

Identification and Mitigation of Denial of Service Attacks on Electric Vehicle Charging Stations



By

Abdul Basit

00000327013

MS IT - 2K20

Supervisor

Dr Asad Waqar Malik

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters
of Science in Computer Science (MS CS)

In

School of Electrical Engineering & Computer Science (SEECS) ,


National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(February 2023)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Identification and mitigation of Denial of Service attacks on electric vehicle charging stations" written by Abdul Basit, (Registration No 00000327013), of SEecs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____  _____

Name of Advisor: _____ **Dr. Asad Waqar Malik** _____

Date: _____ **30-Jan-2023** _____

HoD/Associate Dean: _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "Identification and mitigation of Denial of Service attacks on electric vehicle charging stations" submitted by Abdul Basit have been found satisfactory for the requirement of the degree

Advisor : Dr. Asad Waqar Malik

Signature: Asad

Date: 30-Jan-2023

Committee Member 1: Prof. Hasan Ali Khattak

Signature: Hasan Ali Khattak

28-Jan-2023

Committee Member 2: Dr Farzana Jabeen

Signature: Farzana

Date: 28-Jan-2023

Signature: _____

Date: _____

Dedication

This thesis is dedicated to all the deserving children who do not have access to quality education especially young girls.

Certificate of Originality

I hereby declare that this submission titled "Identification and mitigation of Denial of Service attacks on electric vehicle charging stations" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: Abdul Basit

Student Signature: ABasit

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. There is nothing which can payback for His bounties throughout my research period to complete it successfully.

Abdul Basit

Contents

1	Introduction and Motivation	1
1.1	Security Threats on EVCS	3
1.1.1	Types of Attackers	4
1.1.2	DoS Attack	5
1.2	Motivation	5
1.3	Contribution	6
1.4	Thesis Organization	6
2	Literature Review	8
3	Attacks Design in EVCSs	11
4	Proposed Methodology	14
4.1	Dataset	16
4.1.1	Benign	16
4.1.2	DoS Hulk	17
4.1.3	DDoS	17
4.1.4	DoS GoldenEye	18
4.1.5	DoS slowloris	18
4.1.6	DoS Slowhttpstest	19
4.1.7	Heartbleed	19

5	Implementation and Results	21
5.1	System Specification	21
5.2	Evaluation Metrics	21
5.2.1	Accuracy	22
5.2.2	Precision	22
5.2.3	Recall	23
5.2.4	F1 - Score	23
5.3	Machine Learning	24
5.3.1	Decision Tree	24
5.3.2	Random Forest	24
5.3.3	Extra Tree	25
5.3.4	Extreme Gradient Boost	27
5.3.5	Stakings	27
5.3.6	KNN	28
5.4	Summary of the Results	29
6	Conclusion Future Work	32
6.1	Conclusion	32
6.2	Future Works	32

List of Figures

1.1	Attack Scenario in EVCS	4
3.1	Electric Vehicle Charging Stations Components	12
5.1	Accuracy of the Machine Learning Models	30

List of Tables

5.1	Classification report of Decision Tree	25
5.2	Classification report of Random Forest	26
5.3	Classification report of Extra Tree	26
5.4	Classification report of XGBoost	27
5.5	Classification report of Stackings	28
5.6	Classification report of KNN	29
5.7	Machine Learning Results	31

Abstract

The integration of communication technology into the power grid infrastructure has revolutionized the way power is managed and controlled. The new layer of connectivity has enabled bidirectional communication, smart resource management, remote control, and automation, but it has also opened the door to new security risks. With the rise of electric vehicles (EVs), it has become crucial to ensure that charging stations are reliable and secure. To counter the threat of denial of service attacks on EV charging stations, a machine learning-based intrusion detection system (IDS) has been proposed. The IDS uses multiple machine learning algorithm to identify and classify these types of intrusions. The results indicate that all the models used in the IDS have high detection accuracy, with the Random Forest method performing particularly well in terms of accuracy, precision, recall, and f1-score. Future research in this area will focus on the development of reinforcement-based IDS systems to enhance the security of electric vehicle charging station.

