMS

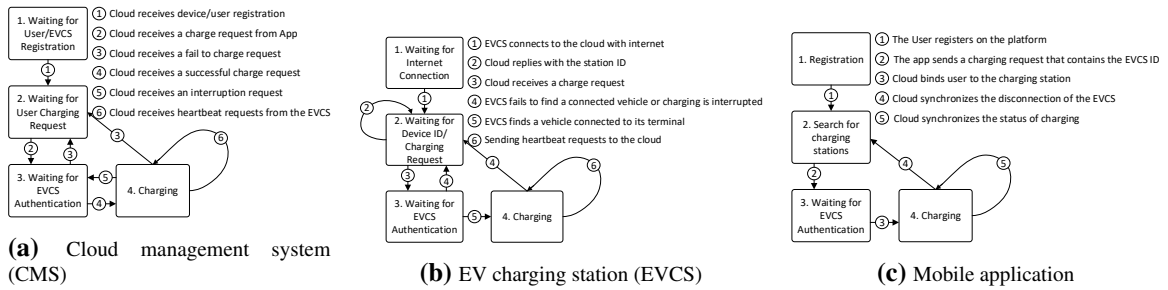


Figure 5.3: High-Level control flow graphs for the three interacting components within the ecosystem.

receives the confirmation, a correlation between the user identifier and the EVCS identifier is established. Finally, the CMS relays the charging confirmation sent by the EVCS to the mobile application.

To put this in a better context, we describe the state transitions inferred from the analysis of different platforms provided by the various EVCS operators in the industry. We validated the interactions that helped us derive the states of the components on a real-time co-simulation testbed and two EVCSs acquired from one of the biggest North American operators. Any action triggered on the mobile application has a cascading effect on the other components and will alter the state of the other components. For example, whenever a user initiates a charging session from the mobile application and transitions from S2 to S3 as illustrated in Figure 5.3c, the EVCS and the CMS will transition from S2 to S3 and eventually S4, the state of each component is dependent on the actions done by the other components. The different components making up the EVCS ecosystem are tightly coupled. The transition of one entity from one state to the other would change the current state of the ecosystem. An ideal system must strictly maintain the three-entity control flow graphs. The legitimate states of the EVCS ecosystem are depicted as a 3-tuple combination. The CMS, EVCS, and the mobile application must strictly maintain the following 3-tuple states at all times to avoid potential attacks. For example, if an attacker is able to induce the cloud and the EVCS to transition to state S3 while the legitimate user is still in state S2 shows that the charging session was hijacked by a third party (adversary). Moreover,

another 3-tuple state if triggered can also lead to similar consequences by forcing the cloud and the EVCS to transition to state S4 whereas the legitimate user mobile application is in state S1 or S2. Whenever a legitimate user looking to charge his vehicle using a certain EVCS the cloud state and the EVCS state should allow future transitions rather than being blocked. If user B is charging user A is not allowed to access the EVCS remotely. Note that all components have the same numeric state at all times, except at S1, where the CMS can be in S1 while the other components could be in either S1 or S2. Consequently, through this work we aim to dissect the intricate interactions of the components to trigger illegal states. The control flow graphs show the general operation of the EVCS ecosystem and show the type of information that is shared during the process. Thus, through our analysis, we identify several vulnerabilities related to the trust model adopted in these ecosystems. Moreover, a deep understanding of a system allows us to identify flaws in the ecosystem design which are related to how the components interact with each other. Thus, we performed a systematic unraveling of the different components and their interactions.

5.4 Identified Vulnerabilities

In what follows, we present examples of the identified vulnerabilities that can be exploited to perform remote attacks against the EV charging ecosystem and the various involved stakeholders (e.g., EV consumers and the power grid). We leveraged the described analysis methodology along with the inferred traffic/interactions in the previous sub-section to identify vulnerabilities.

Specifically, we discovered three major security weaknesses that are inherited from design and implementation flaws in the studied EV charging mobile application (Type 1). The EVCS charging platform does not strictly comply with the legitimate 3-tuple states. We found that the three entities stay in multiple unexpected 3-tuple states. The first unexpected state is (S4, S4, S1/S2); the CMS and the EVCS transition to the charging state, whereas

the mobile application user is either still in registration or EVCS discovering state. This illegal state combination when exploited by an adversary could allow for remote charging/discharging session hijacking.

In what follows, we elaborate further on the root cause of such behavior, the identified vulnerabilities, and their implications. It is worth noting that some mobile applications (e.g., EVMatch) mitigate the first unexpected state by forcing the user to reserve a spot beforehand. Whereas, other applications (e.g., Tata Power EZ Charge) hinder remote hijacking by forcing users to scan a QR code on the EVCS. However, when statically analyzing the mobile applications, QR codes are saved in a temporary file in the external SD card, which allows an on-device attacker to get access to that information to hijack the charging sessions. It is worth noting that some applications hide the access to charging behind payment gateway (e.g., buying store credit). However, adversaries can overcome this by buying store credit, which will provide access to the charging infrastructure. In what follows we focus only on the flaws that can be used to manipulate the EVCSs for attacking the grid.

Flaw 1 (F1): Unverified ownership. Ideally, an EV user should be the owner or authorized user of the vehicle. Thus, the EV user should be the sole entity to authorize any form of control or action on the vehicle. Interestingly, our static and dynamic analysis results indicate that the mobile applications do not verify user ownership over target vehicles when initiating charging requests. In other words, an EV charging mobile application user can initiate a charging request to any vehicle connected to the network since both the mobile application nor the CMS do not have a mechanism to bind the application user to the target vehicle. Given that all communications of the mobile application go through the CMS, it is imperative to have the CMS verify critical operations such as vehicle identification and ownership management. On the other hand, access to vehicles from an unauthorized user is not verified within the EV charging mobile application platforms. Therefore, rendering

the EV exposed to any user who can claim ownership and control over its charging functions when connected to the EVCS. Thus, leading to unexpected behaviors and potentially exploitable states.

Flaw 2 (F2): Improper authorization for a critical function. Starting and stopping charging operations on a given EVCS are considered examples of critical functions, which could be abused by adversaries (unauthorized users) to destabilize the operations of the EVCS and the connected power grid. This was clearly demonstrated in [17], where the authors measured the impact of mass charging and stopping operations on the power grid. Ideally, an EV owner/operator should be the only user authorized to perform critical functions on the vehicle. Exposing critical functionalities essentially allows adversaries to control charging sessions, which is considered as the first step toward initiating mass charging attacks on the grid. Nevertheless, our analysis results indicate that there is a lack of authorization, which allows any actor to perform critical functionalities on the connected EVs. This is closely related to our first finding (F1), it is mainly due to the fact that the binding step happens only based on the user and charging station IDs without further verification of the EV ownership or binding to specific mobile users. Therefore, an adversary can utilize fake accounts to hijack sessions.

5.4.1 Attack Scenarios

An attacker can leverage the discussed vulnerabilities to launch various malicious activities against the EVCS and its operations such as remote charging sessions hijacking. To do this, attackers need to control a number of adversarial accounts (i.e., bots), which provide access to existing charging services through mobile applications. Note that adversaries do not need to exploit or hijack user accounts to create the required botnet. The attackers can easily create their own botnet of legitimate mobile application accounts. The only security measures in place rely on SMS or email authentication/verification during account creation

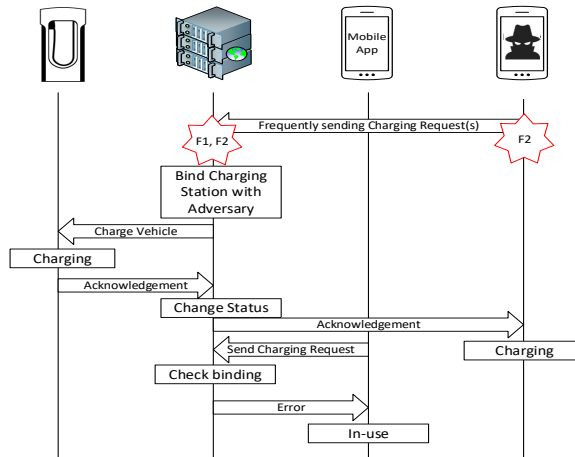


Figure 5.4: Sequence diagram depicting remote charging/discharging scenario.

(e.g., one-time password and email verification). In practice, an attacker could rent service from online SMS and Email providers such as Twilio [116], which provide communication APIs to handle the verification processes. Additionally, attackers can create as many fake email accounts as needed for the verification process.

Remote Charging Session Hijacking. After analyzing the interactions of the mobile application with the different entities, we developed an understanding how the mobile application could be used as an attack surface against the power grid. We found that the studied platforms are vulnerable to session hijacking. By utilizing these vulnerabilities, attackers can initiate unauthorized EV charging sessions with the aim to impact the power grid. Ideally, the CMS should only allow a charging request if the request is issued from the account owner that is bound with the EV. However, we found that the CMS does not perform any account-based authorization or check during charging. In other words, there is a decoupling between the user account and the EV that is connected to the EVCS. The user is coupled with the EVCS ID only. Thus, an EV connected to an EVCS can be charged remotely regardless of whether the user initiating the request is the legitimate owner of the EV. The CMS does not perform the necessary checks to validate whether a user is allowed to perform this action, or if the user is the actual owner of the EV. Thus, allowing

adversarial accounts to unlawfully hijack charging sessions.

As illustrated in Figure 5.4, the adversary can leverage the combination of F1 and F2 to initiate unauthorized charging requests to take control over the charging session that should have been initiated by the actual EV owner. When the attacker uses an adversarial account to start charging requests, the CMS will establish a connection with the adversary. It is worth mentioning that the actions performed by the adversary are fully legitimate and within the scope of the permitted functionalities of the ecosystem. Consequently, legitimate EV owners can no longer control their charging session. The only way for the user to stop the charging is by physically unplugging the EV.

After, the user's charging session has been hijacked, the user can no longer control the EVCS. While this could raise an alert for a security-savvy user, other users may simply disregard this behavior as long as they see that their EV is charging. It is worth mentioning that even the security-savvy users will not notice the attacker's actions unless they regularly check their mobile applications during the charging process. Additionally, even when the attack is noticed by these users, the root of the problem cannot be traced back to the adversary. Only the CMS has the knowledge to trace back the attack's origin. However, due to F1 and F2 described above, the CMS considers the attacker's actions legitimate.

We describe in Table 5.1 the possible attack scenarios based on the functionalities instilled in the mobile application showing that 29 out of 31 mobile applications are vulnerable to remote mass charging attacks. We exclude Tata Power EZ Charge and Electrify America as they require inputting the EVCS ID number that is physically placed on the charging station HMI. Moreover, there are only 19 out of the 31 mobile applications that provide remote start and stop of charging and allow adversaries to launch oscillatory load attacks with their advanced control on the stopping of charging.

Remote Discharging. Vehicle-to-Grid (V2G) capabilities are one of the attractive features of EVs that can one day transform the EV battery into a distributed storage to support

the power grid operation. Willing EV owners would register themselves as users willing to contribute to supporting the grid during peak hours for an incentive (e.g. financial incentives larger than the cost of charging) through utilizing their mobile application. However, as demonstrated above, the current system architecture lacks traceability and end-to-end authentication. An adversary can hijack a session, as explained above, and register themselves as a legitimate user that is willing to contribute to such a V2G scheme. This allows the adversary to gain monetary compensation by discharging other users' vehicles. We acknowledge that such a class of attacks is not feasible at this moment due to the shy adoption of V2G capabilities in the ecosystem and the absence of wide-scale compensation programs. However, the advancement towards such capabilities being instilled in the ecosystem requires improving the current ecosystem architecture and strict access control mechanisms, for safe and secure operation.

5.4.2 Attack Feasibility

In this section, we study the feasibility of launching wide-scale coordinated charging/discharging attacks. In these attacks, we assume that users connect their EVs to the EVCSs before starting a charging session. We also assume that the vehicle remains connected for a period of time after the end of the charging. In the aims to understand user behavior and predict it, the authors in [117] highlight that EV owners do not necessarily start charging right after plugging in. Additionally, a time window exists between plugging in and charging an EV, which is the time a user needs to pull out the phone to start a charging session. Moreover, according to Almeghrebi et al., [118], another time window exists, where customers leave their vehicles for an extended time when parking at the workplace or overnight beyond finishing charging. Some users even leave their vehicles for longer than 24 hours. This attack window is only applicable to mobile applications that provide remote start and stop services. These time windows are exploitable by the adversary that

can utilize them to launch remote session hijacking.

In [119], the authors utilize a multistep hybrid LSTM neural network to predict EVCS occupancy. They base their analysis on public charging data from the City of Dundee, UK in 2018. The number of charging stations plug-in simultaneously during the day fluctuates reaching 300 charging sessions at 10:00 a.m. during the weekend and 400 charging sessions during the weekday. The number of charging sessions starting at peak times during the day is expected to increase as more customers adopt electric vehicles as a means of transportation with the rapid and increased deployment of public charging stations. Thus, the feasibility of remote charging session hijacking at scale increases.

To execute an attack by exploiting these vulnerabilities, an adversary needs information about the user's behavior. By understanding user behavior, the adversary can time and coordinate the attack to increase its success rate. The attacker can extract information, from the mobile applications (Type 1 and Type 2), similar to [120], where the authors predicted user behavior based on arrival time, duration, departure time, etc. An adversary can utilize the online interface (mobile application/web portals) to gather information. The information we gathered through a tool we devised is the start charge time (arrival time), and departure time if a vehicle connected within the attack windows. We used Appium [121] to automate mobile application scraping which can be used for web applications scraping. We then identify target EVCS and monitor their utilization and status. We collect information about the EVCSs that are in use, allowing us to track arrival and departure times. Moreover, whenever the station's status changes from "in-use" to "available", we send a probe charging request to check if there is an EV connected to the EVCS. The collected data is used to model user behavior and allow us and the attacker to target peak EV connectivity hours to hijack charging sessions at scale [119].

```

GET /ocpp/ HTTP/1.1
Host:
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: yc0jOdENekBP7wdb6P1lg==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Protocol: ocpp1.6
User-Agent: Python/3.10 websockets/10.3


HTTP/1.1 101 Switching Protocols
Date: Thu, 08 Dec 2022 18:46:11 GMT
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: SX/d29VizNfPpmpmC4xjyxGHQI=
Sec-WebSocket-Protocol: ocpp1.6

```

Session confirmation

Hi [REDACTED]
Here is the confirmation of your session on the [REDACTED] network.

Transaction Number: [REDACTED]
Transaction Date: 2022-11-11 07:01:41 PM

 Withdrawal on balance

SESSION	
Affiliate network:	[REDACTED]
Site name:	[REDACTED]
Station name:	[REDACTED]
Start date:	2022-11-11 07:01:41 PM
End date:	2022-11-11 10:55:12 PM
Duration:	3h 53m 31s
Energy used:	18.37 kWh
Started with:	Application IOS

Figure 5.5: EVCS device registration with the CMS using our real-time co-simulation test-bed. **Figure 5.6:** Session confirmation showing the success of our attack by hijacking the charging of an idle vehicle.

5.4.3 Attack Demonstration and Verification

In this section, we evaluate and verify our observations, inferences, and conclusions by using a real-time co-simulation test bed. We create a replica of the real EVCS ecosystem by integrating real charging station hardware with a production-grade CMS. The EVCS hardware utilizes OCPP v1.6 and communicates with the CMS backend to perform device registration initially which is described in Chapter 3. The EVCS and the CMS then continuously communicate with each other over WebSockets to ensure that the EVCS is alive by either sending a heartbeat notification or through the WebSockets ping-pong request/response. Indeed, during device registration, the EVCS will send an HTTP request to the CMS backend which gets upgraded to a WebSocket connection. The HTTP header includes the EVCS identification number which could be the serial number or an operator-defined ID as demonstrated in Figure 5.5. Consequently, we were able to confirm and verify our observations and inferences regarding the interactions of the different components discussed previously.

Accordingly, we aim to demonstrate our new attack vector to verify our conclusions on the vulnerabilities that exist in the EVCS mobile operators. To leverage the aforementioned design flaws that rely on the authentication scheme adopted by the different EVCS mobile operators we first identify two EVCSs mainly in heavily populated areas. We monitor these

EVCSs and record their utilization by gathering information from the mobile interface of the applications for 3 days starting the 8th of November 2022 till the 10th of November 2022. During that time we performed reconnaissance to understand the utilization of the EVCS. EVCS₁ showed heavy arrival during evening hours (between 6:30 PM and 7:30 PM) whereas EVCS₂ showed heavy arrival between 4:00 PM and 5:00 PM. We note that to identify the utilization and arrival we note the change in the state of the EVCS. When charging an EV, the EVCS shows an “in-use” state rendering it unavailable to other users. Consequently, on the 11th of November 2022 we execute our attack as a proof of concept on the EVCSs. We leverage the lack of rate limiting to send multiple charging requests every 3-4 minutes from the adversarial account we created using the legitimate mobile application channels. After sustaining the attack for almost 30 minutes starting at 6:30 PM for EVCS₁ and starting at 4:00 PM for EVCS₂. Consequently, at 7:01 PM and at 4:18 PM we were able to successfully hijack the charging session of the EVCS₁ and EVCS₂ respectively. Consequently, we show in Figure 5.6 the confirmation of a successful charging of a vehicle that does not belong to us for almost 3 hours. Moreover, we would like to note that the attack was demonstrated and verified on two different mobile applications, i.e., one that only allows a start charge functionality and another that allows remote start and stops charging functionality. We also note that vulnerabilities that allow the adversary to remotely start and stop a session could be used in combination with other High-wattage IoTs to impact the power grid. In [16], the authors demonstrate the impact of controlling a large botnet swarm of high-wattage IoTs that could be used to impact the power grid by launching load-altering attacks. Load-altering attacks impact the power grid by inducing grid instability (e.g., frequency instability). Finally, the same attack workflow could be launched to impact the power grid using the other mobile applications as they follow the same procedures and policies to authenticate users.

5.5 Attack Implications

In what follows, we discuss the remote charging session hijacking attack scenario along with its implications on the power grid and EV users, respectively.

5.5.1 Attack Implications on the Power Grid

As demonstrated in the analysis, attackers can leverage the identified vulnerabilities to compromise user accounts and perform synchronized large-scale cyber-attacks against the integrated infrastructure [22, 113, 122, 123]. With the EV charging ecosystem being a new and wide attack surface, it is an attractive target for exploitation by organizations with enough resources to conduct large-scale attacks against the power grid, by utilizing the mobile application to perform stealthy attacks.

Consequently, an adversary could initiate a distributed botnet attack utilizing thousands of malicious accounts to send charging requests and hijack as many sessions as possible to amplify the attack. The behavior of arrival and departure at charging stations almost coincides with the demand behavior of the power grid [120], as demonstrated in the utility demand curve of California, United States of America [124] and New South Wales (NSW), Australia [125]. Additionally, by exploiting the identified vulnerabilities and initiating the remote charging session hijacking attack (Section 5.4.1) during the described attack windows (Section 5.4.2), an adversary can remotely orchestrate hijacked charging sessions to synchronize a wide scale attack that can disrupt the power grid operations.

In this section, we study the impact of synchronized mass charging attacks on power system economics (i.e., generation cost and transmission line losses). We then examine how adversaries with some knowledge of the grid topology can craft targeted mass charging attacks in order to overload and trip transmission lines. Finally, we study the power grid stability subject to oscillatory load attacks that can cause violation of the safe frequency operation limits and load shedding. Oscillatory load attacks can be performed using 16 of

the applications that provide on-and-off remote control capabilities without requiring the user to scan a QR code.

To amplify the attack impact on the grid, an adversary with knowledge of the grid can craft targeted and smarter attacks. A small number of compromised charging sessions with enough knowledge of weak buses allow the adversary to disrupt the power grid operations. Power grid information can be estimated through monitoring the measurements of the power grid to estimate the topology, using MILP programming, machine learning, and voltage and load monitoring [17, 126, 127, 128, 129, 130, 131, 132]. Various stability techniques and strategies could then be used by adversaries to locate the most sensitive/vulnerable buses, such as PV and QV curves [17, 133].

We demonstrate the impact of the attacks on the 7-bus test case introduced by Glover et al. [134] (Figure 5.7a), which is commonly used for research purposes [17]. We utilize this grid due to its built-in optimal power dispatching capabilities, unlike the work in [113]. Moreover, this 7-bus test case provides the generation costs formulas that will allow us to study the economic impacts on the utility. To achieve a close to realistic simulation of the power grid behavior during peak and off-peak demand hours, we scaled the grid loads based on the NSW [125] power grid load profile using Equation 1.

$$HourlyLoad_{scaled} = \frac{GloverLoad \times HourlyLoad_{NSW}}{AverageLoad_{NSW}} \quad (1)$$

To this end, we use PowerWorld [135] which is a power simulator that allows us to analyze the steady-state power flow, transient stability, generation costs, and other power system operations. Unlike [16, 17], here we study the economical aspects of an attack such as generation cost and line losses respectively. Along with that, we also take into account the load shedding mechanism that is used by the utility to regulate power generation in case of a sudden drop in frequency below certain thresholds [136] to demonstrate more realistic attack implications. The different attack simulations and results are demonstrated below. In

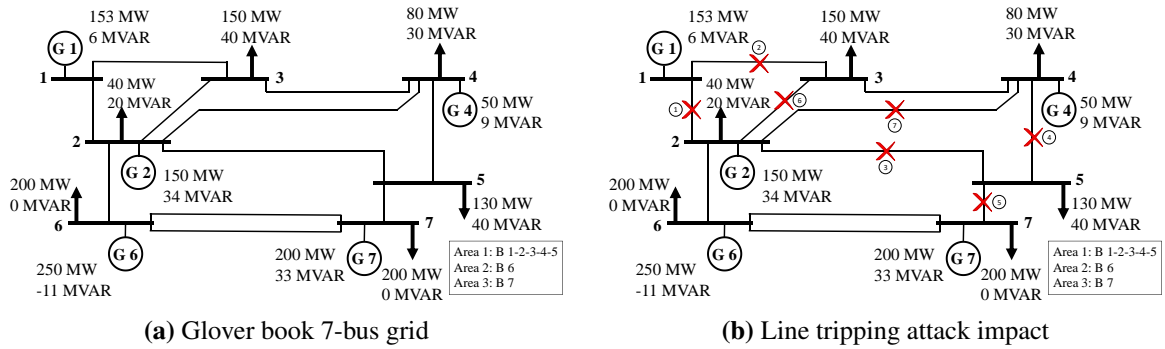


Figure 5.7: Overview of the (a) Glover book 7-bus grid and (b) the impact of the line tripping attack scenario.

what follows, the attack is initiated by compromising 84 MW of EV load that is equivalent to 7636 EVs charging at the 11 kW Level 2 chargers. The current numbers of EVCSs and EVs are not enough to mount such attacks, however, the growth in the EV numbers will soon provide a large enough surface to make it possible [17]. It is worth noting that mobile applications allow cross-product communication and control, thus, increasing the scale of the attack as more vendors join these platforms. Moreover, as the EVCS market move towards wide adoption of level 3 chargers, the higher power entails higher risk.

Economical impact. The attacker can cause the power utility to incur economic losses by launching EV attacks against the power grid. To study the economic impacts of a mass charging attack on the grid, we examine the transmission line losses and the power generation cost during different loading conditions and under different attack scenarios. To perform mass-charging attacks 30 applications allow us to perform such an attack, whereas the rest prevent remote mass-charging by forcing the adversary to scan a QR. We used the scaled load profile to demonstrate the incurred cost and losses at different grid loading conditions. Namely, we focused on the peak load (943 MW), the average load (800 MW), and the minimum load (677 MW) conditions that we will refer to later as off-peak load. We simulate 3 different attack scenarios against the test grid by (1) distributing the attack load randomly, (2) distributing the attack load equally among the 6 load buses and (3) distributing the attack load proportionally among the different load buses. It is worth

highlighting that Scenario (1) represents a random distribution of the EV charging attack load to simulate an adversary with no knowledge of the grid topology. Scenarios (2) and (3) represent attacks by an adversary with limited knowledge of the grid and geographical knowledge of load size and EV distribution respectively.

Generally speaking, the attacks will increase the transmission losses under all loading conditions. However, under higher loading conditions, the same attack will cause more incremental losses due to the increased power flow in the lines. Line losses are calculated as $P_{loss} = R_{line} \times I^2$ thus at higher loading conditions, the same attack load will result in more losses. It is worth noting that under the different attack scenarios, the total incremental line losses were almost equal. This is due to the fact that, the total load of the different attack scenarios is the same and that we do not have any extra long transmission lines that will have significantly different losses under different attack load distributions. The normal and incremental losses are demonstrated in Figure 5.8a. The no-attack losses under the different loading conditions were 3.1 MW (off-peak), 3.4 MW (average), and 4.3 MW (peak), which lead to an increase of 16.13% at off-peak loading conditions, 17.65% during average loading conditions and 18.6% during peak conditions. Thus, this simulation clearly demonstrates that the attack impact on system losses is amplified when the power demand was the highest, which also coincides with the time during which EV connection to the chargers is the highest.

In the case of attacks against generation cost, each attack scenario differs based on the optimal power dispatch performed by the utility to reduce the overall cost. Figure 5.8b presents the total generation cost of the system when no attack occurs and during the three attack scenarios mentioned above. As Figure 5.8b demonstrates, the total added cost due to the attack is higher during peak loading conditions across all attacks. More importantly, the proportional attack scenario caused the highest extra cost. To put things into context, the no-attack cost at off-peak, average, and peak loading conditions were

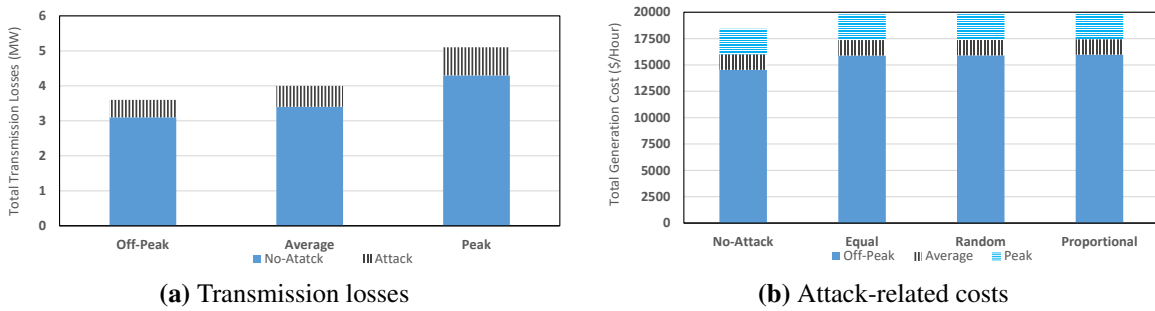


Figure 5.8: Incurred (a) transmission losses and (b) costs due to various attack scenarios.

\$14,545.28/Hour, \$16,009.39/Hour, and \$18,438.10/Hour respectively. The added cost due to the proportional attack is \$1,423.83/Hour under off-peak conditions, \$1,426.45/Hour under average conditions, and \$1,451.95/Hour under peak conditions. This demonstrates how an attacker can force the utility to increase its generation and incur extra costs.

One aspect not present in the simulation was the usage of peak generation units. This was left out due to the absence of these units in the Glover grid in Figure 5.7a. These units are usually fast-ramping units used by utilities and power grid operators during peak hours when the large baseline generation units do not have sufficient capacity to supply all the load. These peak generation units are usually operated for a few hours a day only due to their high operation cost. This means that if the attack occurs at a time when peak generators are being utilized, the extra cost would be higher. Another aspect of repeated long-term attack worth mentioning, is that mass charging attacks, especially at peak hours, will cause extra transformer loading. This extra loading would reduce its lifetime and would require more frequent maintenance intervals causing extra maintenance costs.

An attacker with a long-term target of causing the utility to incur extreme losses can repeat the hijacking of charging sessions over long periods of time. For instance, launching the above attack for one hour during peak times every day for an entire year will create an extra generation cost totaling \$529,962 for the utility based on the Glover grid and the generators' cost functions. To put things into a better context, scaling this attack up to the NSW grid will cause \$4,615,967 extra cost for the utility per year. To this end, an

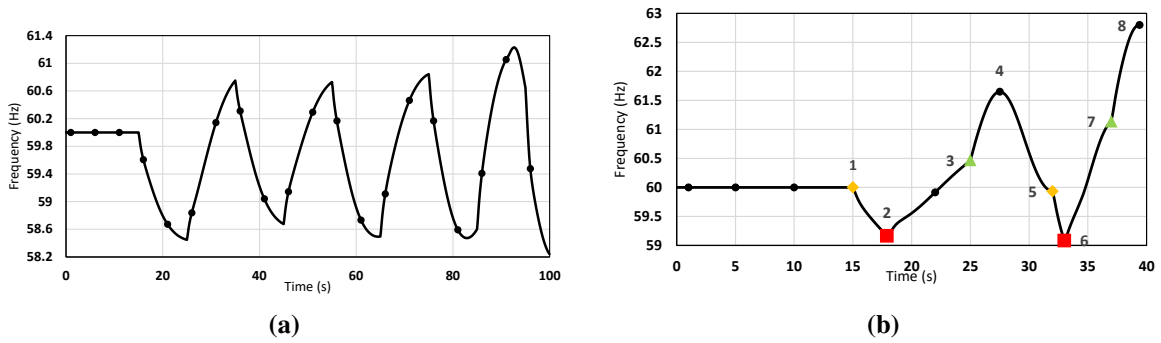


Figure 5.9: Frequency behavior over time (a) without load shedding, and (b) with load shedding.

attacker might choose to compromise a smaller number of EV charging sessions and choose different sets of EVs every day to remain stealthier and still cause millions of dollars of losses to the utility in extra generation costs.

Overloading and tripping transmission lines. Another type of impact that might be desired by the attacker is causing line overloading and tripping by crafting a targeted attack against the grid. This attack has more severe and immediate consequences since it can leave consumers without electricity. In the previous set of attacks, some lines got highly loaded but none of them reached an overloaded state. The same EV load however can be used by attackers with topology knowledge to target certain lines in order to cause cascading line failure. The attacker will only require knowledge of the topology and estimate values of the loads and power flows but not the line parameters. This information can be found online and in multiple public access databases and websites.

To simulate such attacker behavior, we targeted bus 4 and bus 5 with a synchronized 20 MW and 64 MW EV charging attack respectively. This attack overloaded and tripped the line connecting bus 1 and bus 2 after which multiple lines would be overloaded and tripped. In total, seven lines would trip successively in the order shown in Figure 5.7b. The successive line tripping would lead to islanding each of buses 1, 3, 4, and 5. While the load at bus 4 will be supplied by power from the generator at the same bus, the loads at bus 3 and 5 will lose their power supply and thus the grid will lose a total of 280 MW which

Table 5.2: Attack Scenario description and impact.

#	Time(s)	System State	Action	Action By	Impact (state change)
1	15	System is operating normally	Total attack load of 40 MW initiated	Attacker	System frequency starts dropping
2	17.9	Frequency drops below 59.3	5% load shedding	Utility	5% of total consumers lose electricity. System frequency starts rising
3	25	The frequency peaks and is regulated by the automatic generator control	Turning off all compromised charging sessions	Attacker	System frequency spikes
4	27.5	The frequency starts dropping due to the automatic generation control	Automatic action of generation control system “no human intervention”	Automatic	System frequency is being reduced to stabilize the system
5	32	The frequency was reduced by the automatic generation control	Total attack load of 80 MW initiated	Attacker	System frequency starts dropping faster than step 1 due to the larger attack load and the reduced generation after load shedding
6	33	Frequency drops below 59.3	5% load shedding	Utility	Additional 5% of total consumers lose electricity (10% total). System frequency starts rising
7	37	The frequency peaks and is regulated by the automatic generator control	Turning off all compromised charging sessions	Attacker	Causes a larger spike in frequency than step 3 since the EV load that was turned off is larger than that of step 3
8	39.4	Frequency exceeds 61.8 Hz [136]	Generators should be tripped instantaneously	Utility	Sequential generator tripping until system frequency stabilizes.
>8	>39.4	Utility trips generators immediately. The system inertia drops.	The attack impact is larger causing more tripping.	Attacker	As more generators are tripped, the system reaches a state of blackout.

represents a loss of electricity to 35% of the consumers.

Power grid instability. Another attack that takes advantage of load manipulation is an oscillatory load attack that can impact the frequency stability of the power grid. This attack revolves around the concept of creating a demand surge to cause a frequency drop on the grid followed by a drop in demand to cause the frequency to overshoot. In the first step, the attacker will use the compromised accounts and hijack charging sessions to initiate mass charging to increase the power load. This extra power load would create an imbalance between the increased load and the generated power causing the generators to slow down resulting in a frequency drop. The second step of this attack happens when the system starts its recovery. The attacker would switch off the compromised charging sessions, initiated in the first step, to cause a frequency increase that is amplified by the operator’s effort to increase the speed in response to the initial step. The attacker would then alternate between

these steps for the desired duration. The impact can be amplified by launching the attack when the system has lower inertia due to the presence of a high share of renewable energy resources.

Given the dependence of the grid's transient behavior on the generator and turbine models, we utilized the automatic control models common to studies similar to ours. It is important to note that the utilization of different control models will change the exact values of the simulation but the general shape and behavior remain the same. This demonstrates that the attacks can be successful under different conditions, but their magnitudes might need to be scaled based on the different conditions to achieve the desired impact. In our study, we used the machine model "GENSAL", the exciter model "IEEE T1" and the turbine governor model "IEEE G2".

The oscillatory load attack is simulated against the grid in Figure 5.7a by initiating the oscillatory load behavior described above on buses 3 and 5. The attack is initiated by starting a mass charging session equivalent to 20 MW (at time $t=15s$ and stopping it at $t=25s$ after the system starts to increase generator speed to compensate and the frequency starts to rise. This charging and stopping behavior are repeated periodically every 10 seconds while increasing the attack load at each bus by 5.5 MW every cycle. The frequency behavior of the grid that results after such an attack is demonstrated in Figure 5.9a where we can see the frequency fluctuation due to the oscillatory load behavior. The importance of this attack is that it does not require huge loads to cause the frequency fluctuations depicted in Figure 5.9a. Even when the compromised EV numbers are much less than in the example above, a sustained oscillation will hinder the system's return to normal operation. Sustaining this attack would damage the turbines due to the constant acceleration and deceleration.

In the previous example, we assumed no grid protection mechanism was used by the utility and that the attacker followed a semi-naive approach in which the attack period is predetermined and does not change as a result of actual conditions on the grid. In this

iteration of the attack, we assume the utility will utilize load shedding when the frequency drops below preset thresholds. The threshold that is violated in the attack is the 59.3 Hz threshold after which the utility will immediately disconnect 5% of the total load in order to compensate for the fast dropping frequency [136]. This utility behavior is depicted in Figure 5.9b by the squares at time $t=17.9s$ and $t=33s$. The behavior depicted in Figure 5.9b is a response to a more advanced oscillatory load attack requiring the attacker to know and observe the actual grid response to tune the attack load and period. The attacker and utility interaction at every step is summarized in Table 5.2.

An extension of the oscillatory load attack can be achieved by utilizing the reverse power discharge through the V2G functionality similar to a work performed in [3]. By initiating this V2G at the instance we stop the mass charging, the attacker can cause a larger frequency spike. It is worth mentioning, that by instilling V2G capabilities in mobile applications the adversary can then utilize it to increase the oscillatory attack effect on the grid.

Chapter 6

Uncovering Covert Attacks on EV Charging Infrastructure: How OCPP Backend Vulnerabilities Could Compromise Your System

6.1 Threat Model and Analysis Methodology

In this paper, we provide a security analysis framework to assess the OCPP backends of multiple operators and to discover their vulnerabilities. Unlike previous works, the precursors of our attack do not rely on a set of assumed and unverified vulnerabilities. In what follows, we detail the adversarial objectives, required assets, and adversarial capabilities. We also present the required precursors of the attack, which are attainable from online sources by the attacker.

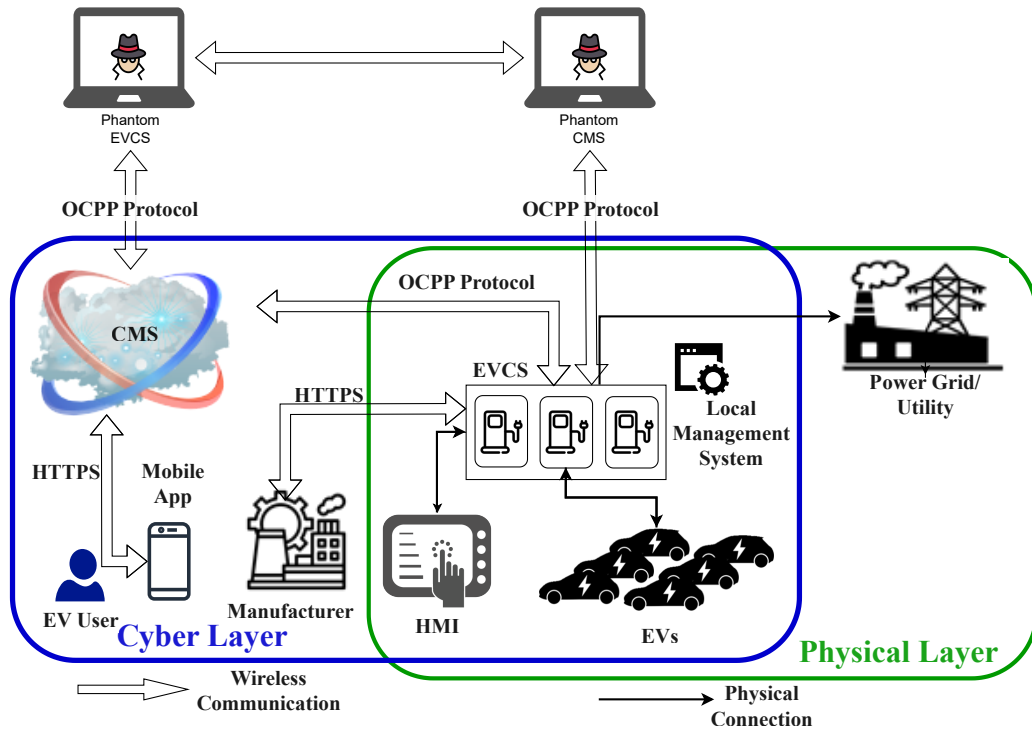


Figure 6.1: An overview of the EV charging ecosystem’s cyber/physical layers. The phantom CMS is added only to show a unique and advanced attack scenario (AS5).

6.1.1 Threat Model

An adversary can utilize reverse engineering methods and penetration testing techniques along with the flexibility provided by the EVCS operators to study the communication between a legitimate EVCS and the OCPP backend on the charging management system. The objective of the adversary is twofold: (i) monitor and collect the communicated messages between the OCPP backend and the EVCSs to acquire sensitive information such as the device identity information and OCPP backend links; and (ii) implement a phantom EVCS and a phantom CMS using the acquired artifacts to substitute the legitimate EVCS and CMS (Figure 6.1) while hijacking the communication between the CMS and the actual EVCS within the ecosystem. By doing so, the adversary aims to fabricate and/or manipulate the exchanged messages to impede the CMS visibility over its infrastructure. The adversary also aims to hijack and substitute legitimate EVCSs

to monitor ongoing communications. These capabilities can be also used to cause a DoS that impacts the network of EVCSs and the CMS.

Note that the different components of the ecosystem are considered trustworthy, and non-malicious. Thus, the adversary does not need to infiltrate other components. Instead, the adversary only focuses on interfering with the communication process with the OCPP backend. Therefore, the main objective of the adversary is to use the security weaknesses of the OCPP backend to substitute the legitimate EVCS with a phantom one while hijacking the communication between the CMS and the actual EVCS within the ecosystem. Nevertheless, to achieve these objectives, the adversary must perform further operations, detailed in the following subsections:

Reconnaissance and Feasibility

The adversary needs to collect the required information to prepare for the intended attack. This information includes the EVCS IDs and the OCPP backend link, which is used by the numerous EVCSs to communicate with the CMS remotely. Such information could be obtained from various online resources. For instance, an adversary can utilize one of the numerous online EV charging maps provided by operators to get the IDs of the deployed EVCSs. Moreover, one can purchase a private EVCS and link it to an operator that supports the deployment of private charging stations within their network. Note that these operators often allow private EVCSs to connect to their backends to become visible as part of the operator's public charging network. To enable this feature, the operator will provide the private EV owner with the OCPP backend link and assigns an ID to their EVCS. The EVCS IDs can be either assigned/generated by the operator (e.g., randomly generated ID) or obtained from the device itself (e.g., serial numbers) [137]. Additionally, the adversary can collect such information by leveraging online device search engines and/or a series of penetration testing techniques, which have been verified to be feasible in previous work:

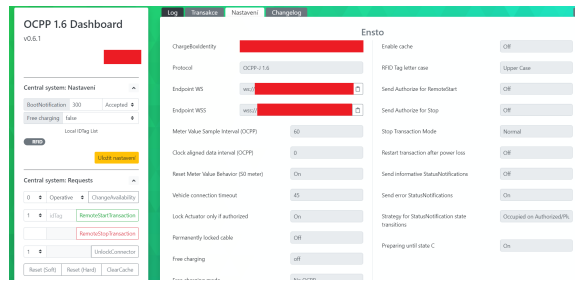


Figure 6.2: OCPP backend discovered on Zoomeye [1].

- Discover OCPP backends using online device search engines such as Zoomeye [1], Shodan [33], and Censys [138]. As demonstrated in Figure 6.2, the adversary can perform a simple lookup into these repositories using EVCS-related keywords (e.g., “OCPP backend”), to obtain information about deployed CMS and their OCPP backends.
- Utilize online device discovery techniques along with security analysis methodologies to identify severe vulnerabilities (e.g., XSS, SQL Injection, and CSRF), which enable hacking into the deployed EVCSs and their management systems [17, 19, 35]. Once the EVCS is compromised, the adversary will be able to extract the EVCS ID and the OCPP backend link.
- Use the device search engines to discover EVCSs that are configurable over the internet (e.g., with no authentication, or with weak/default credentials). The adversary can discover EVCSs in the wild and then exploit the usage of default credentials to configure the EVCS, as discovered in [23, 50]. Once logged in, the adversary can learn the ID and OCPP backend link (Figure 6.3).

It is worth noting that EVCS IDs can have different types/forms. For instance, some of these IDs represent the EVCS’s serial numbers, MAC addresses, or randomly generated numbers that are assigned by operators to the deployed devices [137]. Therefore, some of these IDs are guessable and could be brute-forced. Consequently, upon utilizing any of the above-mentioned methods, the adversary could access the OCPP backend link and obtain

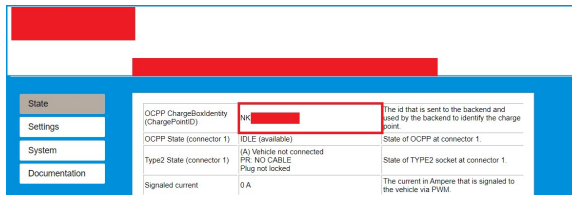


Figure 6.3: EVCS discovered using Zoomeye [1] that publicly exposes the critical information, namely the EVCS ID.

the EVCS IDs for a given operator. For example, in Figure 6.3, we illustrate an example of a discoverable EVCS using an online device search engine (Zoomeye [1]). After inspecting the discovered host, we were able to obtain all its information, including its EVCS ID, since the operator intentionally provided this information on the EVCS’s web interface.

EVCS Commissioning Breakdown

To discover the OCPP backend vulnerabilities, a testbed is used to identify the vulnerabilities and help us create Proof of Concepts (PoC)s that were later used on the 16 live systems to prove the existence of the same vulnerabilities [Anonymized]. The testbed enables us to perform an in-depth analysis while achieving a realistic understanding of the system without performing any intrusive actions on the real-life deployed systems. To achieve a realistic implementation, the testbed was validated in terms of the OCPP client and server implementations with production-grade EVCSs and CMS backends. Thus, making our testbed implementation a replica of production systems. The testbed implementation mimics the deployment of 16 live operators worldwide and takes into consideration TLS and authentication mechanisms. The testbed is described in Chapter 3. However, through our work, we show that EVCS operators need to raise the security bar by using more rigorous authentication and authorization mechanisms. Moreover, the testbed is used to identify the vulnerabilities and help us create PoC that were later used on the 16 live systems to prove the existence of the same vulnerabilities. We leverage the implemented testbed to analyze the EVCS commissioning and registration procedure when adding a new device to the

operator's OCPP backend. It is worth highlighting that studying the commissioning procedure of the EVCS allows us to understand the authentication mechanism and the workflow that is implemented by the backend systems. As shown in Figure 6.4, the commissioning procedure for any EVCS consists of the following main steps:

- i. Whenever a new EVCS is connected to the network, the responsible personnel (e.g., technician) for commissioning the EVCS will use the assigned EVCS ID to register the device on the operator's CMS and establish an OCPP backend link.
- ii. The operator will establish the OCPP backend link, which is an independent WebSocket endpoint that is always listening for connections from the connected EVCSs. In WebSockets, the client initiates the communication by sending an HTTP request that gets upgraded into WebSocket communication.
- iii. The user will input the OCPP backend link into the configuration page, which can be accessed remotely or locally.
- iv. At this stage, the EVCS will reboot automatically to reconfigure itself to the new settings. During this brief time, the EVCS will be offline as it tries to configure itself and establish communication with the CMS.
- v. The EVCS returns to its idle state and initiates the boot sequence where the EVCS communicates with the CMS aiming to establish a WebSocket communication.
- vi. The EVCS will start a TCP handshake with the OCPP backend.
- vii. A WebSocket handshake is then initiated by upgrading the HTTP handshake into a full-duplex communication. This is because WebSocket communication is based on a single TCP connection, whereas HTTP is half-duplex.

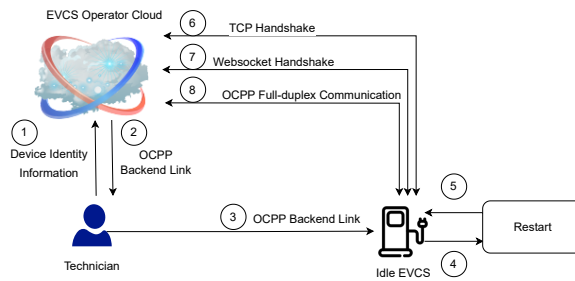


Figure 6.4: EVCS commissioning lifecycle.

viii. At the final step, an OCPP full-duplex communication is established using WebSockets technology. This enables seamless bi-directional communication between the EVCS and the CMS.

Note that beyond Step vii, the EVCS will follow the OCPP boot sequence, which incorporates a request-response mechanism where the EVCS sends a `BootNotification` request that contains information about the EVCS such as `IMSI/ICCID` values, `Model`, and `SerialNumber` information, to name some. Consequently, the CMS replies with a `BootNotification` confirmation. Moreover, the CMS then sends a `GetConfiguration` request to retrieve the configuration items defined by the OCPP such as authorizing remote transactions and number of connectors. Then, the EVCS remains idle (ready to charge), while waiting for incoming requests from the CMS or manual demands for charging by EV users.

As shown in Figure 6.5, the content of a captured packet from the exchanged traffic between an EVCS and the CMS within our testbed reveals sensitive information such as the device registration information (EVCS ID) and the OCPP backend link (anonymized in the Figure). This sensitive information is necessary for creating the connection between the client EVCS and the OCPP backend on CMS server. To generalize our findings in terms of the EVCS commissioning procedure across different operators and EV charging stations, we examined sixteen widely deployed CMS, which are operated by different vendors. Our analysis indicates that the described EVCS commissioning and registration procedures are

```
GET /ocpp/AL0012***** HTTP/1.1
Host: *****
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: yc00jOdENekBP7wdb6P1lg==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Protocol: ocpp1.6
User-Agent: Python/3.10 websockets/10.3

HTTP/1.1 101 Switching Protocols
Date: Thu, 08 Dec 2022 18:46:11 GMT
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: SX/d29VizNfPpmpmC4xjyxGHQI=
Sec-WebSocket-Protocol: ocpp1.6
```

Figure 6.5: HTTP and WebSocket Handshake.

common across various deployments of the EV charging ecosystem. Additionally, the device commissioning procedure is backend-specific and does not rely on the EVCS that is connected to the OCPP backend.

6.1.2 OCPP Backend Vulnerabilities

To this end, we inspect the vulnerabilities of the OCPP backend against the OWASP Top 10 security risks [139]. We use our implemented testbed to investigate vulnerabilities associated with injection attacks, along with access control and authentication weaknesses. Given the identified weaknesses, we create a list of PoC exploits to validate the vulnerabilities and test them on the analyzed CMS that is supported by 16 different operators/vendors. The list of used PoC exploits is available in the following GitHub repository [Anonymized]. In what follows, we provide further details about the two examined classes of security weaknesses, which allow the adversary to compromise the OCPP backend:

Injection Vulnerabilities

We analyze the OCPP communication through disassembly and the breakdown of the communication flow. We focus on finding entry points that lack input cleansing and validation. Thus, to find insertion points (e.g., GET/POST parameters), we perform system

fuzzing on the WebSocket handshake using various crafted payloads by intercepting HTTP request/response traffic using Burpsuite [88]. The payloads include but are not limited to, remote code execution, SQL Injection, and directory traversal, to name a few. The payloads are used with our PoCs to analyze the systems and infer whether they incorrectly interpreting the used payloads. Note that the PoCs will not inflict any harm to the tested systems and can only be used to indicate if the analyzed system is exploitable or not. The payloads of the PoCs can be found in the following GitHub repository [140]. Additionally, we provide an attack script in our GitHub repository [Anonymized], which can be used to reproduce the tests and confirm the identified vulnerabilities. However, we do not provide any details of how to extend this exploit to launch any of the proposed attack scenarios.

Note that the input parameters can be modified to perform further testing. For instance, as shown in Figure 6.3, we modified the EVCS identifier to inject a payload and confirm that the OCPP backend logic does not accept manipulated input parameters. Additionally, we manipulated several OCPP packets such as the BootNotification and sent them to the backend to test the security of the backend against OCPP manipulation. Consequently, we tested other OCPP packets sent by the EVCS, which include parameters such as the response to GetDiagnostics and GetConfiguration OCPP requests. Our analysis verified that the tested OCPP backends utilize strict input validation and sanitization mechanisms that prohibit an adversary from hijacking the execution flow by forcing the OCPP backend to interpret malicious payloads. Thus, we confirm that the deployed security mechanisms by the OCPP backend will protect the EV ecosystem against the aforementioned injection attacks.

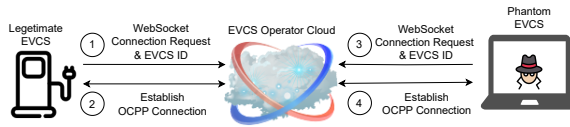


Figure 6.6: Access control and authentication vulnerability analysis.

Access Control and Authentication Vulnerabilities

Access control and authentication failures have been ranked first and seventh in the OWASP Top 10 security risks [139]. Moreover, these security risks will allow the adversary to exploit the system and set foot into the OCPP backend. The analysis of the OCPP backend in our testbed shows that the authentication mechanism is based on the value of the registered EVCS ID, which is sent as part of the Get parameter. Given that such information is discoverable (as described in Section 6.1.1), we develop a phantom EVCS to imitate the functionalities of the OCPP Client that is found on the EVCS. We illustrate the setup for our access control testing mechanism in Figure 6.6. We create a generic phantom EVCS, which can be used to test any OCPP backend without the need for further customization. The phantom EVCS was implemented in Python [77], and by following the standard documentation that can be found on our anonymized GitHub repository [Anonymized]. Moreover, the phantom EVCS imitates a legitimate charging station by exchanging OCPP messages including `HeartBeat` notifications that are used to ensure that the client is always alive. Finally, the phantom EVCS is running OCPP v1.6, which is the most widely deployed version of OCPP by EVCS operators worldwide.

Throughout our analysis, the objective of the adversary is to create a phantom EVCS that can connect to the OCPP backend to obtain the OCPP backend link and pose as a legitimate charging station in the network. This can be done following two assumptions: (i) by using the ID of a registered charging station to create a phantom EVCS, which replaces the legitimate EVCS by hijacking the OCPP backend link; and (ii) by utilizing other non-registered EVCS IDs to establish a connection with the backend. Ideally, the OCPP

```
GET /ocpp/AL0012***** HTTP/1.1
Host: *****
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: Y0uV9twZ1VSMufPeKuuFIA==
Sec-WebSocket-Version: 13
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Protocol: ocpp1.6
User-Agent: Python/3.10 websockets/10.3

HTTP/1.1 101 Switching Protocols
Date: Fri, 11 Nov 2022 23:23:34 GMT
Connection: upgrade
Upgrade: websocket
Sec-WebSocket-Accept: pMY2xx7ja0ibXDSiMuWMCofJTQA=
Sec-WebSocket-Protocol: ocpp1.6

.....x...Z.....A.....A...U...A.....O.....6.....Z.....Z.....
.....%x[3,"ce6f83e9-*****
*****{"currentTime":"2022-11-11T23:23:34.828Z","interval":
180,"status":"Accepted"}].H(2,dce7ebf0-*****
*****,"GetConfiguration",{"key":[]}).....Y.*.f.c.g`./4.5.6.g.+.;7.d?.4.3.c?.
).a.h.l.s.c.o.l.c.).c.$.$.g.b.l..8.g.q.*.t.j.g.<.j.
```

Figure 6.7: TCP flow of OCPP backend acceptance of the connection from the phantom EVCS (test date: Nov. 11, 2022).

backend should refuse to connect to the phantom EVCS in these two cases. However, our analysis results indicate that the OCPP backend can only detect/block communication with phantom devices that utilize non-registered IDs. While on the other hand, a registered EVCS ID can be used with a phantom device to replace legitimate charging stations and hijack the OCPP backend link successfully, as shown in Figure 6.7. Moreover, we discovered that the CMS does not limit the number of connections that are made from different devices that use the same EVCS ID. Therefore, we were able to create more than one phantom EVCS at a time (e.g., tested with up to 10 devices) while establishing a successful connection to the OCPP backend. Finally, we verified these findings by testing the vulnerability on 16 real-life OCPP backend deployments. Note that we refrain from mentioning the operators’ names due to the sensitivity of the services provided by them and the fact that revealing such vulnerabilities might impact the security of their operation on the power grid.

6.1.3 Ethical Consideration and Responsible Disclosure

OCPP Backend Discovery, Analysis, and Testing

To discover OCPP backends, we relied on passive Internet scanning data previously collected by third-party device search engines (e.g., Shodan [33] and Zoomeye [1]). No active scans have been done. We note that the discovered vulnerabilities on the live systems did not impact their performance. We discovered them through also coordinated testing/staging environments provided by some of the operators. We also validated with the operators that the PoCs do not cause harm to the analyzed systems while monitoring the tested assets. Additionally, we did not observe any damaging outcome from testing the PoCs with the operators that provided testing environments. We note that the PoCs contain passive payloads that do not exploit the vulnerabilities and in the worst-case scenario any unexpected side effects. Additionally, we extend previous work to implement new tests using the created co-simulation testbed [141]. Their implementation has been validated and verified to be a replica of a production-grade environment. Thus, we performed some tests on it to prevent impacting real systems.

Responsible Disclosure

As part of our ethical consideration, the vulnerabilities were documented and disclosed to the affected parties prior to the publication of our findings. The presented discoveries are dated back to November 11, 2022. Thus, the vendors/developers were provided with more than 7 months to validate these vulnerabilities before publishing the results. Among the 16 EVCS vendors/operators, only two have acknowledged the discovered vulnerabilities while asking to remain anonymous in our reporting. Additionally, while the vendors verified the identified vulnerabilities, they refrained from assigning CVEs due to the sensitivity of the environment and the fact that publishing such results would expose their live infrastructures to possible cyber-attacks. Note that the remaining 14 vendors/operators have not responded

to us to date. We initiated a coordinated vulnerability disclosure with impacted operators and notified them about the attack vector, feasibility, and impact. The notification campaigns were initiated in November 2022 and continued till July 2023. In the notification, we also provided the operators with mitigation mechanisms that would limit the attackers' ability.

Data Privacy

We obtained an approved institutional review board (IRB) based on data retention/-management policy regarding the gathered data and hosts. Specifically, we retained data collected from the OCPP Backends for the duration of the analysis, after which, to preserve the privacy of data and reliability, we removed from our machines, all data gathered during the study of the affected host instances.

6.2 Analysis Results

In this section, we discuss how the vulnerabilities can be leveraged to perform different attacks. This is the first work that discovers vulnerabilities in the EV charging ecosystem that can be leveraged to perform wide-scale attacks against operators' cloud systems rather than individual EVCSs. Consequently, we discuss the attack workflows in detail and present the different attack scenarios that leverage such vulnerabilities.

6.2.1 OCPP Backend Vulnerabilities Analysis

As described in Section [6.1.2](#), our analysis indicates that an adversary can implement one or more phantom EVCS using registered charging station IDs and use them to hijack the OCPP backend links to impersonate legitimate EVCS. Indeed, our analysis results demonstrate the lack of proper security measures to protect the OCPP backend, which

implies a series of security weaknesses and vulnerabilities. As shown in Table 6.1, we identified vulnerabilities showing their related CWEs [142]. CWEs provide a systematic and standardized way for classifying all software weaknesses and vulnerabilities [143]. In what follows, we list and describe the 6 zero-day vulnerabilities we discovered. It is worth highlighting that a zero-day vulnerability is a security flaw discovered before the vendor/-operator rectifies the issue after becoming aware of its existence [144].

- **Improper and Weak Authentication:** by leveraging the phantom EVCS and successfully connecting to the OCPP backend we discover improper authentication where the phantom EVCS claimed the identity of a legitimate EVCS and the backend did not prove that the claim is incorrect. However, we note that the tested OCPP backends utilize an authentication mechanism based on the EVCS ID. This means that only phantom EVCSs using registered IDs will be able to connect to the backend thus, partially limiting the breach into the ecosystem. However, once a legitimate EVCS ID is exploited by our phantom EVCS, the authentication mechanism fails to prove the identity of the client.
- **Improper Restriction of Communication Channel to Intended End Points:** the OCPP backends establish communication channels for phantom EVCSs and do not properly ensure it is interacting with the correct client. The phantom EVCS is able to hijack and substitute a legitimate EVCS and gain the same level of access.
- **Improper Control of Interaction Frequency:** the OCPP backend does not control the number of parallel connections that could be made from the phantom EVCSs possessing the same source IP utilizing the same legitimate EVCS IDs. To test that we connect 10 phantom EVCSs in parallel without being restricted.
- **Improper Restriction of Excessive Authentication Attempts:** this is closely related to the previous CWE, however, in this case, we utilize an unregistered EVCS ID. The OCPP backend did not attempt to block our connections making it susceptible to EVCS ID

Table 6.1: Vulnerabilities discovered in each of the 16 live CMS operators’ backends.

Vulnerability	CWE	Specific EV Ecosystem Vulnerability
Improper and Weak Authentication	284/287	When a phantom EVCS claims to have a given identity the backend does not sufficiently verify and validate that.
Improper Restriction of Communication Channel to Intended Endpoints	923	Attackers can spoof the real EVCS using a phantom EVCS, thus gaining the same level of access as the real EVCS and continuing to communicate without validating the identity ever again.
Improper Control of Interaction Frequency	799	The backend does not limit the interaction with the EVCS which allows a phantom EVCS/real EVCS to send excessive messages leading to DDoS.
Improper Restriction of Excessive Authentication Attempts	307	The backend does not implement a mechanism by which it can block repeated connections to it allowing the attacker to easily brute force the correct EVCS IDs or even launch a DoS attack on the authentication endpoint.
Session Fixation	384	The backend maintains all previous sessions open with all EVCSs having the same ID but only communicates to the last EVCS with that given ID
Reliance on Single Factor in a Security Decision	654	The authentication process of an EVCS relies solely on the EVCS ID that can easily be obtained by the attacker

guessing attacks.

- **Session Fixation:** after connecting the phantom EVCS the OCPP backend establishes a new client session without invalidating the legitimate one. The legitimate EVCS now is unavailable and does not receive any information from the backend although, a session exists and is maintained by both entities.
- **Reliance on Single Factor in a Security Decision:** the tested OCPP backends rely on only the EVCS ID for authentication, which is considered as part of the root cause problem that has led to the discovery of these vulnerabilities.

The different vulnerabilities discovered show a very concerning trend in the development and deployment of the OCPP backends. The different vulnerabilities are chained together to provide an attack vector that exploits authentication vulnerabilities in the different attack workflows described below. Moreover, we validated our discovery on 16 real-life OCPP backends that are distributed worldwide. We discovered a set of vulnerabilities that can be used by the adversary to infiltrate networks of EVCSs without being detected by the

operator's CMS. It is crucial to highlight that such attacks not only impact the EV charging stations and their users but also can have a significant impact on the operations of the interconnected power grid. Thus, introducing a new attack vector against the EV ecosystems and the connected critical infrastructure. Additionally, we would like to highlight that these vulnerabilities cannot be mitigated by simply adopting the newer version of OCPP that includes a TLS encryption mechanism. While TLS provides a mechanism for both entities (OCPP Client and OCPP Backend) to communicate securely and encrypt the data, preventing unauthorized parties from eavesdropping or tampering with the information, it does not provide an authentication mechanism and a patch for the presented vulnerabilities. To examine this issue further, in our testbed, we utilize a secure WebSocket configuration. Yet, the attacks remained successful since its success is independent of any encryption. Additionally, we tested our attack on the testbed after deploying OCPP 2.0.1 including its TLS mechanism, yet the attack success was not impacted. In all of these tests, the vulnerabilities still existed and the attacks remained successful. Especially, the discovered vulnerabilities are due to backend implementation and its security controls. Additionally, such vulnerabilities are not currently impacting the performance of live systems because they are not actively exploited in the wild. In what follows, we provide further information on the proposed attack workflows along with a discussion of possible attack scenarios.

6.2.2 Attack Workflow

In this section, we present details of two attack workflows and validate them on our created testbed [Anonymized]. We use the testbed to avoid performing any intrusive action on the live systems. The testbed provides a real-life replica of a production-grade EV charging ecosystem that has been validated by our industrial partner, Hydro-Quebec, which is one of the biggest utilities in North America. Thus, providing a realistic security assessment.

Testbed Setup. We leverage our created testbed to simulate legitimate EVCS and register

its device ID by establishing a `WebSocket` connection and following the commissioning steps, as described in Section 6.1.1. After that, we follow similar steps to add our phantom EVCS using the registered ID that belongs to the legitimate EVCS. We notice that the OCPP backend accepts the connection from the phantom EVCS while replying with the `GetConfiguration` request, which confirms the device registration. Furthermore, we initiate a `RemoteStartTransaction` from the backend and note that the phantom EVCS will receive it exclusively. This indicates that the OCPP backend will always maintain its communication with the most recently registered client only. On the other hand, we verify that both legitimate and phantom EVCSs will perceive active connections to the OCPP backend by continuously sending ping-pong messages to the backend.

In addition to the testbed analysis, we leverage a list of created PoC exploits to test the vulnerabilities on real-life OCPP backend deployments in each of the 16 operators/vendors. We discuss our analysis results by presenting two attack workflows:

Attack Workflow 1 - Remote EVCS Substitution. As illustrated in Figure 6.8, we showcase the overall procedure that could be followed by an attacker to remotely replace legitimate victim EVCS(s) on the operator's network. Note that the legitimate EVCS (victim) in this case is assumed to be temporarily offline. This could happen due to any reason with the power or Internet connection, or due to software-related procedures (e.g., rebooting). Given this offline window, the adversary will leverage a phantom EVCS to send a `WebSocket Connection` using the ID of the offline victim EVCS. The OCPP backend confirms that the ID is found in the database of registered EVCSs and then allows the consequent communication with the phantom EVCS, thus establishing an OCPP connection. At this stage, the cloud will show that the EVCS is online whereas, in fact, the legitimate EVCS is still offline.

To verify our findings, we analyzed examples of publicly accessible OCPP backend deployments, which reveal the EVCS IDs (as described in Section 6.1.1). Specifically, we

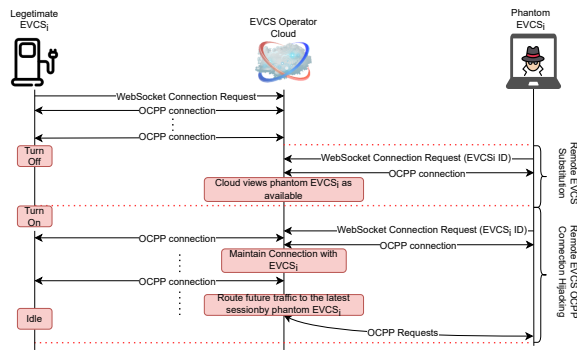


Figure 6.8: EVCS substitution and OCPP connection hijacking attack PoC workflows.

look up registered EVCSs that have an “offline” status. We leverage our list of PoC exploits (GitHub repository [Anonymized]), which are used to infer the vulnerabilities without causing harm to the systems, to test the vulnerabilities on the selected 16 OCPP backend deployments in the wild. Interestingly, we were successful in replacing all the victim EVCSs with our phantom devices while tricking the CMS to update the status of the EVCS to “online”. Finally, we verified the status update in relation to our phantom EVCS deployment by disconnecting the device and observing the status of the EVCS, which turned to “offline”, respectively.

Attack Workflow 2 - Remote EVCS OCPP Connection Hijacking. In this attack workflow, we study the effects of leveraging our phantom EVCS to impersonate an online EVCS, which is actively communicating with the OCPP backend. The legitimate EVCS maintains an ongoing OCPP connection over a `WebSocket` by sending ping-pong messages to maintain an open connection. In parallel, we run our phantom EVCSs with the IDs of the legitimate EVCSs as they continue to operate normally. Surprisingly, our analysis reveals that the phantom EVCS receives the `GetConfiguration` request from the OCPP backend, which verifies the accepted connection to the backend. Thus, the backend will be fooled to communicate with the phantom EVCS for future requested operations.

To this end, we leverage the created PoC for this attack workflow (GitHub repository [Anonymized]) to test the vulnerabilities on selected examples from the 16 deployed

OCPP backends. Our analysis reveals that regardless of the number of newly established connections, the OCPP backend will always accept them as long as use a registered EVCS ID. Indeed, the CMS will only check for a valid/registered EVCS ID without further validation/verification of the received messages or the total number of existing connections. This will cause the CMS to connect and interact with any phantom EVCS, which is abusing a registered EVCS's ID. Additionally, the connection between the legitimate EVCS and the OCPP backend was still maintained, as the victim EVCS will continue to send the ping-pong messages to maintain its `WebSocket` connection. This clearly demonstrates a session fixation security issue. Ideally, the OCPP backend should maintain one connection at a time with respect to a given EVCS ID. Finally, this attack could be extended to multiple EVCSs due to the lack of rate limiting on the accepted request by the analyzed real-life OCPP backend deployments from 16 operators. This could open the door for brute forcing attacks by adversaries who will attempt to guess the EVCS IDs.

6.2.3 Attack Scenarios

In this section, we discuss possible Attack Scenarios (AS) by following the described attack flows and leveraging the identified vulnerabilities in the OCPP backend:

AS1 - OCPP Backend Denial of Service. We leverage attack workflows 1 & 2 to launch 10 phantom EVCSs that have the same valid (registered) device ID. We note that the OCPP backend accepts all registration requests while maintaining a consequent connection with each one of the devices. Following the same approach, the adversary can consume the resources of the backend by launching many phantom EVCSs while causing a large-scale DDoS attack, which is leveraging the lack of rate limiting and improper access control mechanisms. The adversary can also use automated scripts to generate and coordinate a significant number of requests (e.g., `Authorize`, `BootNotification`, and `DataTransfer`) from the existing phantom EVCS towards the OCPP backend within

a short period. These messages are usually initiated by the EVCS, and as described in [89], they can be utilized by adversaries to impact the availability of the backend and cause possible DoS.

AS2 - EVCS Denial of Service. As described in attack workflow 2, while the communication link with the OCPP backend was hijacked by the impersonating EVCS, the legitimate EVCS will continue its normal operation and remain unaware of the situation. Consequently, all the remote functionalities provided by the OCPP protocol will no longer be received by the legitimate EVCS and it becomes virtually out of service. Note that during normal operations, EVCS users will launch the corresponding mobile applications to request charging sessions for their EVs. Ideally, the CMS forwards these requests to the real EVCS. During this attack, all charging requests will be forwarded to the phantom EVCS, and deprive the legitimate EVCS from receiving any request (i.e., DoS).

AS3 - Data Collection and Poisoning. Data mining is crucial for modern companies that tend to perform extensive data logging to get a deeper understanding of their business. Therefore, EVCS operators will leverage the information collected about EVCS utilization to plan future expansions, study user behavior [145], predict attacks on their network, and monitor fleets, to name a few. However, an adversary can leverage attack workflows 1 and 2 to interfere with the communication by substituting the real EVCS in the charging infrastructure. This results in two possible attack scenarios:

AS3.1 - Passive Attack. Attackers can eavesdrop on the communication being sent from the CMS to the phantom EVCSs to gather information regarding the utilization of legitimate EVCSs. Such information can be used later by adversaries to plan and initiate coordinated attacks on the power grid [35].

AS3.2 - Active Attack. The adversary can utilize phantom EVCSs to actively inject fake/-tailored information about the utilization of the existing EVCSs (e.g., by changing their status or charging rate). This is done to skew the operator's data mining process while

gradually poisoning the data collection to degrade the downstream behaviors of any learning models that might be implemented by the operator [146]. Thus, data poisoning will detrimentally impact the infrastructure especially when the operators rely on this data to detect future attacks, as proposed in [147].

AS4 - Firmware Theft. The EVCS vendors usually employ proprietary firmware as a part of their CMS, which could be installed on their charging stations. To keep these firmware up-to-date, the EVCS operators will leverage the OCPP protocol to initiate `UpdateFirmware` requests from the OCPP backend, which in return will automatically fetch the new firmware from the remote host. Therefore, an adversary can utilize phantom EVCSs in listening mode to monitor the OCPP communications for the `UpdateFirmware` request to steal any new software/firmware updates. As described in previous work [19, 35], an adversary can dissect the collected firmware image to perform in-depth security analysis by dumping/mounting the embedded filesystem to explore the various directories and files and discover entry points for remote attacks.

AS5 - Persistent Covert Attacks. Covert attacks aim at compromising the EV charging ecosystem while covering the adversary's actions from being detected. The objective of the adversary is to keep the infrastructure available while remaining persistent in the system (e.g., as a man-in-the-middle) to perform subsequent actions/attacks. As illustrated in Figure 6.9, this is a two-stage attack, which combines OCPP backend and EVCS firmware vulnerabilities.

At the first stage, the adversary will leverage EVCS firmware vulnerabilities [17, 19, 35] to exploit its management system and access its configuration page. The attacker will then update the configured OCPP backend URL of the real operator on the EVCS to redirect the communication to the phantom CMS using the new OCPP backend link. At this stage, the legitimate EVCS will be disconnected from the operator. This impacts the availability of EVCS, which could be detected by the operator (e.g., reported by consumers). Second,

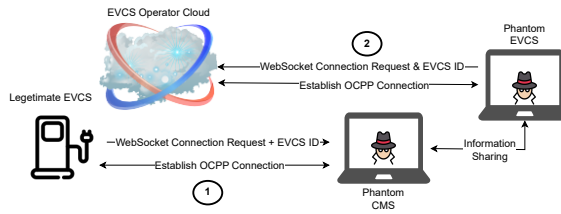


Figure 6.9: An overview of the two-stage persistent covert attacks (AS5).

to achieve a covert/stealthy attack, the adversary will leverage attack workflow 2 to hijack the OCPP backend and register a set of phantom EVCSs that will replace the legitimate EVCS (i.e., using the same device IDs). The operator’s CMS will perceive the legitimate EVCSs to be connected/functional since it cannot differentiate between the phantom and legitimate EVCSs. Finally, to maintain a stealthy/covert attack, the adversary will leverage the connections between the phantom CMS and EVCSs to relay all the exchanged messages between the legitimate operators’ CMS and the real EVCSs. As a result, the adversary can propagate the attack to impact the integrity and the confidentiality of the ecosystem and even extend it to impact the availability while leaving the operator completely oblivious. To this end, we implement the covert attack scenario on our testbed. Specifically, we implemented the OCPP backend using the documentation provided in [77] to develop the introduced phantom CMS. In what follows, we extend AS5 to devise further covert attack scenarios.

AS5.1 - Stealthy Information Theft. This attack is a stealthy extension of AS3.1 where the adversary can collect all the information sent by the legitimate CMS and EVCS to build a full database of the actual operation of the charging network of a specific operator.

AS5.2 - Stealthy Firmware Theft. This attack represents a stealthy extension of AS4 where the adversary can now remain connected to the CMS without impacting the availability of the legitimate EVCSs. By doing so, the attacker can keep collecting the new firmware releases without raising any alarms with the operator. Collecting these updates allows the attacker to dissect this firmware and perform reverse engineering to find further

EVCS vulnerabilities.

AS5.3 - Stealthy Data Poisoning. This attack represents a stealthy extension of AS3.2. Due to its stealth, this attack can be scaled to much larger extents than AS3.2 and perform long-term data poisoning campaigns to skew the operator's datasets and machine learning models away from the real behavior of the ecosystem.

AS5.4 - Stealthy Ransom Attack. An adversary can leverage the covert attack setup to demand ransom from the operators. This can be done by increasing the proportion of the controlled EVCS within the network to maximize the impact of any consequent DoS attacks on the overall operations. The attacker will then demand a ransom from the operator to release control of their infrastructure. Note that the operator CMS will perceive a normal connection to all its EVCSs and therefore, making it extremely difficult to locate the compromised EVCSs within its network.

AS5.5 - Stealthy Power Grid Attack. The adversary can launch attacks on the power grid without the operator having any visibility over the actual behavior of the EV charging infrastructure. Note that the power grid is required to maintain a constant frequency of 50/60 Hz based on the specific standards in each geographic region. However, following our proposed covert attacks, an adversary can synchronize the operations of the controlled EVCSs to force the grid to deviate from the normal operation point and induce forced oscillations that can damage this grid. This is done by forcing synchronized switching behaviors (e.g., charging ON and OFF) to cause the frequency oscillations on the power grid [17, 19, 50].

6.3 Discussion

In this section, we use simulation analysis to discuss a stealthy power grid attack example (AS5.5). Additionally, we recommend generic countermeasures to remediate the targeted systems and mitigate future attacks.

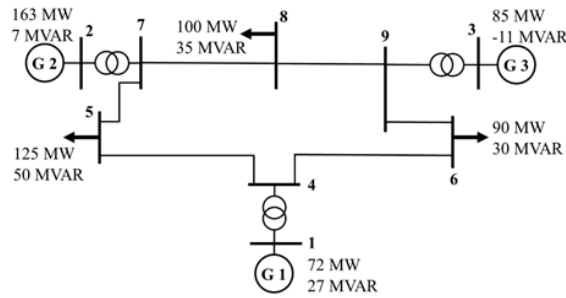


Figure 6.10: WSCC 9-bus grid.

Demonstration of AS5.5. To this end, we use simulation analysis to demonstrate Attack Scenarios 5.5 (AS5.5) and test it on the Western System Coordinating Council (WSCC) 9-bus grid [148], as shown in Figure 6.10. This setup is an approximation of the WSCC to an equivalent system with nine buses and three generators, which is commonly used for grid stability testing when simulating the Western Interconnect in the U.S. and Canada [149]. The simulation was performed using Hypersim, a real-time power grid simulator developed by OPAL-RT [150]. The simulation time-step is $50\mu\text{s}$.

We use the simulated grid to demonstrate the impact of AS5.5 in the form of a coordinated switching attack. Switching attacks are the manipulation of the power grid’s load following a certain oscillatory trajectory often periodic to induce forced oscillations in the power grid’s frequency causing it to deviate from safe operating points. The attack is initiated by increasing the power demand to cause a frequency drop followed by a sudden decrease in load to force a frequency spike. To simulate this scenario, we initiate the EVCS load oscillations every 5 seconds (at $t=5$) followed by an On/Off pattern to cause frequency fluctuation on the grid.

In this iteration of AS5.5, we consider that the adversaries successfully compromised 3,750 EVCSs to be used to attack the grid. As per the International Energy Agency, the average charging rate of a public EVCS is 24kW [151]. As a result, the attacker in this scenario initiates an EVCS switching attack having a magnitude of 90 MW. The impact of this attack on the power grid’s frequency is demonstrated in Figure 6.11. The impact of this

attack causes the frequency to oscillate continuously and to deviate beyond the safe limit of 61.5 Hz (2.5% deviation). Beyond 61.5 Hz at $t=9.8$ seconds (4.8 seconds after attack start), the generator protection relays will trip to protect them from further damage [152]. Thus, this attack will result in a blackout on the grid leaving the consumers without electricity. This simulation demonstrates the feasibility and severity of such attacks, which impact all consumers connected to the power grid while causing huge financial losses [153].

Mitigation and Prevention. Mitigating the discussed vulnerabilities in this work requires the joint effort of the operator and the vendor. The vendor would be required to provide security features that aid the operator in securing the ecosystem. The root issue that enables the discussed attacks is the lack of proper authentication and authorization mechanisms in the OCPP backend. Therefore, maintaining the authenticity of the EVCS and the OCPP backend is crucial for the system and data integrity. In what follows, we provide general recommendations to mitigate future attacks:

Strict Device Authentication

The authentication mechanism is based on a single factor, which is the EVCS ID, which raises a big concern, thus there should be a strict device authentication mechanism. To ensure that we suggest embedding unique client certificates into each EVCS. These certificates can be managed by a distributed storage technology like blockchain to ensure their authenticity. This would require the vendor to alter their firmware to instill this ability. Each message sent by the client should then be signed by a certificate and upon communication, the cloud should be able to verify the authenticity of the message. Thus, making it hard for the adversary to fool the OCPP backend. Additionally, TLS which is originally used to protect data-in-transit can be extended to use certificates to authenticate the different components. However, three techniques can be implemented.

- Client-side authenticated TLS. This would prevent server spoofing which would

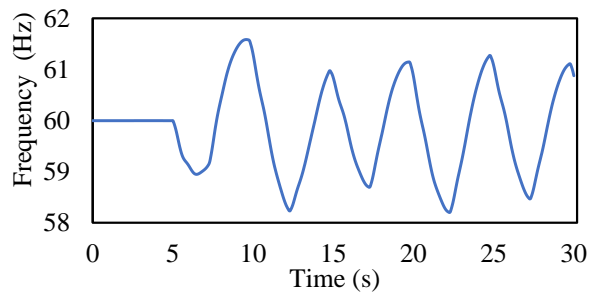


Figure 6.11: Frequency output under AS5.5 with 3,750 compromised EVCSs.

leave the adversary able to spoof the EVCS and create phantom EVCSs

- Server-side authenticated TLS. This would only prevent client spoofing and the creation of phantom EVCSs.
- Mutually authenticated TLS. This type of mitigation would prevent server and EVCS spoofing and the creation of phantom EVCS. However, it does not prevent DDoS attacks.

Mutually authenticated TLS is recommended. This ensures that the parties at each end of the connection are who they claim to be. Both entities will be able to verify that they both have the correct private key. our work shows the (in)security of the OCPP backends and its widespread. While these security best practices are well known, the EVCS ecosystem operators are not meeting the minimum security bar to ensure a reliable and secure system. Such changes require firmware updates and development from the vendor and the operator at the same time to instill security controls. However, the presence of multiple operators and multiple vendors that supply each operator makes it challenging to drive standardization among them. The mutual authentication using TLS certificates, however, does not address the vulnerabilities categorized by CWEs 799, 307, and 384. We address their possible solutions below.

Routine Security Checks

The device ID should be routinely changed for all EVCSs and generated randomly using an algorithm that leverages hard-to-guess information and does not follow a specific pattern. Additionally, we suggest performing authorization checks routinely without trusting devices and following a zero-trust approach. The OCPP backend should maintain strict 1-1 communication with the device without allowing other entities to utilize their communication information. This can be done by maintaining a status table for each online entity and checking against the database continuously and routinely. Moreover, the operator should implement rate limiting on their endpoint preventing excessive authentication and connection requests. We plan in our future work to elaborate further on the mitigation procedure.

Limitations and Future Work. We discuss some limitations of this work and possible ways to address them. While we analyzed 16 widely used OCPP backend implementations to verify the identified vulnerabilities in various vendors/operators, it is worth noting that obtaining information about all available operators and their deployed systems is a challenging task. This is due to the sensitive nature of their operation and the proprietary nature of these systems, which are often closed to analysis, even for research purposes. Additionally, we relied on the authenticity of the identified vulnerabilities reported in recently published work (e.g., Nasr et al. [35]) to discuss the feasibility of exploiting the EVCS firmware as a part of attack scenarios (e.g., covert attacks AS5). However, while some vendors have partially addressed these vulnerabilities, the majority of the studied systems in previous work remain vulnerable and exploitable. In our future work, we will extend the existing testbed to include further components while performing an active analysis/exploitation within the EV charging ecosystem. Finally, we plan to purchase various actual EVCSs to extend our testbed and perform more realistic and vendor-specific analysis to determine the security posture of the OCPP backend implementation.

Chapter 7

Edge-based detection and localization of adversarial oscillatory load attacks orchestrated by compromised EV charging stations

7.1 Threat Model

An adversary that can compromise and control a large number of EVCSs is considered. There are multiple attack vectors that can be used by the adversary to impact the power grid that is taken into consideration in the detection mechanism. Namely, the different attack vectors are described below:

- The internal components of an electric vehicle that have internet connectivity such as the On-Board Diagnostics (OBD) port that can be accessed physically or wirelessly and grant access to the Controller Area Network (CAN) bus, which could be leveraged by the attacker to control the vehicle and its charging [22].

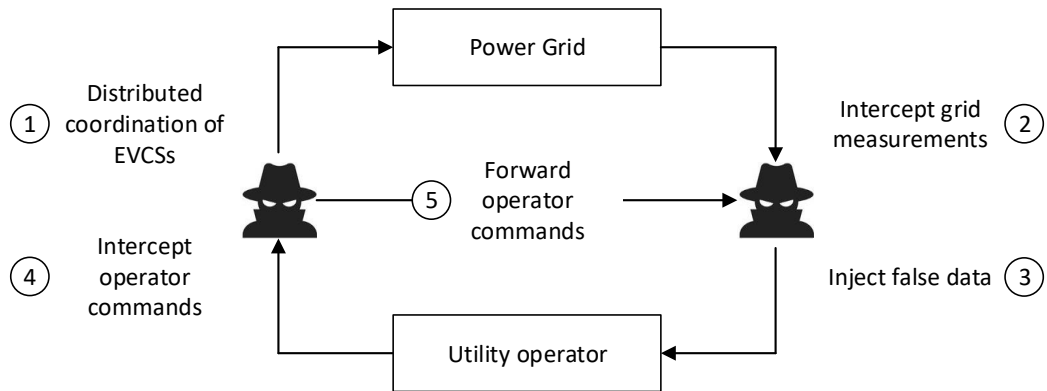


Figure 7.1: Overview of the covert attack.

- The mobile application which is the component responsible and the enabler for the commercialization of the EVCS ecosystem could be used by the adversary by leveraging the lack of end-end authentication between the user and his vehicle [50], allowing the adversary to opportunistically take advantage of connected vehicles to the charging station.
- CMSs are vulnerable to remote attacks. The adversary can exploit one or more operators' management systems (multi-operator) the adversary can perform attacks against the power grid by commanding a large distributed EVCS botnet. The adversary could create different combinations of attacks by leveraging multiple CMSs.
- The OCPP protocol is also taken into consideration which has been found vulnerable to MitM attacks that could be used to initiate and bypass any protection mechanism deployed on the cloud [20].

Consequently, unlike in [3], the attacker, after gaining control of the EVCSs, can launch various types of oscillatory load attacks not limited to inter-area oscillation oscillatory load attacks. Moreover, the adversary does not necessarily take advantage only of public charging stations but could also leverage privately owned charging stations.

The attacker is assumed to have the capability to launch covert attacks [154] as illustrated in Figure 7.1. The adversary, as shown in Step 2, controls a considerable number of

EVCSs and can command a coordinated oscillatory load attack against the grid. However, to thwart the utility operator's detection mechanisms, the adversary intercepts (Step 2) measurements and readings that the operator collects to monitor and estimate the state of the grid and injects false data (Step 3) which deceives the physical-layer detection mechanism hosted by the utility operator, and thus renders the grid oblivious of the grids' actual state. It is worth mentioning that the adversary injects data that resembles the normal behavior of the grid. Consequently, the utility operator sends commands to the power grid components (e.g., generators) to perform some actions to stabilize the grid based on historical data (e.g., load demand trends), thus the adversary intercepts these commands (Step 4) and forwards them to the false data injector so that the operator can see expected data trends and would not trigger an alarm at the physical layer (Step 5). The attacker can establish covert channels by injecting malware/ransomware [75] (e.g., BlackEnergy malware injected into Ukraine's power grid [155], Stuxnet Malware infected Iran power grid [156]) into the networked controller and arbitrarily alter the control logic. In this work, these threat vectors are addressed using these detection and mitigation mechanisms. Where the attacker's main goal is to induce forced oscillations that would impact the frequency of the grid.

Now, oscillatory load attacks require the coordination of numerous charging stations simultaneously, and it is acknowledged that the current number of EVCSs is not enough to launch the proposed attacks. However, with the current exponential increase in the adoption of electric vehicles and the rapid deployment of EVCSs to match the adoption rate, such attacks pose a great threat to power grid stability. To demonstrate the feasibility of such attacks the New South Wales (NSW) grid is chosen, whose size is similar to the NE-39 bus grid used in the dataset collection. The NSW grid has an average load of 6989MW [157] and a total number of registered vehicles of 5,892,206 [158]. Scaled to fit the 6097MW 39-bus grid, the total number of vehicles in the grid would be 5,155,681. Assuming a future projection of 50% EV penetration, the grid will contain over 2.5 million EVs. As per the

International Energy Agency (IEA) [80], based on the mixture of available EVCSs, the average charging rate per EVCS is 24kW. Based on these statistics, these attacks only require a small portion of the available EVs to be successful. The largest attack magnitude, for instance, constitutes 30% of the grid load. This translates to only 3% of the available EVs. By comparison, the smallest attack magnitude only requires 1% of the available EVs. When this analysis is performed for the 9-bus system, used in the distributed mitigation section, it is noticed that it only requires 2.6% of the available EVs when 50% penetration level is assumed.

7.2 Methodology and System Model

In this section, the detection and mitigation methodology and conceptual model are discussed. A discussion of the system model is provided, followed by a detailed discussion of the distributed detection methodology. The data-set curation and collection are also discussed. Finally, the distributed mitigation methodology is discussed, and an overview of the real-time co-simulation testbed is provided on which the mitigation mechanism [2] is demonstrated.

7.2.1 System Model

This approach attempts to ensure fault tolerance in the deployment of the detection mechanism while handling oscillatory and adversarial oscillatory attacks. To mitigate covert sophisticated attacks that allow adversaries to deceive traditional physical-layer detection mechanisms, detection should occur at different levels of the interconnected system to build resiliency into it. To address the limitation of previous work, e.g., [3] and other centralized detection mechanisms, a deep learning model deployed on the EVCS is proposed since, the EVCS possesses the ability to collect information about the true operations of the

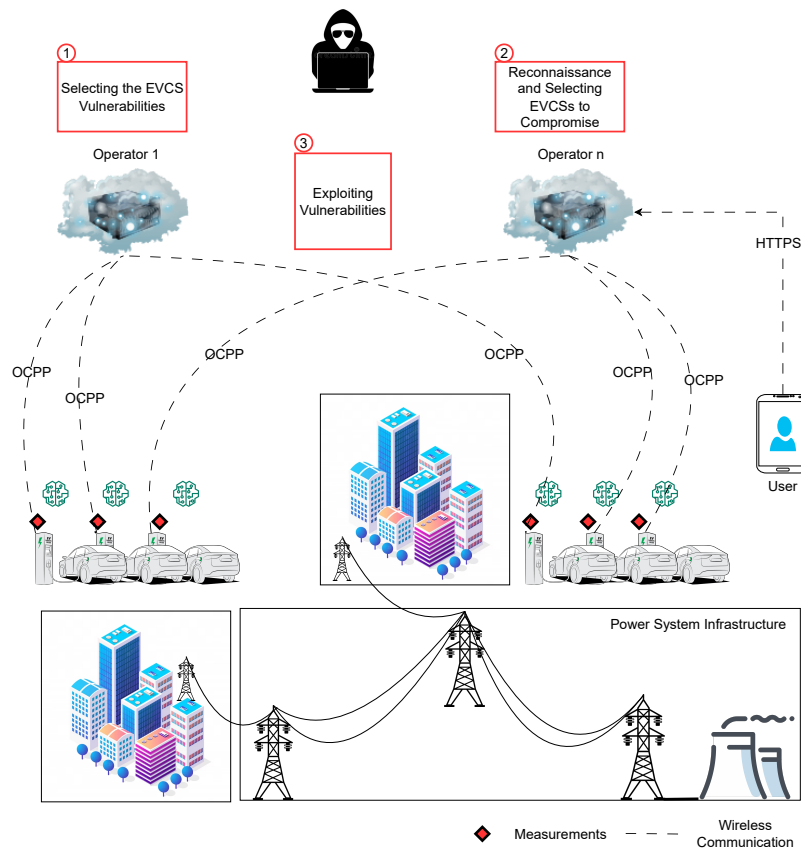


Figure 7.2: Illustrate the different blocks used to simulate the cyber-attack scenarios.

charging stations and power characteristics (e.g., frequency). This presents an advantage over CMS-based detectors where a compromised OCPP connection allows the adversary to inject bi-directional false data that would affect the detection mechanism deployed there. Moreover, the EVCS is the component that is utilized by adversaries to perform physical attacks by compromising other components (e.g., mobile application, CMS, or OCPP). Thus, securing the EVCSs would prevent attacks initiated from any vulnerable point in the EVCS ecosystem. Finally, centralizing the detection mechanisms creates a single point of failure, and maximizes the risk of exposing the deep learning model, and polluting the data since recent incident reports and studies show the vulnerability of the system at scale. Whereas the deployment of a deep learning model on the EVCS would hinder the ability of the adversary due to the distributed and independent operation of charging stations (increased

resiliency).

Consequently, it is illustrated in Figure 7.2 the different blocks that simulate the cyber-attack scenarios. Namely, an adversary follows three steps to launch attacks against the power grid. From a cyber layer perspective, the adversary aims at changing the behavior of the charging station through different attack vectors that ultimately, achieve similar results. In the first stage of the attack, the adversary should choose the attack vector. Different attack vectors could be exploited that are described in section 7.1. Namely, the adversary could exploit the mobile application and take advantage of idle vehicles connected to the charging stations by leveraging the lack of end-end authentication between the driver and their vehicle [50]. Moreover, the adversary could attack the cloud system of the operators to gain control over the charging stations remotely and change their charging behavior [17]. Additionally, the adversary could intrude on the OCPP connection [20] breaking the integrity of the system and allowing the adversary to act as a MitM. In the second stage of the attack, the adversary will perform reconnaissance. The adversary will monitor the behavior of charging stations through the mobile application to understand the utilization behavior of the EVCS. The adversary then selects charging stations based on the different attack vectors mentioned. Finally, the adversary will be able to synchronize the on/off behavior of the targetted EVCSs to cause forced frequency oscillations on the power grid. However, using the proposed approach a machine-learning model is deployed on each EVCS that utilizes measurements from the EVCS to feed the machine learning models with information. The EVCS collects the frequency and the events are then encoded and fed into the machine learning model that bakes them into the decision-making process. Since the EVCS is the final element in the attack chain, the mentioned attacks cannot surpass it as the case with the centralized approach that can be surpassed using MitM. Using MitM to hijack the OCPP connection would result in the centralized detection mechanism not being able to see any of the requests being sent to the EVCS thus the attacker would avoid the

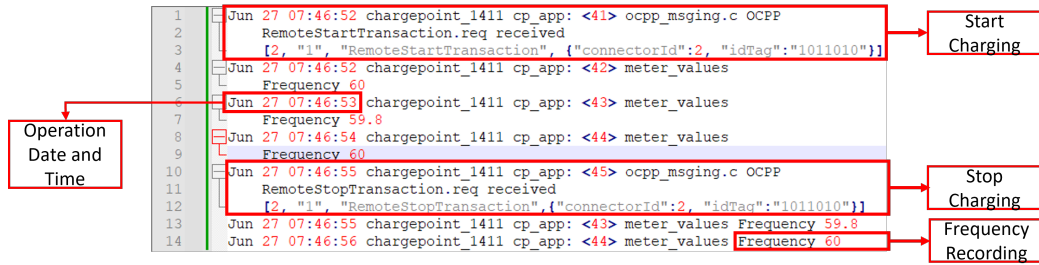


Figure 7.3: EVCS log showing the different features that could be extracted from the charging station logs.

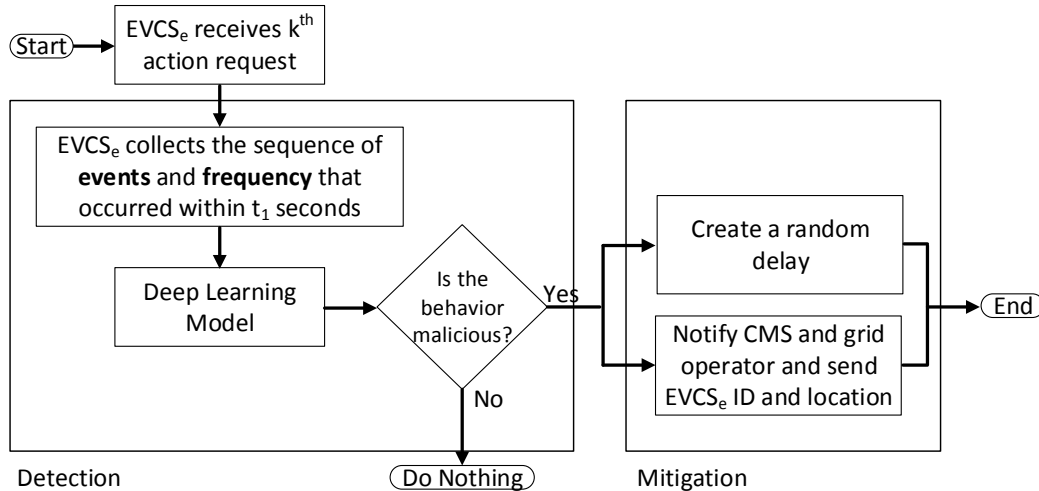


Figure 7.4: Flow chart describing the detection mechanism.

detection mechanism altogether.

Consequently, to deploy a deep learning model on the charging station, new features should be derived compared to the work of [3], which utilizes information that only the CMS has access to (e.g., change in a load of vehicles during a certain δ time). Thus, the usage of various deep learning techniques is investigated to detect attacks against the power grid initiated by the EVCS ecosystem. To the best of our knowledge, this work is among the first to investigate a decentralized cyber-detection mechanism deployed on the EVCS ecosystem to protect the grid from the new vulnerabilities of this cyber-physical system. Residential and public charging stations both have log files to record all the operations/events of this station. In Figure 7.3 a sample EVCS log is shown where each transaction might include the following information: EVCS ID, operation type (e.g., charging,

or stop), operation date, start operation time, stop operation time, charging rate, type of the charger, and the variation of the frequency of the load bus that the EVCS is connected to overtime. It is worth noting that the OCPP protocol provides a functional block that enables charging stations to send periodic meter values (e.g., voltage, reactive power, etc.). Thus, using the telemetry data collected by the charging station, the power grid frequency, which is directly linked to the speed of the generators, can be directly recorded by the EVCS with high granularity by measuring the period of the voltage waveforms that are sampled over time. It is worth highlighting that electric devices (e.g., charging stations) will exhibit the same frequency as the bus they are connected to. Thus, to collect grid frequency measurements the utility monitors and collects measurements from the buses which incidentally have connected EVCSs. The recent industrial technology advances increased the connectivity of cyber-physical systems that are monitored and controlled by Supervisory Control and Data Acquisition (SCADA) systems that use advanced computing, sensors, control systems, and communication networks [159]. SCADA systems allow power grid operators to gather real-time telemetry data about the grid. This information that the grid operator can acquire from the buses can be used for training deep learning models since, when deploying the model each charging station should be able to gather this information by itself.

Accordingly, the charging station can store each operation in its log file and use it to detect anomalies in the usage of the charging station, which indicates that there is a possible attack initiated from the EVCS ecosystem against the grid. This information can be updated in the log file actively. However, since the utility (power grid operator) is the main entity affected by oscillatory attacks, it will take responsibility to gather information from different operators and distribute trained global models to the connected EVCSs. Collaboration with the utility by various EVCS operators is mandatory to allow a collective view of

multi-operator attacks. The utility will use past data to train and deploy deep learning models on the charging stations to alleviate any future privacy concerns the operators might have about sharing their data. This work focuses on public charging stations to create a distributed detection mechanism. Due to the unique features that have been intentionally chosen to address the limitations introduced by a cloud-centric detection mechanism, the detection mechanism is applicable to private charging stations as well. The unique features that were used to ensure the flexibility and the ability of this solution to make stand-alone decisions on any charging station without requiring it to communicate with the management system and share information about the utilization of the charging station. This makes this approach privacy-preserving and suitable for application on private EVCSs in addition to public EVCSs.

DETECTION MECHANISM: Figure 7.4 gives an overview of the proposed detection mechanism to be deployed at the charging station. When an EVCS_e receives a charging request (k^{th} request), the EVCS_e retrieves the events that occurred in the last t_1 seconds from its logs. Similarly, it retrieves the frequency readings that have occurred and are collected within the same period from its logs. This information is fed into a machine/deep learning model to detect maliciousness of the events that occurred within the last t_1 seconds. By leveraging the combination of the cyber data (series of events) and physical data (frequency on the power grid) that are tightly coupled in case of a coordinated oscillatory load attack, a deep learning model is created that will extract the temporal and spatial relationships between the sequence of readings over time. The observed behavior of the charging station and the underlying infrastructure is used to characterize oscillatory load attacks and differentiate them from the normal functioning of a charging station. It is worth highlighting, that the t_1 seconds is a rolling window, and the detection mechanism is real-time. It only requires the t_1 window to detect the attack. This also means that no additional extensive data logs are required to be kept on the EVCS_e since the proposed algorithm will not use

any of the data prior to the t_1 rolling window.

MITIGATION MECHANISM: If the deep learning model labels the sequence of events as malicious, the EVCS_e will create a delay block that will randomly delay request between 0 to 4 seconds to disrupt the synchronization of the oscillatory load attack and notifies the CMS and grid operator by sending the EVCS ID and location. The mitigation mechanism allows distributed and independent decision-making for each charging station, thus ensuring fault tolerance in the proposed mitigation mechanism. Consequently, to test this mitigation mechanism on the proposed testbed which is used to study the impact of the EV ecosystem on the power grid. The results (discussed later) show the effectiveness of neutralizing the impact of an oscillatory load attack on the generator's speed and minimizing the risk and the costs incurred by a successful attack. It is worth highlighting that the independence of the proposed techniques from the features or artifacts that need a global knowledge of the ecosystem and grid provides the flexibility needed to deploy the proposed detection-mitigation mechanism on any EVCS.

7.2.2 Distributed Detection Mechanism Methodology

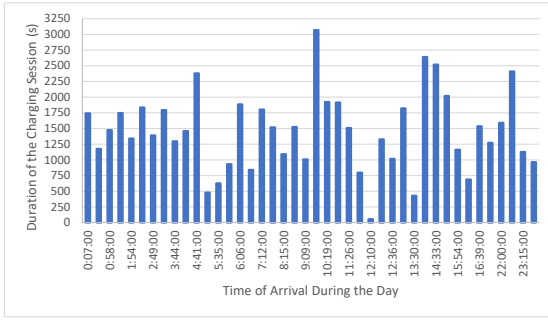
Given the limited number of previous works, which discuss the detection and localization of oscillatory load attacks, along with the limitation of previous detection approaches, the aim is to deploy an edge-based AI-enabled detection mechanism on the charging station itself by leveraging cyber and physical characteristics (e.g., charging events and power grid frequency variation) to identify and characterize malicious and benign behaviors. More specifically, the devised methodology attempts to leverage the behavioral characteristics of an oscillatory load attack to propose an effective edge-based, decentralized oscillatory load attack detection.

To achieve the objectives, understanding the normal behavior of charging stations by

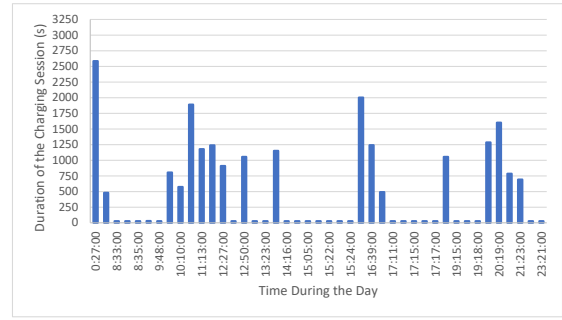
examining a real-life EVCS dataset is crucial. This dataset was obtained from Hydro-Quebec as part of a legal agreement and research collaboration. Hydro-Quebec owns and operates, through a subsidiary, the public EVCSs in Quebec. This data is used to understand the behavior of the public EVCSs ensuring normal behavior and extracting certain features that allow the utility to build its own realistic data-driven normal EVCS behavior. First, identify the state changes of a charging station. The charging station alternates between three states: idle, charging, and discharging. The discharging state is when the vehicle is used to inject power into the power grid using the V2G technology that is available in some EVCSs. A few pilot projects have been launched worldwide (e.g., Pacific Gas and Electric Company (PG&E) in California) and they use the EV load to support the power grid during peak hours. Whenever a charging station receives a charging request it transitions from idle to charging, and when it receives a stop charge request the charging station goes back to the idle state. However, the duration that the charging station spends in any of the states needs to be understood since the oscillatory load attack is tightly coupled with the total attack load and the time spent in each state. The dataset is acquired from 7,000 EVCSs located in different geographical locations of Quebec from 2018 to early 2022 to cover all four seasons of Canada and their corresponding influence on charging behavior. The data contains multiple principal metrics about charging sessions (e.g., start time, end time, and duration of charging). The normal behavior of the charging stations falls under two general observations 1) normal behavior of a charging station with charging > 5 minutes; 2) switching behavior of an individual charging station that does not impact the grid. Consequently, the charging behavior of one heavily utilized and one lightly utilized EVCS located around the downtown area in Montreal is analyzed. The average duration of EVCS 1 (Figure 7.5a) is 24 minutes with a minimum of 55 seconds. Whereas EVCS 2 recorded an average duration of 8 minutes with a minimum of 26 seconds. After further analysis of EVCS 2 (Figure 7.5b), a switching behavior is observed at 8:33 A.M which was followed by two

other switches at 8:34 and 8:35. Similar behavior was repeated at 9:47, 15:21, 17:11, and 19:15. The two charging station behavior patterns are identified based on their utilization where their hardware specifications are the same (providing an 11 kW charging rate). It is worth mentioning that, a switching behavior occurring simultaneously on numerous charging stations would be considered a coordinated oscillatory load attack. This observation shows that the behavior of a charging station by itself is not enough to detect oscillatory load attacks because it might cause numerous false positives and false negatives due to the presence of a switching behavior during the normal operation of a charging station. As the number of charging stations increases, this phenomenon is expected to increase among charging stations. Moreover, since detection occurs on the charging station that does not have any information about other charging stations, by coupling the events happening on the charging station with the frequency readings over the studied t_1 time. The frequency is a global variable shared between all charging stations that are connected to the same bus, which allows the charging station to gain global knowledge of the EVCSs connected to the same bus while keeping the detection local to itself. During a synchronized oscillatory load attack, events that occur on the EVCSs are tightly coupled with grid behavior. Therefore, the events that occur on the charging station (e.g., start time, end time, and duration) and the grid behavior (e.g., frequency) over time are coupled. Hence, using the mentioned features, this approach aims to detect and localize a synchronized oscillatory load attack with the fine granularity of identifying the charging stations that were compromised to perform such attacks.

DATA SYNTHESIS AND COLLECTION: A crucial part of the proposed detection mechanism is creating a comprehensive and realistic dataset that resembles both normal and malicious behaviors. Since the existence of such attack data is scarce in real life and due to the unique features chosen, a realistic data-driven EV load profile is created. To achieve this, a Poisson arrival process of EVs to each EVCS is simulated independently. The charging



(a) Charging Station 1



(b) Charging Station 2

Figure 7.5: Normal charging behavior of two different charging stations.

time of these EVs is then simulated as a truncated Gaussian distribution. The parameters of the arrival and charging time models are specified for different periods during the day and for different seasons. These parameters are tuned based on the Hydro-Quebec EVCS dataset. Finally, the impact of normal charging on the power grid is simulated along with the behavior of the power grid as a result of the different oscillatory load attacks launched. The simulated dataset will be used to train the proposed detection model due to the lack of real data with the required granularity (0.5 seconds) published online. This work focuses on anomaly detection for the detection of synchronized oscillatory load attacks. To this end, the behavior of the EVCSs and grid under normal and attack conditions are coupled. As mentioned above, this method offers the coupling of the cyber events occurring on the charging station to the physical data, i.e., power grid frequency behavior.

The normal arrival of new charging requests at a charging station is coupled with the normal frequency behavior of the bus to which the charging station is connected. The arrival of charging requests during an oscillatory load attack is coupled with the abnormal frequency behavior of the bus to which the charging station is connected. To this end, the IEEE New England 39-bus system [160] was built in MATLAB Simulink to gather the required power grid data. MATLAB-Simulink 2020a Specialized Power Systems Toolbox which is widely used for system stability studies [161] is used. The Simulink Power System

Toolbox models all the different components of the power system (i.e., loads, lines, transformers, generators, and generator control systems). Given the dynamic behavior of the power system, it is mostly governed by the control system of the generators and the models commonly adopted by stability studies are used i.e., round rotor type synchronous machine block of Simulink, generator exciter model IEEE T1, turbine speed governor IEEE G2, and a power system stabilizer based on IEEE Std 421.5. The simulations were performed with a simulation step size of 1ns.

The implemented model would allow the study of the transient and steady-state behavior of the system. To simulate the normal frequency fluctuations of a power grid, random load blocks to all load buses were added. IEEE grid models have constant loads which usually represent the average load of the bus. However, real consumer behavior is random during a short span of a few minutes such that its average is what is reported and planned by utilities. This gives rise to the need to simulate small random perturbations in loads of the power system which would lead to normal frequency variations. The random load blocks added to all load buses are constituted of a random number generator and a dynamic load block which are provided in Simulink. The magnitude of the random number generator is scaled by the nominal load of the bus it is connected to and a multiplication block with a percentage cap that is changed in every simulation run. This setup is used to control the real and reactive power of the dynamic load block. The power factor of the random load block is maintained at 0.8 lagging to simulate benign consumer load variation. In half the simulations, the random source was set to follow a Gaussian distribution and in the other half, it followed a Uniform distribution to increase the randomness in the data and simulate close to real-life load perturbation. To avoid the pattern effect of pseudo-random number generators and to ensure true randomness, the Mersenne Twister algorithm with a period length of $2^{19937} - 1$ is used and the shuffle command before every simulation is run to randomly select new seeds for the random number generator and guarantee further

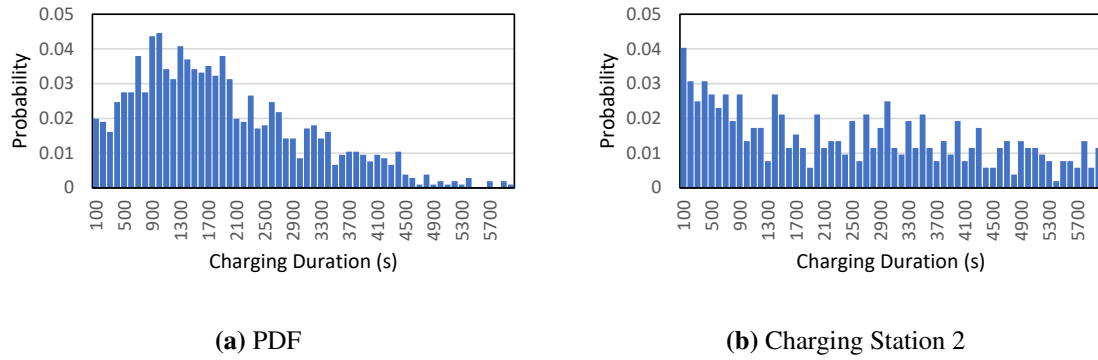


Figure 7.6: Normal charging behavior of two different charging stations.

randomness.

The constructed system is used to create a dataset of 5,000 normal (no attack) scenarios and 5,000 oscillatory load attack scenarios by collecting cyber layer measurements, EVCS events, and physical layer measurements, grid frequency, from the simulated system. The normal dataset constitutes 80% of a behavior similar to EVCS 1 in (Figure 7.5a), whereas the other 20% follow the behavior similar to EVCS 2 in (Figure 7.5b). note that the behavior depicted in Figure 7.5 demonstrates the normal behavior of two charging stations that are found in the dataset acquired from the industrial partner Hydro-Quebec. Figure 7.6 represents the probability of the charging session duration of the 2 aforementioned EVCSs. These 2 histograms were constructed based on the entire records extending from 2018 to 2022. These 2 histograms demonstrate how both EVCSs have charging sessions whose duration is less than 100s. Figure 7.6b also demonstrates how EVCS 2 has double the probability of experiencing charging sessions with a duration of less than 100s as compared to EVCS 2.

Due to the similarity between charging station 2 and an EVCS under attack, frequency recordings are included as part of the decision-making process. Moreover, the following 4 scenarios are identified and classified as normal behavior. Denoted by λ is the arrival rate used for the Poisson process that was used to simulate the arrival of vehicles to an EVCS following the work in [3]. The data was obtained from an industrial partner to determine

the different average arrival rates of EVs to the EVCSs across the province of Quebec in Canada:

- Very slow charging station switching (normal charging request start and stop) and normal bus frequency behavior. Charging events with a very low arrival rate (e.g., $\lambda < 6$ event per 60 minutes), while the grid shows a normal frequency fluctuation.
- Very slow charging station switching and abnormal bus frequency behavior. Charging events with a low arrival rate (e.g., $\lambda < 6$ event per 60 minutes), while the grid shows abnormal fluctuation in the frequency. This abnormal grid behavior can either result from some sudden benign disturbance on the grid or from an attack that does not involve the charging station in question.
- Slow charging station switching and normal bus frequency behavior. Charging events with a high arrival rate ($\lambda > 6$ events per 60 minutes), with a normal frequency fluctuation as a result of normal consumer behavior.
- Fast charging station switching and normal bus frequency behavior. Charging events with a very high arrival rate ($\lambda > 6$ per 60 seconds), coupled with normal frequency fluctuation. This case represents a few actual cases that were monitored where the EV owner connects and disconnects a few times a minute at the time of arrival. Furthermore, the absence of any abnormal frequency behavior means the absence of any abnormal behavior on the power grid and thus the absence of a synchronized attack.

Two cases of attack behavior can be identified including fast charging station switching with abnormal bus frequency where the adversary performs periodic attacks and records the events and the frequency fluctuation. The attack frequency was as fast as 1Hz. The second case is slow charging station switching with abnormal bus frequency where an adversarial attacker with more resources can compromise more charging stations than required for the

attack and distribute the switching behavior among them to remain stealthier. If the attack for example requires n charging stations switching at a frequency of f Hz, the attacker can compromise $m \times n$ charging stations and switch them at a frequency of $\frac{f}{m}$ to obtain an identical aggregate effect but with a fraction of the switches on every charging station and remain stealthier.

The switching of the charging stations is crafted in a way such that the aggregate load on the buses adheres to the following behavior. For the slow switching attack scenario, the load is distributed such that the aggregate switching load has a duty cycle (the proportion of time during which an electrical device is operated) of 35%, 50%, and 60% each constituting a third of all cases. Moreover, all three oscillatory load attack variations are crafted with an aggregate period between 1s and 2 s (0.5 -1 Hz) and an aggregate attack magnitude between 10% and 30% of the bus load. This provides a comprehensive dataset that simulates the different types of oscillatory load attacks (switching and dynamic) that are discussed in Chapter 2. The different types of attacks differ in their magnitude, periodicity, time of the attack, and stealthiness; however, their collective impact on the power grid is comparable in terms of forcing abnormal frequency oscillations. Consequently, this work aims at detecting oscillatory load attacks to mitigate their impact on the power grid.

Using the normal and attack data discussed above, the detector was trained on a wide array of data spanning a wide range of attack durations, duty cycles, and instances of switching which resulted in a wide range of frequency disturbances. This is performed to train the detector to identify general attack behavior rather than a single specific type of attack (duration, frequency, instance of switching). This allows the Deep Learning model to correctly classify attacks that it has never been trained simply by extracting features that resemble the wide variety of attacks it had been trained on.

Given the obtained dataset of cyber events on the charging station and with frequency

data from the grid, a time-series selection/representation approach is adopted. After coupling the two features that vary with time, where one of these features (events on charging station) affects the other feature (frequency) it transforms the problem from a time series classification to a multivariate time-series classification with two axes of difficulty. Temporal and spatial relationships are learned and mined using deep learning techniques to extract the variation of the features concerning time and how features vary from each other.

Each charging station monitors the previous 120 seconds in rolling windows whenever it receives a request and fetches readings from its log file. The events and frequency are recorded every 0.5 seconds, which results in 240 readings for each feature per instance. The rolling window size and the number of readings are fixed, due to simulation environment limitations of 120, and 240 respectively. In this study, deep learning models are devised to detect attacks and optimize them to detect attacks after 5 seconds and 10 seconds of attack start, which are called detector-5 and detector-10, respectively. This means that the last 5 seconds or 10 seconds of the 120-second window will have attack features whereas the rest would be normal behavior. We use these two attack windows to identify attacks early as compared to previous work that depended on monitoring 20 seconds of the attack to make a detection [3]. Through this approach, the proposed detection mechanism can successfully detect almost all attacks as early as 5 seconds by only viewing the beginning of the attack. Since the proposed methodology is based on a rolling window, this allows the algorithm to identify attacks if any false negatives occur in previous windows. Furthermore, since this algorithm utilizes a rolling window, it can start detecting the attacks, with some degree of success, as soon as 1s after the attack starts. Decreasing the size of the attack windows and optimizing for it would decrease the accuracy of the machine/deep learning model as stealthy attack scenarios would be misclassified. In the stealthy version of the attacks, the attacker launches slow oscillatory attacks as well as distributes the switching among multiple charging stations. This means that in short windows of time, the charging station

behavior would look normal since only one event or possibly no events at all occur making their behavior look completely normal. Consequently, 5-second attack windows are chosen to detect a wide variety of attacks without compromising on accuracy. The window rolls in intervals of 1s (split into two 0.5s sub-intervals) which means that the proposed detector can start recognizing the attacks within 1s of their attack start.

FEATURE SELECTION: Given the obtained dataset of events coupled with their power grid frequency readings, two feature selection/representation approaches were adopted. A sequence of observations that are taken sequentially in time, defines the data to be time series. Consequently, to use a set of time series $\mathcal{D} = \mathcal{X}_{i=1}^N$ as input for the deep learning algorithms, and maps each time series \mathcal{X} of set \mathcal{D} into a matrix of \mathcal{N} rows and \mathcal{M} columns by choosing \mathcal{M} data points of the two variables (events and frequency) from each time series \mathcal{X}_i as elements of the feature vector. This allows the deep learning model to take into account temporal and spatial information and find the correlation between the events and the frequency.

CLASSIFICATION MODELS: Given the features selected and the complexity of relating cyber data and physical data to perform the classification, a classification algorithm is needed to handle this data and preserve its properties. To this end, different deep-learning classification models are implemented and evaluated. Specifically, Recurrent Neural Networks (RNNs) are used to capture the order, occurrences, and structure of the events. A special type of RNN is leveraged, namely Long Short-Term Memory (LSTM). LSTMs preserve the errors that will be backpropagated through layers that allow LSTMs to continue learning over many time steps. LSTM is unique in its capability to learn what information to store in long-term memory. LSTM also allows the neural network to identify patterns and sequences in the data by learning temporal relationships between multiple time steps while utilizing memory gates. These features allow LSTM to capture the temporal relations between events in a multivariate time-series data classification problem.

A special type of Convolutional Neural Network is explored, namely a 2D Convolutional LSTM. In the proposed multivariate time series classification, it is important to capture spatial interpretation and relationships. The events that occur on the charging station, in case of synchronized switching attacks, are tightly coupled with the power grid behavior. Capturing spatial information between the cyber layer and physical layer features allows the algorithm to capture the correlation between events on the EVCS and the power grid frequency behavior. Thus, ConvLSTM nodes possess convolutional capabilities to handle spatial information and LSTM capabilities to handle temporal information, solving dual-axis data relationships [162, 163]. ConvLSTMs were used to overcome the major limitation of LSTMs in finding spatial relationships between features over multiple time steps [162]. Unlike LSTM which flattens the data and loses any spatial relationships, ConvLSTM replaces the LSTM gate in each LSTM cell with convolution operation made up of several filters of square matrix kernels. By doing so, ConvLSTM captures underlying spatial features by convolution operations in multiple-dimensional data while preserving the temporal relationship between the data as well. We compare to vanilla LSTM to show how the ConvLSTM can improve the detection by identifying spatio-temporal relationships between the features studied.

MODEL EVALUATION AND COMPARISON: Several standard methods were followed to evaluate the overall effectiveness of the implemented classification models to compare their outcomes. More specifically metrics such as accuracy, recall, precision, and F-measure are used. Moreover, a confusion matrix is also used, which is a useful method for discussing the effectiveness of the implemented deep learning models. The confusion matrix shows the number of data instances that were classified correctly using the model (true positive and true negative) and the number of data instances that were misclassified by the model (false negative, false positive).

Algorithm 1: Algorithm describing the conceptual model of the detection and mitigation mechanisms

```

Inputs :  $CS_{log}$ : Charging station logs; //Events and frequency logs
1  $Conv - LSTM_5$ : the detector-5 deep learning model; //model for detection within
the first 5 seconds
2  $Conv - LSTM_{10}$ : the detector-10 deep learning model ; //model for detection within
the first 10 seconds
Output:  $L_{test}$ : the prediction class for the test sample in  $CS_{log}$ .
 $d_1$ : the delay of the incoming requests.
3 while True do
4   foreach Event  $e_i(t) \in CS_{log}$  do
5     do in parallel
6        $E_{prior}_i \leftarrow$  collect events that happened prior to  $e_i$  in the last 5 seconds
7        $F_{prior}_i \leftarrow$  collect frequency readings that happened prior to  $e_i$  in the last 5 seconds
8        $E_{prior}_i \leftarrow$  encode( $E_{prior}_i$ ); //encode the events to 0s and 1s
9        $F_{prior}_i \leftarrow$  collect frequency readings that happened prior to  $e_i$  in the last 10 seconds
10       $F_{prior}_i \leftarrow$  scale( $F_{prior}_i$ ); //scale the data to enhance the machine
learning prediction and minimize the bias of the machine
learning
11       $x_i \leftarrow$  ( $E_{prior}_i, F_{prior}_i$ ); //events and frequency readings tuple that
is used as an input to the model  $M_1$ 
12       $L_1 \leftarrow$  Conv-LSTM5( $x_i$ ); //predict the class of the behavior
recorded
13      while  $L_1$  or  $L_2$  is Abnormal do
14         $d_1 \leftarrow$  Random Delay0 < d ≤ 4seconds; //continue generating random
delays to the new incoming requests
15        do in parallel
16          wait( $d_1$ )
17          executeEvent( $e_i$ )
18        do in parallel
19          NotifyOperator(); //Report abnormal behavior to the
operator/utility
20      do in parallel
21         $E_{prior}_i \leftarrow$  collect events that happened prior to  $e_i$  in the last 10 seconds
22         $E_{prior}_i \leftarrow$  encode( $E_{prior}_i$ ); //encode the events to 0s and 1s
23         $F_{prior}_i \leftarrow$  collect frequency readings that happened prior to  $e_i$  in the last 10 seconds
24         $F_{prior}_i \leftarrow$  scale( $F_{prior}_i$ ); //scale the data to enhance the machine
learning prediction and minimize the bias of the machine
learning
25         $x_i \leftarrow$  ( $E_{prior}_i, F_{prior}_i$ ); //events and frequency readings tuple
that is used as an input to the model  $M_2$ 
26         $L_2 \leftarrow$  Conv - LSTM10( $x_i$ ); //apply the detector-10 deep learning
model to decrease false negatives
27        while  $L_1$  or  $L_2$  are Abnormal do
28           $d_1 \leftarrow$  Random Delay0 < d ≤ 4seconds; //If any of them is abnormal add
a delay
29          do in parallel
30            wait( $d_1$ )
31            executeEvent( $e_i$ )
32          do in parallel
33            NotifyOperator(); //Report abnormal behavior to the
operator/utility
34        if  $L_1$  and  $L_2$  is not Abnormal then
35           $d_1 \leftarrow$  0; //If the prediction of both models did not show
abnormal behavior then stop the mitigation mechanism

```

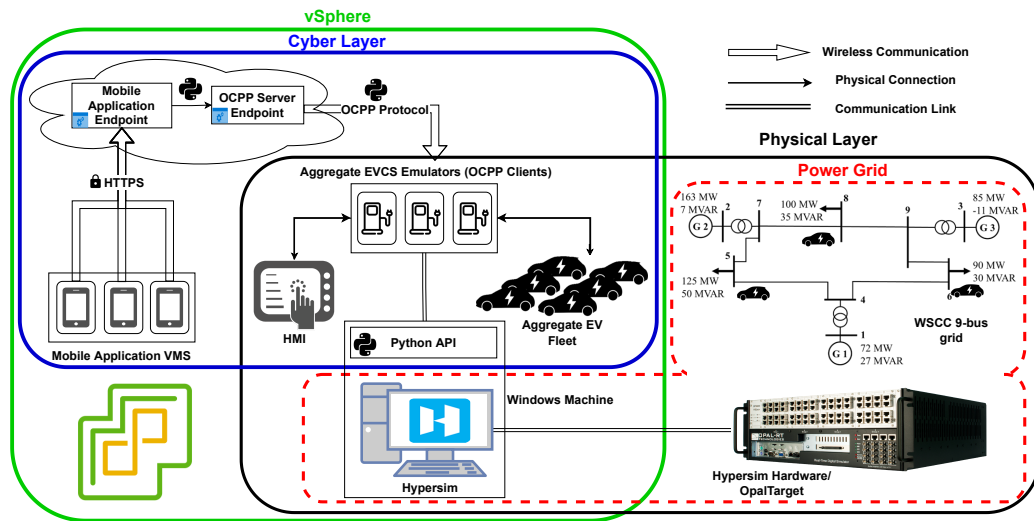


Figure 7.7: Co-simulation architecture [2]

7.2.3 Distributed Mitigation Methodology

This work aims to mitigate the impact of oscillatory load attacks in a distributed and lightweight manner and assist the power grid to return to its normal state easily.

Consequently, in this section, a lightweight and distributed mitigation mechanism against oscillatory load attacks is evaluated and discussed. After locally detecting the attacks within 5 seconds on an EVCS as discussed in the previous sections, a charging station can either discard a request or create a random delay by taking this decision independently. In [3], the authors proposed a centralized physical-layer mitigation technique that requires the utility to upgrade its existing generators with new control mechanisms. While the oscillations were successfully damped using their method, the oscillations on the power grid were never completely eliminated. Moreover, a centralized cyber-layer mitigation technique can only mitigate attacks launched through the cyber infrastructure of public charging stations. Due to the aforementioned limitations, a lightweight and distributed mitigation mechanism that can be deployed on the charging station itself (public and/or private) is proposed. After detecting attacks, each station independently creates a random delay for all incoming requests to break the attacker’s synchronization ability and hence minimize the impact on

the grid after a persistent attack. The main goal is to create a lightweight and distributed cyber-layer mitigation mechanism that aligns with the EVCS ecosystem deployment. The charging stations are characterized by low computing power which motivated the need for an independent mitigation mechanism. Consequently, the effectiveness of using the random delay to mitigate the impact of forced oscillations on the power grid is studied. In future work, with the aim to minimize the delay, we plan to utilize deep learning techniques that tailor the delay to each attack type based on the behavior and the load on the grid. In Algorithm 1, the conceptual model of the algorithm and how it is integrated with the mitigation mechanism is described. The detection is an online algorithm that runs in real time to detect oscillatory load attacks. The detector-5 detection model detects attacks early and the detector-10 detection model detects any false negative data samples that resulted from the first step. This two-step continuous detection technique is implemented to lower the false negative impact on the power grid and provide continuous monitoring over the ecosystem, when the first detection technique detects an attack it creates a random delay for any new request ranging between 0 and 4 seconds. However, this delay is removed if the detector-5 and detector-10 deep learning models stop classifying the rolling windows as attacks. Consequently, the detector-10 model can then be utilized to minimize the number of false negatives that might have been misclassified by the detector-5 deep learning model.

7.3 Experimental Results

As described in section 7.2.2, this work focuses on oscillatory load attacks initiated by exploiting vulnerabilities in the EV charging ecosystem. Periodic and stealthy attacks were studied where an attacker with enough resources can group charging stations and alternate the switching between different groups leading to inconspicuous behavior on the charging station.

7.3.1 Distributed Detection Mechanism Results

The proposed multivariate time series representation is used by taking the temporal and spatial relationship of the two features (events and frequency) into consideration. Several deep learning models were implemented such as LSTM for temporal relationships and ConvLSTM from Spatiotemporal relationships between the multivariate time series. The proposed models are tested on detector-5 and detector-10.

DATA PRE-PROCESSING: To feed the data to the deep learning algorithms, the features are encoded by converting the events (start and stop) to numerical values. Moreover, the frequency readings are normalized by re-scaling the data and fitting all the frequency data points between 0 and 1. To preserve the shape of the original distribution and the information embedded in it, the normalization can be represented as follows:

$$x_{inormalized} = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (1)$$

Where x_i is any value from the feature x (e.g., frequency), $\min(X)$ is the minimum value from the feature, and $\max(X)$ is the maximum value of the feature. MinMaxScaler is used to normalize the frequency feature vectors and obtain $x_{inormalized}$. Finally, each class (normal and abnormal) is mapped using binary label encoding, to 0 or 1. After that, the data is split into training (80%) and testing (20%) subsets.

MODEL SELECTION AND EVALUATION: Different deep-learning models are studied to classify oscillatory load attacks based on behavioral events on the charging station and their consequent effect on the power grid as represented by the frequency readings. The structure and layers of these models are described in the following sub-sections, along with the evaluation results. Moreover, the Adam Optimizer [164] is chosen for this classification problem. Adam outperforms other optimizers, such as Root Mean Square Propagation (RMSprop) and Adaptive Gradient Algorithm (AdaGrad), because of its bias-correction which helps Adam towards the end of the optimization as gradients become sparser [165].

Systematic enhancement of the outcomes requires iterative layer addition to a simple model (fewer layers) until a relatively good fit is reached (no under-fitting or over-fitting). Consequently, hyper-parameter tuning is performed using the Random Search algorithm. Finally, the hyperparameters that yielded the highest F-measure among the runs are identified. The parameters tuned are the learning rate in the Adam optimizer, the proportion of drops, the number of neurons in a layer, the size of batches, the number of epochs, filter size, and kernel size. Finally, the speed of each model is evaluated as a measure of their computational performance and the training time to measure the complexity of the model.

HYPER-PARAMETER TUNING AND APPLIED RANDOM SEARCH: It is worth mentioning different techniques were applied to decrease overfitting and achieve a good fit on training data. Dropping techniques were used, which refer to dropping out/ignoring units (i.e., neurons) during the training phase with a certain probability [166]. Moreover, batch normalization technique is used to stabilize the distribution of inputs (over a mini-batch) to a given layer during training. This helps in dramatically reducing the number of training epochs required to train deep networks [167]. After applying hyper-parameter tuning to find the best parameters to achieve a good fit since there exists a large number of variables that can be tuned to enhance the training. Finally, binary cross-entropy as a loss function is used.

For each model, a Random Search is applied and a refined Random Search algorithm to tune the hyperparameters that will yield the best results. In a deep learning model, various parameters contribute to finding the best fit for the training data (e.g., learning rate, decay, batch size, etc.). The Random Search algorithm uses a random combination of these values and trains the deep learning model on all these combinations. In the first step of the 2-step Random Search, 500 different combinations of these hyperparameters are generated, and selected the combination that resulted in the highest F1 score. The random search uses all of these 500 combinations and trains the deep learning model 500 times. The model and

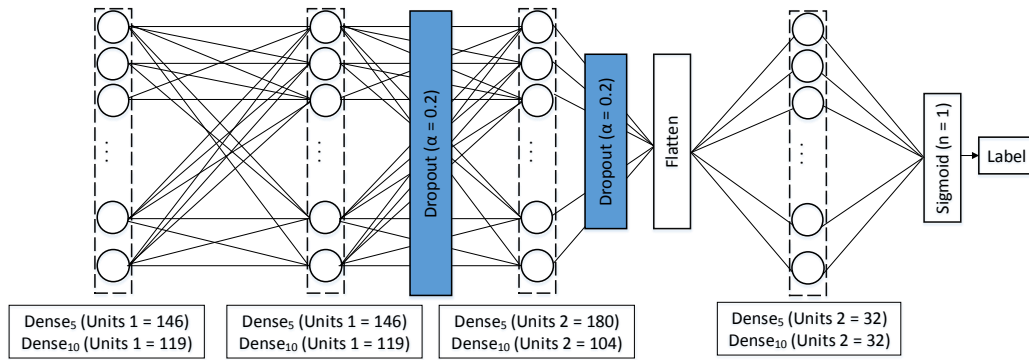


Figure 7.8: Structure of the Long-Short Term Memory Model.

the parameters that achieved the highest F1 score were selected. For refined results, in the second step of (refined) Random Search 100 different combinations of hyperparameters are generated in the 10% realm of the ones found in the first step, to try enhancing the results further. The random search performs similarly to meta-heuristics and grid search, however, with a lower computational cost [168].

In the following subsections, the analysis of the performance of the deep learning algorithms that were implemented to detect switching attacks is presented. The deep learning model that shows the highest F-measure score is selected, as it is more representative of the false negatives and false positives in the data. Moreover, the judgment is based on the number of false-negative, where an attack is misclassified as normal.

LONG-SHORT TERM MEMORY (LSTM): The LSTM model as a benchmark against other spatiotemporal deep learning models. The developed LSTM architecture is depicted in Figure 7.8 and consists of the following layers:

- **Input Layer:** The input of the network is 240 x 2 of encoded events and frequency reading collected over time.
- **LSTM Layer:** This is the main building block of an LSTM deep neural network and is responsible for learning the order dependency in the feature space.
- **Fully Connected Layers 1 and 2:** After the LSTM layer, a fully connected layer is added

with a Leaky ReLU activation function. To decrease overfitting batch normalization is applied along with a dropout. The output is then fed into another fully connected layer with a Leaky ReLU activation function. The Leaky ReLU activation function was developed to overcome one of the major shortcomings of the ReLU activation function. The ReLU activation function faces the "Dead ReLU" issue that occurs during backpropagation when no learning happens as the new weight remains equal to the old weight. Followed by batch normalization and a dropout layer. The output of these layers is three-dimensional, consequently, a copy of the output is collapsed into one dimension.

- Fully Connected Layer 3: The one-dimensional output is then fed to a connected layer initialized by a truncated normal distribution. Consequently, the last layer combines the features learned in previous layers and applies a Sigmoid function to output a probability between 0 and 1 that indicated the class of the data samples.

The LSTM Algorithm's optimal hyper-parameters achieved through the refined Random Search are presented in Table 7.1. Based on the optimal number of training epochs and the optimal batch size, the detector-5 and detector-10 LSTM algorithms were trained over 714 and 1200 iterations respectively. After running the random search (500 runs), a relatively good accuracy is achieved (97.5%) and F-measure scores (97.493%) on the detector-5 dataset. Moreover, a 500-run random search is run to tune the LSTM model on the detector-10 dataset which achieved a better accuracy (99.4%) and F-measure scores (99.405%) as shown in Table 7.2. In the experiments, it is crucial to look at how well the model classified attacks. The confusion matrix (Table 7.3) shows that using the detector-5 dataset, 972 attack samples were classified correctly, whereas 40 data samples were misclassified. Moreover, using the detector-10 dataset, 1002 attacks out of the 1012 attacks were classified correctly using the proposed novel Long Short-Term Memory deep learning model, and around 1% of the attacks were incorrectly classified. This confirms that oscillatory load attacks need more in-depth analysis to improve the accuracy of the model

Table 7.1: Optimized Hyper-Parameters for the Implemented Models

	detector-5		detector-10	
	LSTM	ConvLSTM _{2D}	LSTM	ConvLSTM _{2D}
Learning Rate	0.014717	0.0001939	0.00070810	0.0001
Drops	0.34	0.2	0.2	0.18
Batches	56	30	40	34
Units 1	146	150	119	176
Units 2	180	32	104	16
Units 3	32	-	32	-
Epochs	5	6	6	7
Filter 1	-	4	-	5
Kernel Size 1	-	6x6	-	5x5
Filter 2	-	8	-	8
Kernel Size 2	-	5x5	-	5x5

especially since the detector-5 dataset achieved a high false negative (3%). It is worth mentioning, that the smaller the attack window viewed by the detection system the earlier attacks are detected. Moreover, it is noted that the behavior of charging stations in normal conditions has some similarities with charging stations under stealthy attacks. The attacker, in stealthy attacks, tries to mimic the normal behavior of a charging station by dividing the switching behavior across different groups of charging stations and alternating the switching between them, which could be the reason for such misclassification. Although the misclassification is not huge, the impact that might occur out of successful attacks would cause a devastating impact on the power grid that could lead to tripping lines and overloading generators.

The proposed models were evaluated based on the training time and the time to make a prediction on the 20% test set provided. The training time of the best-performing LSTM deep learning model is 4 minutes for both datasets. The time to train the model is a good indicator of the complexity of the model and the resources needed for future enhancements. Moreover, the time to predict the labels of the 2000 sample test set for the detector-5 dataset and detector-10 dataset is 6 and 12 seconds, respectively, which equates to 0.003 seconds and 0.006 seconds on average for each data sample. The speed of each model is a measure of its computational performance.

CONVOLUTIONAL LONG-SHORT TERM MEMORY (CONVLSTM2D): ConvLSTM

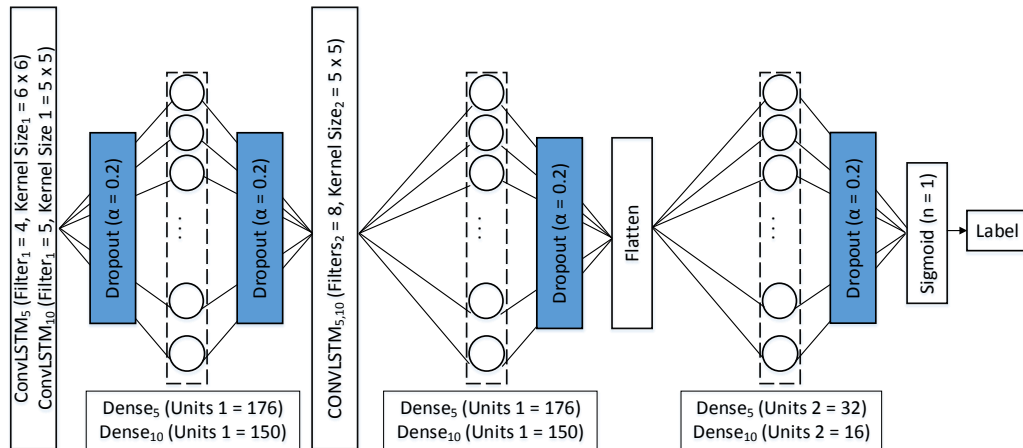


Figure 7.9: Structure of the Convolutional Long-Short Term Memory Model.

Table 7.2: Classifiers Outcomes

	detector-5		detector-10	
	LSTM	ConvLSTM _{2D}	LSTM	ConvLSTM _{2D}
Accuracy	97.500	99.400	99.400	99.800
F-measure	97.493	99.405	99.405	99.803
Recall	96.047	99.111	99.012	99.901
Precision	98.982	99.702	99.801	99.704

which is an LSTM variant. ConvLSTM is a type of recurrent neural network for multivariate spatiotemporal detection. It has convolutional structures that combine the ability of a convolutional neural network to incorporate spatial and temporal correlations into modeling and automatically capture the shared structures across variables (events and frequency). The developed ConvLSTM architecture, depicted in Figure 7.9, consists of the following layers:

- Input Layer: The input of the network is 240 x 2 of encoded events and frequency reading collected over time.
- ConvLSTM Layer 1: This is the main building block of the ConvLSTM deep learning network and is responsible for finding spatiotemporal relationships in the multivariate time series. To decrease overfitting batch normalization is applied with a dropout.

Table 7.3: Confusion Matrices for LSTM and ConvLSTM

		detector-5				detector-10			
		LSTM		ConvLSTM _{2D}		LSTM		ConvLSTM _{2D}	
		N	A	N	A	N	A	N	A
N		978	10	985	3	986	2	985	3
A		40	972	9	1003	10	1002	1	1011

Table 7.4: Classifiers Time

		detector-5		detector-10	
		LSTM	ConvLSTM _{2D}	LSTM	ConvLSTM _{2D}
Training Time		0:04:25	1:45:28	00:04:31	02:27:53
Prediction Time (s)		0.003	0.011	0.006	0.014

- Fully Connected Layer 1: After the ConvLSTM layer, the output is fed into a fully connected layer with a leaky ReLU activation function and followed by batch normalization and dropout.
- ConvLSTM Layer 2: The output is then fed into a second convolutional LSTM layer to derive further Spatiotemporal correlations from the features. Consequently, followed by batch normalization and a dropout layer. Moreover, the output is then flattened into a one-dimensional vector of numbers.
- Fully Connected Layer 2: The one-dimensional output after flattening is then inputted into a fully connected layer initialized by a truncated normal distribution. Consequently, the last layer combines the learned weights in previous layers and applies a Sigmoid function to output a probability between 0 and 1 that indicates the class of the data samples.

The ConvLSTM Algorithm’s optimal hyper-parameters achieved through the refined Random Search are presented in Table 7.1. Based on the optimal number of training epochs and the optimal batch size, the detector-5 and detector-10 ConvLSTM algorithms were trained over 1602 and 1652 iterations respectively. After running the random search (500 runs) on the detector-5 dataset, good accuracy (99.4%) and F-measure scores (99.405%) are

achieved. Moreover, the deep ConvLSTM model on the detector-10 dataset were tested and achieved better accuracy (99.8%) and F-measure score (99.803%). The confusion matrix for both datasets of the ConvLSTM model is depicted in Table 7.3. The classifier was able to correctly classify 985 out of 988 normal samples and 1003 out of 1012 attack samples. The performance of the classifier improved as the attack window in the detector-10 dataset increased. This shows that 0.99% and 0.099% of the attacks were misclassified for both datasets, as compared to the LSTM that achieved a 3.952% and 0.99% false negative rate. It is important to note that the number of misclassified attacks (false negatives) is an important indicator in choosing the best model to detect oscillatory load attacks. The risk of oscillatory load attacks is increasing as a result of the rapid deployment of charging stations [17]. It is acknowledged that the current deployment (number) of charging stations does not allow attackers to impact the power grid, however, with the current advancement and push towards electrifying the transportation system that is being enforced by governments such attacks will entail great risk.

Consequently, the training time of the best ConvLSTM model is evaluated and accumulated to 2 hours approximately. The time consumed during training is substantial compared to the LSTM. This result is expected due to the increase in the number of training parameters tuned (e.g., filter and kernel sizes) that will allow the model to perform convolutional techniques on the input data to extract spatiotemporal relationships. The LSTM variant is labor-intensive in terms of training. However, the computational time of the ConvLSTM model is 22 and 28 seconds which amounts to 0.011 and 0.014 seconds on average per data sample for the detector-5 and detector-10 datasets.

7.3.2 Distributed Mitigation Results

To evaluate the distributed mitigation mechanism, various oscillatory load attacks are launched to study the impact on the grid on the 9-Bus system, which is a simplified abstraction of the Western System Coordinating Council (WSCC) [169] grid in North America. This grid model is a 3-phase balanced transmission grid model. The test-bed setup is restrained to the 9-Bus system however, the general behavior of the different power grids is similar. Thus, the mitigation mechanism is easily reproducible on different power grids. After detecting the attacks within 5 seconds, every charging station adds a random delay to every request with the aim of depriving the adversary of the ability to synchronize attacks on multiple charging stations. Random delays are introduced with a maximum of 4 seconds that are chosen based on the sensitivity analysis performed in [170], which implies a decrease in user performance, and thus behavioral intentions begin to flatten when the delays extend to 4 seconds or longer in web interfaces. Moreover, according to [171, 172, 173], a 4-second delay is deemed tolerable and can even increase to 10 seconds before heavily impacting the user experience. Similarly, in [3] the authors suggested discarding a request if an attack is detected. However, if a false positive attack is detected, the quality of service with respect to a valid customer is affected, which would lead to user frustration. As a consequence, a random delay topped at 4 seconds is used to preserve the quality of service. Moreover, if the maximum delay is increased from 4 to 10 seconds a similar reduction in attack impact is observed. However, 4 seconds was utilized due to its ability to eliminate the forced frequency oscillations caused by a persistent oscillatory load attack while reducing the impact on the overall user experience. This is especially true in the case of false positives where the legitimate user should not be subjected to a large delay. Users will not tolerate substantial delays between initiating or stopping a charging session and the actual start or stop of the session. A maximum delay of 4 seconds fits the guidelines provided by various studies that evaluate and benchmark the acceptable response time when designing

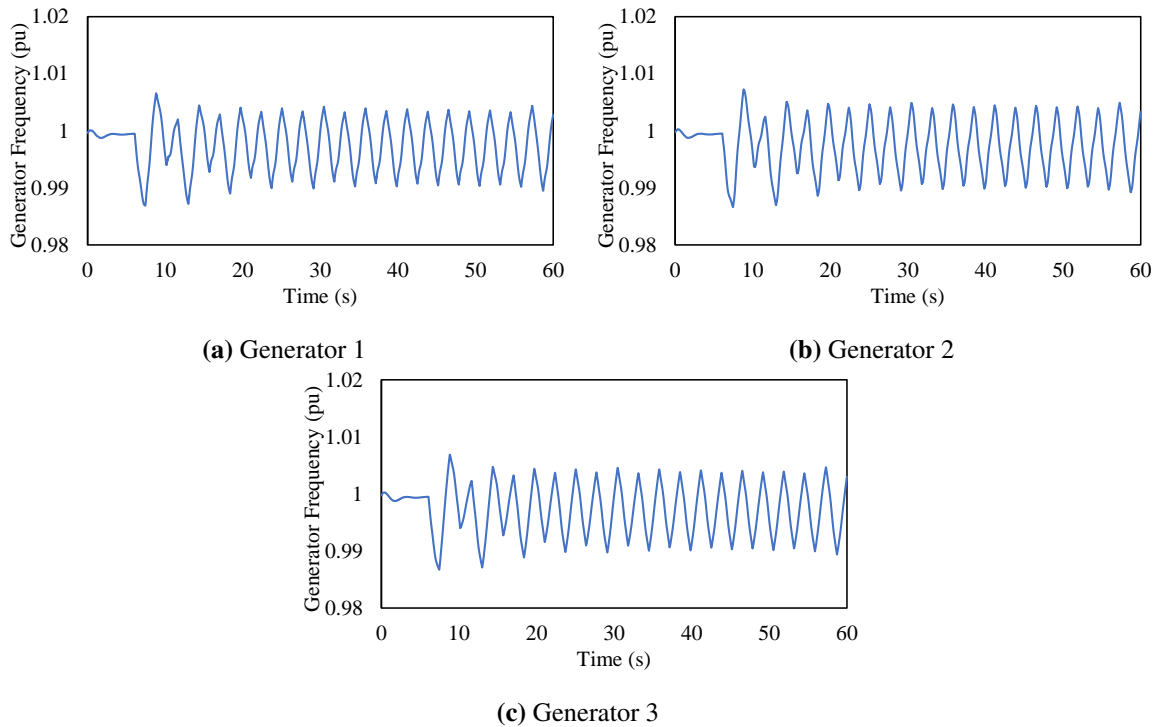


Figure 7.10: The variation of the generator’s speed as a result of an oscillatory load attack.

User Interfaces (UX/UI) in different contexts without affecting user experience [171, 172]. Norman Nielsen Group are world leaders in UX/UI and has identified that the user can tolerate a 0 to 10 seconds delay making the 0 to 4 seconds a tighter bound. Moreover, other more recent studies that focused on the opportunities and limitations afforded by online tools/interface providers suggest the five-second rules as a design guideline for moderated test sessions, which is still longer than the maximum delay of 4 seconds [173].

In what follows, an EV attack equivalent to 84MW load on one bus is demonstrated. This attack is equivalent to about 7636 EVs charging at the 11kW Level 2 chargers. Although such a number might be relatively high, the growth in the EV numbers will soon provide a large enough surface to make it possible [17]. Relative to today’s average charging rate of 24kW, the attacker would only need to compromise 3500 EVCS. Moreover, as the EVCS market moves towards wide adoption of level 3 chargers, the number of needed

compromised charging stations decreases. Level 3 chargers are DC fast chargers that deliver a charging rate of 40kW to 360kW, which means that to perform the same attack scenario 2100 EVs charging at 40 kW or as little as 233 charging at 360kW superchargers are needed.

Now, oscillatory load attacks take advantage of load manipulation to impact the frequency stability of the power grid. This attack revolves around the concept of creating a demand surge to cause a frequency drop on the grid followed by a drop in demand to cause the frequency to overshoot. The adversary uses the compromised load to create an imbalance between the increased load and the generated power, causing the generators to slow down, hence resulting in a frequency drop. Consequently, the attacker switches off the compromised load to cause an increase in the frequency and the generator speed in response to the adversary's actions. The attacker alternates between charging and stopping or discharging to disturb and impact the grid. The variation of the power generation speed due to an oscillatory load attack with a 2.4 seconds period (1.2 seconds on and 1.2 seconds off) is demonstrated in Figure 7.10. The sustained attack hinders the system's recovery causing fluctuations in the speed that would damage the turbines and decrease their lifetime due to the constant acceleration and deceleration. Different attacks could be launched by an adversary, however, a random delay between 0 to 4 seconds encompasses the wide range of attacks and is enough to mitigate this family of attacks. The switching attack demonstrated is persistent through the whole 60-second window of the simulation. The attack impact was neutralized even though the attacker continues the coordinated on/off behavior. The attack is no longer as impactful as before due to depriving the adversary of the ability to synchronize the actual behavior of the EVCSs. Thus, the behavior of each charging station would differ from the other as a result of the randomized delay. Thus, the total aggregate load, as demonstrated in Figure 7.11, fluctuates slightly around 50% of the total aggregate

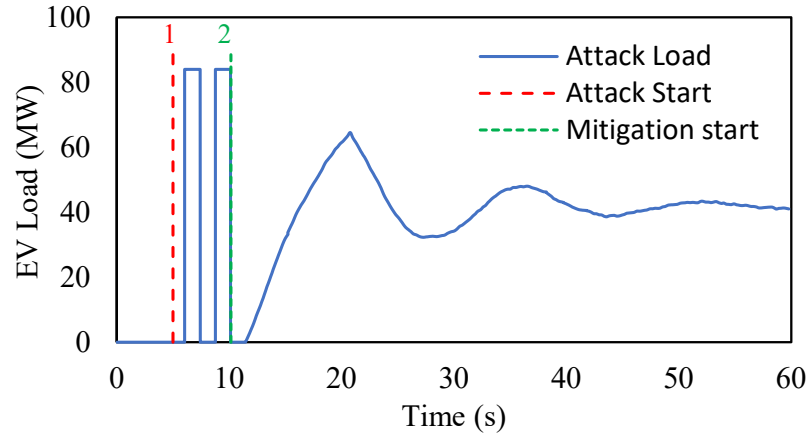


Figure 7.11: Load profile after mitigation.

attacker EVCS load. In future work, we will work on a mitigation mechanism that will utilize deep learning techniques to classify attacks based on their duty cycle to create a smart mitigation mechanism that minimizes the random delay needed.

As shown in Figure 7.11, the attack was launched (step 1) and detected (step 2) after 5 seconds using the proposed novel detection mechanism and consequently, the different charging stations independently initiate their mitigation mechanism and induce a random delay between 0 and 4 seconds to attenuate the impact on the power grid and the generators and prevent attacker synchronization. The main aim here is to stabilize the system by ensuring the disturbance does not impact the system’s performance. Figure 7.11 illustrates the attack load and its variation over time after implementing the proposed mitigation technique on the test bed (behavior of the grid after step 2). In the first few seconds, two peaks are observed showing the switching on and off due to an attack before detection, followed by a gradual distribution of the attack load over a period of 50 seconds as a result of the random delay of consequent start and stop requests. It is clear in Figure 7.11 how the added random delay causes the discrete behavior of the attack load, especially beyond the 40s. The frequency variation after a mitigated attack is shown in Figure 7.12,

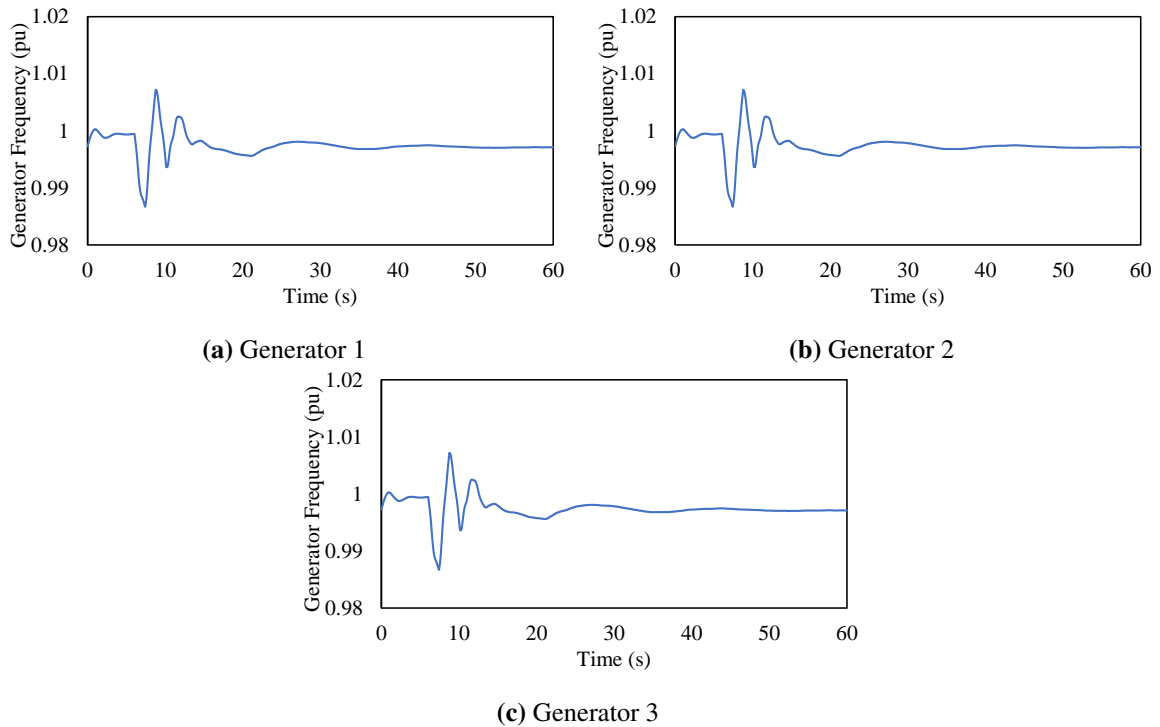


Figure 7.12: The variation of the generator’s speed as a result of an oscillatory load attack followed by mitigation.

which demonstrates the effectiveness of the proposed approach in eliminating the oscillations caused by the same sustained attack that is shown in Figure 7.10. To demonstrate the worst-case scenario that all the attacks are detected by the end of the first 5s of the attack and not before, the initiation of the mitigation technique is delayed till $t=10$ s which is 5 seconds after the attack started. This lightweight mitigation scheme prevents the attacker from synchronizing an attack thus, eliminating the attacker’s ability to impact the power grid. The randomization of the attack load over time results in a gradual increase in the load rather than instantaneous spikes and drops which allows the grid’s generators to cope with the change in demand (behavior after step 2). Along with that, as shown in Figure 7.12, the generator speed starts to get damped at $t=11$ s in 1s of detecting the attack. It is worth mentioning that the system requires 2s to return back to its normal frequency range after the mitigation mechanism is executed. The mitigation strategy was successful in eliminating the impact of the attack on the power grid without the need for adding a physical

control layer to the power grid. Furthermore, the proposed cyber-layer mitigation scheme is comparable to physical layer mitigation schemes. In [3], the control scheme was able to limit the forced oscillation to a safe threshold after 15s of the attack initiation as compared to the 2s. Furthermore, the proposed mitigation mechanism is able to completely eliminate the attack impact reducing the need for continuous acceleration and deceleration of the generators, unlike physical layer mitigation mechanisms that can dampen the dangerous oscillations but can never eliminate them.

Figure 7.11 demonstrates that after detection, the attack load would increase gradually to reach around 48 MW (half the total attack load). This would result in the grid's ability to return to stability since the attack load oscillations have been eliminated. It is worth mentioning that if the attack utilizes the V2G ability of the EVCSs, the proposed mitigation would result in an EV attack load centered around zero, which is even better for the stability of the grid and allows the generator to regulate their speeds easily.

7.4 Evaluation, Comparison, and Discussion

To detect oscillatory load attacks, an approach that leverages the behavioral characteristics of the charging station and the power grid is proposed. This approach is used to detect oscillatory load attacks initiated from the EV charging ecosystem regardless of the specific exploits and vulnerabilities used to compromise the different components that constitute it. The events at the charging station are directly related to the behavior of the power grid, making them the most important data on the cyber layer of the ecosystem.

7.4.1 Discussion of Obtained Results

Furthermore, the result demonstrated that such coordinated attacks have a unique signature constituting the charging event and the frequency on the power grid. Moreover,

some attacks disguised as normal behavior (stealthy attacks) might go undetected. Based on the previous experiments, the choice of features significantly impacted the accuracy and allowed the detection of oscillatory load attacks with high recall and precision values. The detection system on the detector-5 dataset (5 seconds attack window) and the detector-10 dataset (10 seconds attack windows) are evaluated. The precision and recall for LSTM and ConvLSTM improved as the attack window increases from 5 seconds to 10 seconds, as summarized in Table 7.2. The LSTM was less effective on the detector-5 dataset and achieved an F-measure score of 97.493%. However, the LSTM performance improved on the detector-10 dataset. It is observed that the improvement for other metrics studied (accuracy, precision, recall, and a number of false negative samples). The number of misclassified samples decreased from 40 to 10 when the LSTM model viewed a 10-second attack window, as shown in Table 7.3. Moreover, the detector-5 LSTM model misclassified stealthy and non-stealthy attacks whereas, the ConvLSTM only misclassified a few stealthy attacks achieving a low false negative on stealthy attacks with less than 1.7% (0.88% of total attacks). The same stealthy attacks were later detected in the detector-10 ConvLSTM model decreasing the misclassified stealthy attacks to 0.19% (0.099% of total attacks) using the detector-10 detector. Moreover, the other types of non-stealthy attacks were all detected using detector-5 ConvLSTM. This shows that detecting stealthy oscillatory load attacks is not as trivial as other types of attacks and a two-step detection mechanism is best suited to provide redundancy and effectiveness to the detection mechanism to help secure the vital services provided by the power grid. It is worth noting that the detection mechanism works on a rolling basis as long as the charging station is receiving requests, thus, improving the detection ability of the proposed approach. The rolling window detector-5 detector will detect attacks in any smaller time frame but it is worth highlighting that it was optimized for detection after 5 seconds to be able to detect stealthy attacks. The detector-5 detector was tested (without retaining) on windows containing only the first second of the

attack behavior, and it was able to identify a third of the attacks within 1s of their initiation, achieving a recall of 33.3%. In this test, only 3 normal data samples were misclassified as attacks which are consistent with the results of the original detector-5 model. In the stealthy version of the attacks, the attacker launches slow oscillatory attacks as well as distributes the switching among multiple charging stations. This makes the behavior of individual EVCSs in short windows of time look normal since only one event or possibly no events at all occur. Indeed, this is aligned with the finding where the detector-5 model misclassified the stealthy attacks within the first second and labeled them as normal. This shows the need for a rolling window model. Consequently, the detector-5 model was chosen to avoid a compromise between the impact on the grid and the accuracy of the model. In a real-life deployment, the distributed mitigation mechanism in 1 will start mitigating the attacks as soon as the detector classifies a window as an anomaly and does not need to wait for 5 seconds after the attack is initiated.

Moreover, the deep learning model ConvLSTM, an LSTM variant, achieved a better performance than the LSTM on the two datasets. The ConvLSTM showed improvement over LSTM on the detector-5 dataset achieving a 99.405% F-measure score. Moreover, the ConvLSTM also achieved a 99.803% F-measure score on the detector-10 dataset. It is important to note that the performance of the model detector-5 dataset is crucial for the evaluation since early detection of the attack is needed. The LSTM did not perform as well on the detector-5 dataset compared to the detector-10 dataset, whereas the ConvLSTM using the convolutional filters on the input allowed the deep learning model to learn intricate patterns of the attack data which allowed it to effectively classify samples. The ConvLSTM substitutes the matrix multiplication of the LSTM at each gate with convolutional operations that allowed it to extract the spatiotemporal relationships between the multiple timesteps over recorded variables (events and frequency). This relationship is the most crucial aspect in detecting oscillatory load attacks where ConvLSTM showed improved

performance over the LSTM [174]. Moreover, the fully connected LSTM has to unfold the inputs to 1D vectors before processing them, thus losing all the spatial information during the process. Thus, to preserve the spatial features the ConvLSTM uses 3D tensors that preserve the spatial information and determine the future state of a cell by taking into consideration the local neighbors of a cell [162]. The ConvLSTM reaps the benefits of the LSTM with temporal data and the benefits of a convolutional neural network with spatial data, which was important in this study of the two features over multiple timesteps. The ConvLSTM was able to learn the patterns of the attack with only 5-second windows. Indeed, the performance of both classifiers on the detector-10 dataset is expected to be better, since the impact of the switching would increase tremendously showing a significant change in the behavior. Moreover, the number of misclassified samples, most importantly the false negatives, is crucial to evaluate the efficiency of the detection model and how much the classifiers can be trusted to identify attacks. The analysis presented in Table 7.3 shows that the LSTM misclassified 40 and 10 attack samples as normal for the two datasets, which allows the adversary to execute a wider range of adversarial attacks as compared to the ConvLSTM which only misclassified 9 and 1 data samples for the detector-5 and the detector-10 datasets, respectively.

The ConvLSTM outperformed the LSTM in various aspects. However, training the ConvLSTM model took around two hours and a half as compared to the LSTM model which took about 4 minutes. The training time is tolerable in the system model because it is assumed that training is performed by a central authority with enough resources and has access to data from various operators and does not impact the prediction of attacks. Moreover, the prediction time of the ConvLSTM is still in the order of milliseconds which means that although its complex structure requires extra training time, its performance once deployed is not hindered by this complexity. The complexity of the ConvLSTM model arises from the structure of the ConvLSTM layer that utilizes matrix multiplication along

with the kernels and the filters used that increase the trainable parameters in the model drastically leading to high training time. Moreover, the increase in the training time of the ConvLSTM is also due to the batch size and the learning rate where increasing the batch size leads to a poor generalization over the data samples and a small learning rate requires more training epochs given the smaller changes made to the weights each update. In this approach a compromise was made between the training time and the accuracy to provide a reliable deep learning model that is able to effectively detect attacks. Moreover, the ConvLSTM and its convolutional mechanisms applied to features helped detect oscillatory switching attacks with as little as a 5-second attack window. Further, to compare the computational performances of the devised classifiers, their speed is measured in terms of the time required to complete the classification experiments. As illustrated in Table 7.4, the LSTM performed significantly faster than the convolutional LSTM, with 0.003 seconds to complete the classification. Whereas ConvLSTM performed relatively slower, with a computational time of 0.011 seconds. However, the time required by the ConvLSTM is tolerable since the output of the deep learning is almost instantaneous.

7.4.2 Comparison with Existing Detectors

The deep learning model depends on the behavioral characteristics (logged by the charging station during operation). The characteristics allow distributed decision-making where each charging station acts independently. To the best of our knowledge, we are the first to enable a distributed detection mechanism of oscillatory load attacks where models do not need to be deployed on a central management system to perform accurate detection. In [3], the authors devised a detection algorithm that depends on two charging events and the number of vehicles connecting within Δ time. The number of vehicles is an artifact that is only known to the CMS of a specific operator and is not shared. However, in this approach, the frequency reading of the power grid is utilized, which is a shared variable

(artifact) among the charging stations of different operators connected to the same bus. These features enable the detection of multi-operator and stealthy sophisticated attacks and distribute the detection mechanism. The deep learning model can be deployed on every charging station, where each EVCS can make its decision solely based on the artifacts (events and frequency) that can be collected by the EVCS independently.

Furthermore, since this approach is deployed on the charging station itself it prevents MitM attacks that can be launched by an adversary on the OCPP traffic exchange between the CMS and the EVCS itself [20] to control charging stations and perform oscillatory load attacks. The detection approach mitigates various attack vectors by deploying the deep learning model on the component that is used to create an impact (EVCS). It is worth mentioning, that the detection mechanism could be deployed on privately owned charging stations (i.e. EVCSs owned and operated by shopping centers, malls, and parking lot operators who might not be willing to share their utilization information with a central entity needed in [3]). Furthermore, the detection approach requires viewing only 5s of the oscillatory switching attacks as compared to [3] that was tested on 20, 30, and 40 seconds attack periods and resulted in 30%, 10%, 5% false negative rates, respectively. In Table 7.5 a detailed comparison between the decentralized proposed approach and the approach proposed in [3] is provided.

This work highlights that the cyber and physical layer features are a crucial component in enhancing the false negatives rate. The detection mechanism aided in discovering the relationship between cyber layer information represented by events and the physical layer which is represented by frequency recording of the bus that the EVCS is connected to. Moreover, stealthy attacks that utilize slow switching behavior might be erroneously classified by detection mechanisms relying on the cyber data alone. Thus, the spatiotemporal relationship of hybrid data coming from both layers is shown to be suitable to help solve the shortcomings of pure cyber-layer detection mechanisms. While the approach proposed

Table 7.5: Comparison between the decentralized approach and the centralized approach proposed in [3].

Key Differences	Centralized Detection/Mitigation (Approach Proposed in [3])	Decentralized Detection/Mitigation (Our approach)
Processing	All processing is done on a central server or a set of servers	Processing is distributed on the individual EVCSs
Data Collection	Data is collected from all public charging stations	Data is collected locally at each EVCS and does not require sharing with any third-party entity
Communication	Communication between the server and the EVCS is critical and requires a reliable and fast network connection	The EVCS takes decisions independently of each other without requiring communication and incurring any delays
Security	Detection is more vulnerable to attacks since it has a single point of failure and or compromise could bring down the entire system	Decentralized detection is more resilient and the adversary would need to compromise the all the EVCSs individually to remain successful
Accuracy	92%	99.4%
False Negatives	30%	0.8%
False Positives	<1% (not clearly stated)	0.3%
Speed of detection	20+ seconds	5 seconds

in [3] is aimed to solve a similar problem using a centralized detection mechanism the proposed approach in this work shows improvement over it in terms of resiliency and fault tolerance due to the deployment strategy followed.

7.4.3 Robustness and Limitations

In this approach, adversarial oscillatory attacks that other detection mechanisms fail to detect are addressed (e.g., multi-operator, stealthy, and MitM oscillatory load attacks). This improves the robustness of this model and increases the spectrum of various oscillatory load attacks that could be detected by this approach. Considering that the approach’s scope is only to detect coordinated oscillatory load attacks based on the combination of cyber and physical behavioral characteristics, the experiments were performed on the New England 39-Bus System. Data samples from different power grids were not included. Attacks on the grids have various impacts. For example, an attack on a 9-Bus system might not have the same consequences on a 39-Bus system. However, this approach is easily reproducible to make it operational on other power grids. Although this work contributes to understanding and detecting oscillatory load attacks, however, it faces a few current limitations. For instance, the work relies on a supervised learning approach, which cannot classify new,

previously unseen attacks. To overcome this limitation, unsupervised approaches can be considered complementary approaches to face the emergence of any adversarial attacks. Additionally, the proposed approach can be leveraged as a stepping stone to develop new cyber-layer defense mechanisms that prevent and detect oscillatory load attacks.

In this work, it is assumed that the charging station is honest, thus, the adversary can evade this detection mechanism by compromising the charging station itself. However, the adversary needs to compromise all the charging stations needed to mount attacks. The distributed nature of the detection mechanism makes it hard for the adversary to mount attacks easily and ensures fault tolerance in the system, unlike centralized detection mechanisms that provide a single point of failure. Moreover, adversaries need to hack and exploit charging stations with different firmware versions which would require the attacker to find vulnerabilities in the different types of charging stations. Finally, in future work we plan to use federated learning to assist in preserving the privacy of the records during the training period without the need for the power grid operator to get charging behavior data to create an AI-enabled detection model.

Moreover, it is important to note that to evade the mitigation technique, the attacker needs to guess the random number generator (if the operator/manufacturer used a weak random number generator). However, each charging station creates a random delay independently of the other hindering the adversary from discovering the random delay of all the charging stations that are being exploited to mount an oscillatory load attack. Through this work, it is shown that a simple and lightweight random delay mechanism provides an efficient countermeasure to adversaries trying to launch oscillatory load attacks. This mechanism is compatible with the nature of the ecosystem as it doesn't require coordination with the other charging stations which would create an overhead for charging stations that are equipped with limited computing power. However, we plan in the future work to create a framework to support the grid using V2G and mitigate the impact of EVCS cyber-attacks.

Chapter 8

Conclusion and Future Directions

The current security posture of the EV charging ecosystem requires an in-depth analysis of the various components. The wide spread of insecurity highlighted through this thesis sheds light on the need to raise the security bar for the stakeholders in this ecosystem to ensure a reliable ecosystem. In this thesis, we studied the security of the different components of the EVCS ecosystem. We created a real-time co-simulation platform that allows security researchers to study the security of the ecosystem as a whole, unlike previous work that simulated individual components. Consequently, we have created an advanced discovery mechanism to identify EVCSs in the wild to study their security we then use non-invasive techniques to evaluate the security posture concerning remote compromise. Finally, we also study the malware threat landscape and create the first baseline study. Moreover, we also study an understudied component the OCPP backend where we discover 6 zero-days in each of the 16 operators we studied which allows adversaries to launch covert attacks that impact the operator's visibility over the infrastructure among other attacks. Finally, we devise a distributed deep learning algorithm to identify oscillatory load attacks by carefully choosing features that enable the federated aspect of the decision-making. Additionally, we devise a distributed mitigation mechanism to prevent the adversary from synchronizing attacks which would allow them to create oscillatory load attacks. In the future, we plan

Table 8.1: Contributions during the Ph.D. Program

Title	Citation
A Real-Time Cosimulation Testbed for Electric Vehicle Charging and Smart Grid Security	Sarieddine, K., Sayed, M. A., Jafarigiv, D., Atallah, R., Debbabi, M., & Assi, C. (2023). A Real-Time Cosimulation Testbed for Electric Vehicle Charging and Smart Grid Security. <i>IEEE Security & Privacy</i> .
EV Charging Infrastructure Discovery to Contextualize its Deployment Security	Sarieddine, K., Sayed, M. A., Assi, C., Atallah, R., Torabi, S., Khoury, J., ... & Bou-Harb, E. (2023). EV Charging Infrastructure Discovery to Contextualize its Deployment Security. <i>IEEE Transactions on Network and Service Management</i> .
Investigating the security of EV charging mobile applications as an attack surface	Sarieddine, K., Sayed, M. A., Torabi, S., Atallah, R., & Assi, C. (2023). Investigating the security of ev charging mobile applications as an attack surface. <i>ACM Transactions on Cyber-Physical Systems</i> , 7(4), 1-28.
Uncovering Covert Attacks on EV Charging Infrastructure: How OCPP Backend Vulnerabilities Could Compromise Your System	Sarieddine, K., Sayed, M. A., Torabi, S., Attallah, R., Jafarigiv, D., Assi, C., & Debbabi, M. (2024, June). Uncovering Covert Attacks on EV Charging Infrastructure: How OCPP Backend Vulnerabilities Could Compromise Your System. In <i>Proceedings of the 2024 ACM on Asia Conference on Computer and Communications Security</i>
Edge-based detection and localization of adversarial oscillatory load attacks orchestrated by compromised EV charging stations	Sarieddine, K., Sayed, M. A., Torabi, S., Atallah, R., & Assi, C. (2024). Edge-based detection and localization of adversarial oscillatory load attacks orchestrated by compromised EV charging stations. <i>International Journal of Electrical Power & Energy Systems</i> , 156, 109735.

to create passive and highly interactive honeypots that imitate an EVCS to collect traffic and malware samples that would help us understand the threat landscape further. For future work, we plan to study EVCS backends that do not use the standard OCPP communication (e.g., Tesla) to identify the different attack vectors and threats that the ecosystem faces. Finally, devising a monitoring scheme for the ecosystem that would take into account the different components in the ecosystem (cloud, mobile app, backend, etc.).

The bulk of the report focuses on the work that has been performed by the student as part of the Ph.D. program. The different contributions have already been published/accepted for publication in top venues are summarized in Table 8.1. Other collaborations with different colleagues through out my Ph.D. are summarized in Table 8.2.

Table 8.2: Other Co-authorships during the Ph.D. program

Title	Citation
On ransomware family attribution using pre-attack paranoia activities	Molina, R. M. A., Torabi, S., Sarieddine, K., Bou-Harb, E., Bouguila, N., & Assi, C. (2021). On ransomware family attribution using pre-attack paranoia activities. <i>IEEE Transactions on Network and Service Management</i> , 19(1), 19-36.
A Data-Driven Framework for Improving Public EV Charging Infrastructure: Modeling and Forecasting	Al-Dahabreh, N., Sayed, M. A., Sarieddine, K., Elhattab, M., Khabbaz, M. J., Atallah, R. F., & Assi, C. (2023). A Data-Driven Framework for Improving Public EV Charging Infrastructure: Modeling and Forecasting. <i>IEEE Transactions on Intelligent Transportation Systems</i> .
Quality of Service Evaluation and Forecast for EV Charging Based on Real-World Data	Atallah, R., Al-Dahabreh, N., Sayed, M. A., Sarieddine, K., Elhattab, M., Khabbaz, M., & Assi, C. (2023, June). Quality of Service Evaluation and Forecast for EV Charging Based on Real-World Data. In <i>2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)</i> (pp. 280-285). IEEE.
Electric Vehicle Switching Attacks Against Subsynchronous Stability of Power Systems	Abazari, A., Sarieddine K., Ghafouri M., Atallah R., D. Jafarigiv, & Assi C. "Electric Vehicle Switching Attacks Against Subsynchronous Stability of Power Systems" submitted to <i>IEEE Transactions on Industrial Informatics</i>
PEACE: Physics-Enabled Autoencoder Detection of Unknown Load-Altering Attacks in Smart Grids	M. A. Sayed, Nathalie Wehbe, K. Sarieddine, R. Atallah, C. Assi, & M. Debbabi. PEACE: Physics-Enabled Autoencoder Detection of Unknown Load-Altering Attacks in Smart Grids. Submitted to <i>IEEE Power & Energy Society General Meeting (PESGM)</i> .
GridWatch: Load-Altering Attack Detection and Localization Mechanism Powered by a Physics-Assisted Feature Fusion Hybrid Neural Network	M. A. Sayed, K. Sarieddine, Nathalie Wehbe, M. Arfaoui, R. Atallah, M. Debbabi, & C. Assi. GridWatch: Load-Altering Attack Detection and Localization Mechanism Powered by a Physics-Assisted Feature Fusion Hybrid Neural Network. Submitted <i>IEEE transactions on Smart Grid</i> .

Bibliography

- [1] Cyberspace search engine. *ZoomEye*. URL <https://www.zoomeye.org/>.
- [2] Khaled Sareddine, Mohammad Ali Sayed, Danial Jafarigiv, Ribal Atallah, Mourad Debbabi, and Chadi Assi. A real-time cosimulation testbed for electric vehicle charging and smart grid security. *IEEE Security & Privacy*, 2023.
- [3] Mohammad Ekramul Kabir, Mohsen Ghafouri, Bassam Moussa, and Chadi Assi. A two-stage protection method for detection and mitigation of coordinated evse switching attacks. *IEEE Transactions on Smart Grid*, 2021.
- [4] Linda Gyulai. Montreal's climate plan includes ban on non-electric cars downtown by 2030, Dec 2020. URL <https://montrealgazette.com/news/local-news/montreal-releases-climate-plan-including-ban-on-non-electric-cars-downtown-by-2030>.
- [5] Helen Regan. China pledges to go carbon neutral by 2060, Sep 2020. URL <https://www.cnn.com/2020/09/22/china/xi-jinping-carbon-neutral-2060-intl-hnk/index.html>.
- [6] Charles Riley. Europe aims to kill gasoline and diesel cars by 2035, Aug 2021. URL <https://edition.cnn.com/2021/07/14/business/eu-emissions-climate-cars/index.html>.

- [7] Benjamin Shingler. Quebec’s push to go electric won’t get province to emission reduction targets, experts say, Nov 2020. URL <https://www.cbc.ca/news/canada/montreal/quebec-green-plan-1.5802976>.
- [8] Ahmed Yousuf Saber and Ganesh Kumar Venayagamoorthy. Plug-in vehicles and renewable energy sources for cost and emission reductions. *IEEE Transactions on Industrial electronics*, 58(4):1229–1238, 2010.
- [9] Natural Resources Canada. Government of Canada, Oct 2021. URL <https://www.nrcan.gc.ca/energy-efficiency/transportation-alternative-fuels/zero-emission-vehicle-infrastructure-program/21876>.
- [10] Science Innovation and Economic Development Canada. Canada strikes historic partnerships with leading German automakers Volkswagen and Mercedes to help meet gr..., Aug 2022. URL <https://www.canada.ca/en/innovation-science-economic-development/news/2022/08/canada-strikes-historic-partnerships-with-leading-german-automakers-volkswagen-and-mercedes-to-help-meet-growing-demand-for-clean-transportation-so.html>.
- [11] Andrew J. Hawkins. The electric car industry now has its own lobbying group, Nov 2020. URL <https://www.theverge.com/2020/11/17/21571747/zero-emissions-transportation-association-tesla-uber-rivian-lobby>.
- [12] Raphaelle Akhras, Wassim El-Hajj, Michel Majdalani, Hazem Hajj, Rabih Jabr, and Khaled Shaban. Securing smart grid communication using Ethereum smart contracts. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1672–1678. IEEE, 2020.

- [13] Analysis | u.s. government provides cyber budget specifics, Mar 2023. URL <https://www.washingtonpost.com/politics/2023/03/14/us-government-provides-cyber-budget-specifics/>.
- [14] Arielle Berger MacKenzie Sigalos. Gas shortages, price hikes reported amid colonial pipeline shutdown: Cnbc after hours, May 2021. URL <https://www.cnn.com/2021/05/11/gas-shortages-price-hikes-reported-amid-colonial-pipeline-shutdown.html>.
- [15] Lily Hay Newman. The solarwinds body count now includes nasa and the faa, Feb 2021. URL <https://www.wired.com/story/solarwinds-nasa-faa-robot-dog-fight-security-news/>.
- [16] Saleh Soltan, Prateek Mittal, and H Vincent Poor. {BlackIoT}:{IoT} botnet of high wattage devices can disrupt the power grid. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 15–32, 2018.
- [17] Mohammad Ali Sayed, Ribal Atallah, Chadi Assi, and Mourad Debbabi. Electric vehicle attack impact on power grid operation. *International Journal of Electrical Power & Energy Systems*, 137:107784, 2022.
- [18] Russian ev charging stations hacked., Mar 2022. URL <https://www.independent.co.uk/news/world/europe/putin-charging-station-hacked-ukraine-russia-b2026260.html>.
- [19] Tony Nasr, Sadegh Torabi, Elias Bou-Harb, Claude Fachkha, and Chadi Assi. Power jacking your station: In-depth security analysis of electric vehicle charging station management systems. *Computers & Security*, 112:102511, 2022.
- [20] Cristina Alcaraz, Javier Lopez, and Stephen Wolthusen. Ocpp protocol: Security threats and challenges. *IEEE Transactions on Smart Grid*, 8(5):2452–2459, 2017.

- [21] Juan E Rubio, Cristina Alcaraz, and Javier Lopez. Addressing security in ocpp: Protection against man-in-the-middle attacks. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2018.
- [22] Samrat Acharya, Yury Dvorkin, Hrvoje Pandžić, and Ramesh Karri. Cybersecurity of smart electric vehicle charging: A power grid perspective. *IEEE Access*, 8: 214434–214453, 2020.
- [23] Hossam ElHussini, Chadi Assi, Bassam Moussa, Ribal Atallah, and Ali Ghrayeb. A tale of two entities: Contextualizing the security of electric vehicle charging stations on the power grid. *ACM Transactions on Internet of Things*, 2(2):1–21, 2021.
- [24] Joseph Antoun, Mohammad Ekramul Kabir, Bassam Moussa, Ribal Atallah, and Chadi Assi. A detailed security assessment of the ev charging ecosystem. *IEEE Network*, 34(3):200–207, 2020.
- [25] Ocpp 2.0.1, protocols, home, 2021. URL <https://www.openchargealliance.org/protocols/ocpp-201/>.
- [26] YS Wong, KT Chau, and CC Chan. Battery sizing for plug-in hybrid electric vehicles. *Journal of Asian Electric Vehicles*, 4(2):899–904, 2006.
- [27] How do fuel cell electric vehicles work using hydrogen?, 2021. URL <https://afdc.energy.gov/vehicles/how-do-fuel-cell-electric-cars-work>.
- [28] Concetta Semeraro, AG Olabi, Haya Aljaghoub, Abdul Hai Alami, Muaz Al Radi, Michele Dassisti, and Mohammad Ali Abdelkareem. Digital twin application in energy storage: Trends and challenges. *Journal of Energy Storage*, 58:106347, 2023.
- [29] Yihong Li, Qi Tao, and Yadong Gong. Digital twin simulation for integration of

blockchain and internet of things for optimal smart management of pv-based connected microgrids. *Solar Energy*, 251:306–314, 2023.

- [30] Bowen Yang, Lulu Guo, and Jin Ye. Real-time simulation of electric vehicle powertrain: hardware-in-the-loop (hil) testbed for cyber-physical security. In *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*, pages 63–68. IEEE, 2020.
- [31] Anuj Sanghvi and Tony Markel. Cybersecurity for electric vehicle fast-charging infrastructure. In *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*, pages 573–576. IEEE, 2021.
- [32] Hidekuni Toda, Yutaka Ota, Tatsuhito Nakajima, Ken-ichi Kawabe, and Akihiko Yokoyama. Implementation and verification of v2g control schemes on multiple electric vehicles. In *2nd E-Mobility Power System Integration Symposium*, 2018.
- [33] *Shodan*. URL <https://www.shodan.io/>.
- [34] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by Internet-wide scanning. *22nd ACM Conference on Computer and Communications Security*, October 2015.
- [35] Tony Nasr, Sadegh Torabi, Elias Bou Harb, Claude Fachkha, and Chadi Assi. Chargeprint: A framework for internet scale discovery and security analysis of ev charging management systems. *NDSS*, 2023.
- [36] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. Acquisitional rule-based engine for discovering internet-of-things devices. pages 327–341, 2018.
- [37] Takayuki Sasaki, Akira Fujita, Carlos H Gañán, Michel van Eeten, Katsunari Yoshioka, and Tsutomu Matsumoto. Exposed infrastructures: Discovery, attacks and

- remediation of insecure ics remote management devices. *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2379–2396, 2022.
- [38] Takahiro Ueda, Takayuki Sasaki, Katsunari Yoshioka, and Tsutomu Matsumoto. An internet-wide view of connected cars: Discovery of exposed automotive devices. *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–8, 2022.
- [39] Andrei Costin, Apostolis Zarras, and Aurelien Francillon. Towards automated classification of firmware images and identification of embedded devices. In *ICT Systems Security and Privacy Protection: 32nd IFIP TC 11 International Conference, SEC 2017, Rome, Italy, May 29-31, 2017, Proceedings 32*, pages 233–247. Springer, 2017.
- [40] Xu Wang, Yucheng Wang, Xuan Feng, Hongsong Zhu, Limin Sun, and Yuchi Zou. Iottracker: an enhanced engine for discovering internet-of-thing devices. In *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–9. IEEE, 2019.
- [41] Dan Yu, Lilong Zhang, Yongle Chen, Yao Ma, and Junjie Chen. Large-scale iot devices firmware identification based on weak password. *IEEE Access*, 8:7981–7992, 2020.
- [42] Sklyar Dmitry. ChargePoint Home Security Research. https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/12/13084354/ChargePoint-Home-security-research_final.pdf, December 2018.
- [43] Cristina Alcaraz, Jesus Cumplido, and Alicia Trivino. Ocpp in the spotlight: threats

and countermeasures for electric vehicle charging infrastructures 4.0. *International Journal of Information Security*, pages 1–27, 2023.

- [44] Zacharenia Garofalaki, Dimitrios Kosmanos, Sotiris Moschoyiannis, Dimitrios Kallergis, and Christos Douligeris. Electric vehicle charging: a survey on the security issues and challenges of the open charge point protocol (ocpp). *IEEE Communications Surveys & Tutorials*, 2022.
- [45] Richard Baker and Ivan Martinovic. Losing the car keys: Wireless phy-layer insecurity in {EV} charging. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 407–424, 2019.
- [46] Sebastian Köhler, Richard Baker, Martin Strohmeier, and Ivan Martinovic. Demo: End-to-end wireless disruption of ccs ev charging. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22*, page 3515–3517, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505.
- [47] Kaibin Bao, Hristo Valev, Manuela Wagner, and Hartmut Schmeck. A threat analysis of the vehicle-to-grid charging protocol iso 15118. *Computer Science-Research and Development*, 33(1-2):3–12, 2018.
- [48] Mauro Conti, Denis Donadel, Radha Poovendran, and Federico Turrin. Evexchange: A relay attack on electric vehicle charging system. In *Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I*, pages 488–508. Springer, 2022.
- [49] Raju Gottumukkala, Rizwan Merchant, Adam Tausin, Kaleb Leon, Andrew Roche,

- and Paul Darby. Cyber-physical system security of vehicle charging stations. In *2019 IEEE Green Technologies Conference (GreenTech)*, pages 1–5. IEEE, 2019.
- [50] Khaled Sareddine, MA Sayed, Sadegh Torabi, Ribal Atallah, and Chadi Assi. Investigating the security of ev charging mobile applications as an attack surface. *arXiv preprint arXiv:2211.10603*, 2022.
- [51] Samrat Acharya, Yury Dvorkin, and Ramesh Karri. Public plug-in electric vehicles+ grid data: Is a new cyberattack vector viable? *IEEE Transactions on Smart Grid*, 11(6):5099–5113, 2020.
- [52] Eman Hammad, Ahmed M Khalil, Abdallah Farraj, Deepa Kundur, and Reza Iravani. A class of switching exploits based on inter-area oscillations. *IEEE Transactions on Smart Grid*, 9(5):4659–4668, 2017.
- [53] Mohsen Ghafouri, Mohammad Ekramul Kabir, Bassam Moussa, and Chadi Assi. Coordinated charging and discharging of electric vehicles: A new class of switching attacks. *ACM Transactions on Cyber-Physical Systems*, 2022.
- [54] Mohammad Ali Sayed, Mohsen Ghafouri, Mourad Debbabi, and Chadi Assi. Dynamic load altering ev attacks against power grid frequency control. In *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2022.
- [55] Cabell Hodge, Konrad Hauck, Shivam Gupta, and Jesse C Bennett. Vehicle cybersecurity threats and mitigation approaches. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2019.
- [56] Hang Du, Jun Yan, Mohsen Ghafouri, Rawad Zgheib, Marthe Kassouf, and Mourad Debbabi. Modeling of cyber attacks against converter-driven stability of pmsg-based wind farms with intentional subsynchronous resonance. In *2021 IEEE International*

Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pages 391–397. IEEE, 2021.

- [57] Omniyah Gul M Khan, Ehab El-Saadany, Amr Youssef, and Mostafa Shaaban. Impact of electric vehicles botnets on the power grid. In *2019 IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–5. IEEE, 2019.
- [58] Kristien Clement-Nyns, Edwin Haesen, and Johan Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on power systems*, 25(1):371–380, 2009.
- [59] Niels Leemput, Frederik Geth, Juan Van Roy, Annelies Delnooz, Jeroen Büscher, and Johan Driesen. Impact of electric vehicle on-board single-phase charging strategies on a flemish residential grid. *IEEE Transactions on Smart Grid*, 5(4):1815–1822, 2014.
- [60] Anamika Dubey and Surya Santoso. Electric vehicle charging on residential distribution systems: Impacts and mitigations. *IEEE Access*, 3:1871–1893, 2015.
- [61] Hugo Morais, Tiago Sousa, Zita Vale, and Pedro Faria. Evaluation of the electric vehicle impact in the power demand curve in a smart grid environment. *Energy Conversion and Management*, 82:268–282, 2014.
- [62] Soroush Shafiee, Mahmud Fotuhi-Firuzabad, and Mohammad Rastegar. Investigating the impacts of plug-in hybrid electric vehicles on power distribution systems. *IEEE Transactions on Smart Grid*, 4(3):1351–1360, 2013.
- [63] Elif Ustundag Soykan, Mustafa Bagriyanik, and Gurkan Soykan. Disrupting the power grid via ev charging: The impact of the sms phishing attacks. *Sustainable Energy, Grids and Networks*, 26:100477, 2021.

- [64] Mostafa Mohammadpourfard, Yang Weng, Istemihan Genc, and Taesic Kim. An accurate false data injection attack (fdia) detection in renewable-rich power grids. In *2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–5. IEEE, 2022.
- [65] Abdelrahman Ayad, Hany EZ Farag, Amr Youssef, and Ehab F El-Saadany. Detection of false data injection attacks in smart grids using recurrent neural networks. In *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2018.
- [66] Harag Margossian, Mohammad Ali Sayed, Wissam Fawaz, and Zahi Nakad. Partial grid false data injection attacks against state estimation. *International Journal of Electrical Power & Energy Systems*, 110:623–629, 2019.
- [67] Oliver Kosut, Liyan Jia, Robert J Thomas, and Lang Tong. Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid*, 2(4):645–658, 2011.
- [68] Shang Li, Yasin Yilmaz, and Xiaodong Wang. Quickest detection of false data injection attack in wide-area smart grids. *IEEE Transactions on Smart Grid*, 6(6):2725–2735, 2014.
- [69] Yufeng Wang, Zhihao Zhang, Jianhua Ma, and Qun Jin. Kfrnn: An effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network. *IEEE Internet of Things Journal*, 2021.
- [70] Jacob Sakhnini, Hadis Karimipour, Ali Dehghantanha, and Reza M Parizi. Physical layer attack identification and localization in cyber–physical grid: An ensemble deep learning based approach. *Physical Communication*, 47:101394, 2021.
- [71] Arash Moradzadeh, Mostafa Mohammadpourfard, Istemihan Genc, Şahin Serhat

- Şeker, and Behnam Mohammadi-Ivatloo. Deep learning-based cyber resilient dynamic line rating forecasting. *International Journal of Electrical Power & Energy Systems*, 142:108257, 2022.
- [72] Hwei-Ming Chung, Wen-Tai Li, Chau Yuen, Wei-Ho Chung, Yan Zhang, and Chao-Kai Wen. Local cyber-physical attack for masking line outage and topology attack in smart grid. *IEEE Transactions on Smart Grid*, 10(4):4577–4588, 2018.
- [73] Manoj Basnet and Mohd Hasan Ali. Deep learning-based intrusion detection system for electric vehicle charging station. In *2020 2nd International Conference on Smart Power & Internet Energy Systems (SPIES)*, pages 408–413. IEEE, 2020.
- [74] Manoj Basnet and M Hasan Ali. Exploring cybersecurity issues in 5g enabled electric vehicle charging station with deep learning. *arXiv preprint arXiv:2104.08553*, 2021.
- [75] Manoj Basnet, Subash Poudyal, Mohd Hasan Ali, and Dipankar 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)*, pages 1–5. IEEE, 2021.
- [76] Ricardo Misael Ayala Molina, Sadegh Torabi, Khaled Saredidine, Elias Bou-Harb, Nizar Bouguila, and Chadi Assi. On ransomware family attribution using pre-attack paranoia activities. *IEEE Transactions on Network and Service Management*, 2021.
- [77] Mobilityhouse. Mobilityhouse/ocpp: Python implementation of the open charge point protocol (ocpp). URL <https://github.com/mobilityhouse/ocpp>.
- [78] . URL <https://aemo.com.au/>.
- [79] . URL <https://www.abs.gov.au/statistics/industry/tourism-and-transport/motor-vehicle-census-australia>.

- [80] IEA. Global ev outlook 2022 – analysis. IEA, 2022. URL <https://www.iea.org/reports/global-ev-outlook-2022>.
- [81] NLTK. *NLTK*, June 2022. URL <https://www.nltk.org/>.
- [82] Chargemap. Charging stations for electric cars. *Chargemap*. URL <https://chargemap.com/>.
- [83] plugshare. Ev charging station map - find a place to charge. *PlugShare*. URL <https://www.plugshare.com/>.
- [84] Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bhartiya, Patrick Traynor, and Kevin RB Butler. Mo (bile) money, mo (bile) problems: Analysis of branchless banking applications. *ACM Transactions on Privacy and Security (TOPS)*, 20(3):1–31, 2017.
- [85] X de Carné de Carnavalet and Mohammad Mannan. Killed by proxy: Analyzing client-end tls interception software. 2016.
- [86] Suzan Ali, Mounir Elgharabawy, Quentin Duchaussoy, Mohammad Mannan, and Amr Youssef. Betrayed by the guardian: Security and privacy risks of parental control solutions. pages 69–83, 2020.
- [87] Wireshark. *Wireshark · Go Deep*. URL <https://www.wireshark.org/>.
- [88] Burp suite - application security testing software, 2021. URL <https://portswigger.net/burp>.
- [89] Adrian Gabriel Morosan and Florin Pop. Ocopp security-neural network for detecting malicious traffic. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pages 190–195, 2017.

- [90] Uli Harder, Matt W Johnson, Jeremy T Bradley, and William J Knottenbelt. Observing internet worm and virus attacks with a small network telescope. *Electronic Notes in Theoretical Computer Science*, 151(3):47–59, 2006.
- [91] The ucsd network telescope. *CAIDA*, Jan 2018. URL https://www.caida.org/projects/network_telescope/.
- [92] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. {ZMap}: Fast internet-wide scanning and its security applications. pages 605–620, 2013.
- [93] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J Alex Halderman. A search engine backed by internet-wide scanning. pages 542–553, 2015.
- [94] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. *26th USENIX security symposium (USENIX Security 17)*, pages 1093–1110, 2017.
- [95] Joseph Khoury, Morteza Safaei Pour, and Elias Bou-Harb. A near real-time scheme for collecting and analyzing iot malware artifacts at scale. *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–11, 2022.
- [96] Veronica Rammouz, Joseph Khoury, Đorđe Klisura, Morteza Safaei Pour, Mostafa Safaei Pour, Claude Fachkha, and Elias Bou-Harb. Helium-based iot devices: Threat analysis and internet-scale exploitations. In *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 206–211. IEEE, 2023.
- [97] Morteza Safaei Pour, Christelle Nader, Kurt Friday, and Elias Bou-Harb. A comprehensive survey of recent internet measurement techniques for cyber security. *Computers & Security*, page 103123, 2023.

- [98] Kevin Harnett, Brendan Harris, Daniel Chin, Graham Watson, et al. Doe/dhs/dot volpe technical meeting on electric vehicle and charging station cybersecurity report. Technical report, John A. Volpe National Transportation Systems Center (US), 2018.
- [99] Zakir Durumeric, Michael Bailey, and J Alex Halderman. An {Internet-Wide} view of {Internet-Wide} scanning. *23rd USENIX Security Symposium (USENIX Security 14)*, pages 65–78, 2014.
- [100] Yin Minn Pa Pa, Shogo Suzuki, Katsunari Yoshioka, Tsutomu Matsumoto, Takahiro Kasama, and Christian Rossow. Iotpot: Analysing the rise of iot compromises. *Emu*, 9(1), 2015.
- [101] Wei Zhou, Yan Jia, Yao Yao, Lipeng Zhu, Le Guan, Yuhang Mao, Peng Liu, and Yuqing Zhang. Discovering and understanding the security hazards in the interactions between {IoT} devices, mobile apps, and clouds on smart home platforms. In *28th USENIX security symposium (USENIX security 19)*, pages 1133–1150, 2019.
- [102] Published by S. O’Dea and Jun 29. Mobile os market share 2021, Jun 2021. URL <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>.
- [103] apktool - a tool for reverse engineering 3rd party, closed, binary android apps., 2021. URL <https://ibotpeaches.github.io/Apktool/>.
- [104] Dex2jar: Kali linux tools, Oct 2021. URL <https://www.kali.org/tools/dex2jar/>.
- [105] Jd-gui: Kali linux tools, Oct 2021. URL <https://www.kali.org/tools/jd-gui/>.

- [106] MobSF. Mobile security framework (mobsf), 2021. URL <https://github.com/MobSF/Mobile-Security-Framework-MobSF>.
- [107] Pkumza. pkumza/literadar: Lite version of libradar, 2021. URL <https://github.com/pkumza/LiteRadar>.
- [108] Shengqian Yang, Dacong Yan, Haowei Wu, Yan Wang, and Atanas Rountev. Static control-flow analysis of user-driven callbacks in android applications. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 89–99. IEEE, 2015.
- [109] Tanzirul Azim and Iulian Neamtiu. Targeted and depth-first exploration for systematic testing of android apps. In *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages & applications*, pages 641–660, 2013.
- [110] Gps joystick guide – the app ninjas. URL <http://gpsjoystick.theappninjas.com/>.
- [111] SHIVAM says. Virtualxposed apk 0.20.3 download latest in 2021 [official], Dec 2021. URL <https://virtualxposed.com/>.
- [112] Ac-Pm. Ac-pm/inspeckage: Android package inspector - dynamic analysis with api hooks, start unexported activities and more. (xposed module). URL <https://github.com/ac-pm/Inspeckage>.
- [113] Tony Nasr, Sadegh Torabi, Elias Bou-Harb, Claude Fachkha, and Chadi Assi. Power jacking your station: In-depth security analysis of electric vehicle charging station management systems. *Computers & Security*, page 102511, 2021.
- [114] Jens Schmutzler, Claus Amtrup Andersen, and Christian Wietfeld. Evaluation of

- ocpp and iec 61850 for smart charging electric vehicles. *World Electric Vehicle Journal*, 6(4):863–874, 2013.
- [115] Jens Schmutzler, Christian Wietfeld, and Claus Amtrup Andersen. Distributed energy resource management for electric vehicles using iec 61850 and iso/iec 15118. In *2012 IEEE Vehicle Power and Propulsion Conference*, pages 1457–1462. IEEE, 2012.
- [116] Communication apis for sms, voice, video & authentication. URL <https://www.twilio.com/>.
- [117] Yu-Wei Chung, Behnam Khaki, Tianyi Li, Chicheng Chu, and Rajit Gadh. Ensemble machine learning-based algorithm for electric vehicle user behavior prediction. *Applied Energy*, 254:113732, 2019.
- [118] Ahmad Almaghrebi, Subhaditya Shom, Fares Al Juheshi, Kevin James, and Mahmoud Alahmad. Analysis of user charging behavior at public charging stations. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6. IEEE, 2019.
- [119] Tai-Yu Ma and Sébastien Faye. Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks. *Energy*, page 123217, 2022.
- [120] Zachary J Lee, Tongxin Li, and Steven H Low. Acn-data: Analysis and applications of an open ev charging dataset. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pages 139–149, 2019.
- [121] Automation for apps, 2021. URL <https://appium.io/>.
- [122] Yosra Fraiji, Lamia Ben Azzouz, Wassim Trojet, and Leila Azouz Saidane. Cyber security issues of internet of electric vehicles. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

- [123] Richard M Pratt and Thomas E Carroll. Vehicle charging infrastructure security. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5. IEEE, 2019.
- [124] California iso - demand trend, 2021. URL <https://www.caiso.com/TodaysOutlook/Pages/default.aspx>.
- [125] Australian energy - demand trend, Oct 2021. URL <https://aemo.com.au/en>.
- [126] Guido Cavraro, Andrey Bernstein, Vassilis Kekatos, and Yingchen Zhang. Real-time identifiability of power distribution network topologies with limited monitoring. *IEEE Control Systems Letters*, 4(2):325–330, 2019.
- [127] Seyed Iman Taheri, MBC Salles, and N Kagan. A new modified tlbo algorithm for placement of avrs in distribution system. In *2019 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)*, pages 1–6. IEEE, 2019.
- [128] Guido Cavraro and Vassilis Kekatos. Inverter probing for power distribution network topology processing. *IEEE Transactions on Control of Network Systems*, 6(3):980–992, 2019.
- [129] Keith Moffat, Mohini Bariya, and Alexandra Von Meier. Unsupervised impedance and topology estimation of distribution networks—limitations and tools. *IEEE Transactions on Smart Grid*, 11(1):846–856, 2019.
- [130] Anandini Gandluru, Shiva Poudel, and Anamika Dubey. Joint estimation of operational topology and outages for unbalanced power distribution systems. *IEEE Transactions on Power Systems*, 35(1):605–617, 2019.
- [131] Deepjyoti Deka, Michael Chertkov, and Scott Backhaus. Topology estimation using

- graphical models in multi-phase power distribution grids. *IEEE Transactions on Power Systems*, 35(3):1663–1673, 2019.
- [132] Keith Moffat, Mohini Bariya, and Alexandra Von Meier. Real time effective impedance estimation for power system state estimation. In *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2020.
- [133] Prabha Kundur. Power system stability. *Power system stability and control*, pages 7–1, 2007.
- [134] J Duncan Glover, Mulukutla S Sarma, and Thomas Overbye. *Power system analysis & design, SI version*. Cengage Learning, 2012.
- [135] Powerworldthe visual approach to electric power systems, Jun 2021. URL <https://www.powerworld.com/>.
- [136] Bing Huang, Alvaro A Cardenas, and Ross Baldick. Not everything is dark and gloomy: Power grid protections against iot demand attacks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1115–1132, 2019.
- [137] Wei Zhou, Yan Jia, Yao Yao, Lipeng Zhu, Le Guan, Yuhang Mao, Peng Liu, and Yuqing Zhang. Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1133–1150, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL <https://www.usenix.org/conference/usenixsecurity19/presentation/zhou>.
- [138] Attack surface management and data solutions, December 2022. URL <https://censys.io/>.

- [139] OWASP Foundation. Owasp top ten. URL <https://owasp.org/www-project-top-ten/>.
- [140] 1N3. 1n3/intruderpayloads: A collection of burpsuite intruder payloads, burpbounty payloads, fuzz lists, malicious file uploads and web pentesting methodologies and checklists. URL <https://github.com/1N3/IntruderPayloads>.
- [141] Khaled Sareddine, Mohammad Ali Sayed, Danial Jafarigiv, Ribal Atallah, Mourad Debbabi, and Chadi Assi. A real-time cosimulation testbed for electric vehicle charging and smart grid security. *IEEE Security & Privacy*, 2023.
- [142] MITRE. Common Weakness Enumeration (CWE). <https://cwe.mitre.org>, 2023.
- [143] Maya Kaczorowski. Using cwe and cvss scores to get more context on a security advisory. *The GitHub Blog*, Feb 2021. URL <https://github.blog/2021-02-09-using-cwe-and-cvss-scores-to-get-more-context-on-a-security-advisory/>.
- [144] Splunk, Mar 2023. URL https://www.splunk.com/en_us/blog/learn/zero-day.html.
- [145] Joseph Antoun, Mohammad Ekramul Kabir, Ribal F Atallah, and Chadi Assi. A data driven performance analysis approach for enhancing the qos of public charging stations. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11116–11125, 2021.
- [146] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

- [147] Khaled Sarieddine, Mohammad Ali Sayed, Sadegh Torabi, Ribal Atallah, and Chadi Assi. Edge-based detection and localization of adversarial oscillatory load attacks orchestrated by compromised ev charging stations. *arXiv preprint arXiv:2302.12890*, 2023.
- [148] Amer S Al-Hinai. *Voltage collapse prediction for interconnected power systems*. West Virginia University, 2000.
- [149] Texas A&M University. Electric grid test case repository. *Texas A&M University College of Engineering*, 2023. URL <https://electricgrids.engr.tamu.edu/electric-grid-test-cases/wsc-9-bus-system/>.
- [150] OPAL-RT. Hypersim. URL <https://www.opal-rt.com/systems-hypersim/>.
- [151] IEA. Trends in charging infrastructure – global ev outlook 2022 – analysis. *IEA*. URL <https://www.iea.org/reports/global-ev-outlook-2022/trends-in-charging-infrastructure>.
- [152] Bing Huang, Alvaro A. Cardenas, and Ross Baldick. Not everything is dark and gloomy: Power grid protections against IoT demand attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1115–1132, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL <https://www.usenix.org/conference/usenixsecurity19/presentation/huang>.
- [153] Natural Resources Canada and U.S. Department of Energy. Final report on the implementation of the task force recommendations. URL <https://www.ieso.ca/en/Corporate-IESO/Media/Also-of-Interest/Blackout-2003>.

- [154] Walter Lucia and Amr Youssef. Covert channels in stochastic cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications*, 6(4):228–237, 2021.
- [155] Evan Perez. First on cnn: U.s. investigators find proof of cyberattack on ukraine power grid | cnn politics. *CNN*, Feb 2016. URL <https://www.cnn.com/2016/02/03/politics/cyberattack-ukraine-power-grid/index.html>.
- [156] Jack Maddox. Stuxnet: Malware more complex, targeted and dangerous than ever. *CNN*, Sep 2010. URL <http://www.cnn.com/2010/TECH/innovation/09/24/stuxnet.computer.malware/index.html>.
- [157] Australian energy market operator. *AEMO*, Jan 2022. URL <https://aemo.com.au/en>.
- [158] Motor vehicle census, australia. *Australian Bureau of Statistics*, 2022. URL <https://www.abs.gov.au/statistics/industry/tourism-and-transport/motor-vehicle-census-australia>.
- [159] Joseph Khoury and Mohamed Nassar. A hybrid game theory and reinforcement learning approach for cyber-physical systems security. In *NOMS 2020-2020 IEEE/I-FIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2020.
- [160] T Athay, Robin Podmore, and Sudhir Virmani. A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, (2):573–584, 1979.
- [161] Gaurav Bhatt and Shaik Affjulla. Analysis of large scale pv penetration impact on ieee 39-bus power system. In *2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 1–6. IEEE, 2017.

- [162] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.
- [163] Yi Zhang, Xiaohan Shi, Hengxu Zhang, Yongji Cao, and Vladimir Terzija. Review on deep learning applications in frequency analysis and control of modern power system. *International Journal of Electrical Power & Energy Systems*, 136:107744, 2022.
- [164] Keras Team. Keras documentation: Adam. *Keras*, 2022. URL <https://keras.io/api/optimizers/adam/>.
- [165] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [166] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [167] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.
- [168] Rafael G Mantovani, André LD Rossi, Joaquin Vanschoren, Bernd Bischl, and André CPLF De Carvalho. Effectiveness of random search in svm hyper-parameter tuning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. Ieee, 2015.
- [169] Information Trust Institute. Illinois center for a smarter electric grid (icseg). *Illinois*

Center for a Smarter Electric Grid ICSEG WSCC 9Bus System Comments, 2022.

URL <https://icseg.iti.illinois.edu/wsc-9-bus-system/>.

[170] Dennis F Galletta, Raymond Henry, Scott McCoy, and Peter Polak. Web site delays: How tolerant are users? *Journal of the Association for Information Systems*, 5(1):1, 2004.

[171] Nick Babich. 22 Basic UX Laws That Every Designer Should Know, Aug 2019.

[172] Jakob Nielsen. Website Response Times. <https://www.nngroup.com/articles/website-response-times/>, June 2010.

[173] Paul Doncaster. Chapter 2 - The UX Five-Second Rules. In Paul Doncaster, editor, *The UX Five-Second Rules*, pages 19–76. Morgan Kaufmann, Boston, 2014. ISBN 978-0-12-800534-7. doi: <https://doi.org/10.1016/B978-0-12-800534-7.00002-0>. URL <https://www.sciencedirect.com/science/article/pii/B9780128005347000020>.

[174] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S Woo. Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic pca. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2123–2133, 2019.