Introduction and Motivation

An electric vehicle charging station (EVCS) is a specialized system designed to provide electric vehicles (EVs) with the means to recharge their batteries. The station is composed of three key components, as described in [1]: computational components, communication and networking, and sensing. The sensing component of an EVCS consists of various sensors that monitor the status of the electrical components within the charging station. These sensors play a crucial part in ensuring the safety and reliability of the charging station by detecting issues such as voltage fluctuations, power outages, and potential hazards. The sensors can either be wired or wireless and are designed to detect a wide range of issues. By continuously monitoring the charging station, these sensors provide valuable information to the computational component, allowing for real-time adjustments and proactive maintenance to be performed, thus ensuring the continued efficient and safe operations of the EVCS.

The communication and networking components of an EVCS play a crucial role in connecting the charging station to various systems, including the internal sensors, supervisory control and data acquisition (SCADA) system, and the local grid. These components are responsible for facilitating communication between the EVs and charging stations, allowing for energy efficiency and availability to be monitored and managed. As described in [2], these components may utilize a range of wireless technologies, such as Wi-Fi, cellular, and Bluetooth, to achieve this communication.

The computational components of an EVCS are tasked with performing a variety of logical, arithmetic, and control functions. These components are responsible for scheduling charging times for EVs, maximizing the quantity of EVs that can be integrated in to

the grid, and authenticating EV owners before charging can commence. As noted in [3], the computational components may also be used to facilitate wireless communications, such as near field communication (NFC), Bluetooth, and radio frequency identification (RFID) which can introduce vulnerabilities into the EVCS.

An intrusion detection system (IDS) is a security tool designed to detect and classify attacks on computer systems or networks. According to [4], IDS can be classified into three main categories depends upon their implementation: host-based IDS (HID), network-based IDS (NID), and hybrid IDS. IDS use a variety of techniques to detect attacks, including stateful protocol analysis (SPA), anomaly-based detection (AD), and signature-based detection (SD). Signature-based detection relies on pre-defined patterns, or signatures, to identify known attacks. The system compares incoming attacks to the stored signatures, and if a match is found, the attack is detected. While this approach is effective for detecting known attacks, it has its limitations as it cannot detect new or unknown attacks. The system administrator must manually update the signatures for new attacks, which can be a time-consuming process and may not always be effective.

Anomaly-based detection-based IDS works by analyzing the behavior of a system or network and identifying deviations from normal behavior. Unlike signature-based detection (SD)-based IDS, AD-based IDS can detect unknown or previously unseen attacks, making it a more flexible approach. Many AD-based IDS systems use network-based IDS (NID) techniques to analyze network traffic for anomalies and machine learning techniques to enhance their ability to detect attacks, as reported in [5]. Stateful protocol analysis (SPA) is another technique used by IDS that involves analyzing the state of a network connection to detect attacks. This approach can be particularly effective in detecting attacks that exploit vulnerabilities in protocols or applications. IDS systems play a crucial role in securing computer systems and networks against attacks. By detecting and classifying attacks promptly, IDS help organizations take the necessary preventive measures to protect their systems.

Anomaly-based detection is a technique used in intrusion detection systems (IDS) that helps in detecting any anomalies or deviations from the normal behavior in a computer system or network. AD-based IDS work by learning and analyzing the behavior of a system or network and detecting any differences or deviations from this normal behavior. This makes it more flexible than signature-based detection (SD) methods, as it can

detect new and unknown network attacks. However, AD-based IDS also have their own limitations. One of the main disadvantages is that it can result in a high false alarm rate (FAR), meaning that it may trigger false alarms for attacks that are not actually happening. This can cause frustration for system administrators, as they may need to spend time investigating false alarms instead of focusing on more pressing security issues. Additionally, AD systems may go offline temporarily in order to update their understanding of normal network behavior after detecting a new type of attack, which can be disruptive to normal operations. Despite these limitations, AD is still considered an effective technique for intrusion detection, especially when combined with other methods like machine learning. The use of AD can help organizations take proactive measures to protect their computer systems and networks from potential attacks.

Stateful protocol analysis (SPA) is a method utilized in intrusion detection systems for the identification of potential threats or anomalies by comparing the actual behavior of a computer system or network to a set of predefined security specifications. SPA is considered specification-based detection, as it requires the security specifications of critical objects to be extracted and defined before comparison. This method of intrusion detection is different from anomaly-based detection (AD), which focuses on identifying deviations from normal network behavior, rather than comparing behavior to a predefined specification. One advantage of SPA is that it can be effective in detecting attacks that exploit vulnerabilities in protocols or applications, as it examines the protocol states in detail. However, SPA can also be resource-intensive, as it requires significant effort to trace and examine the protocol states. This can be a hindrance for system administrators, who may not have the necessary resources or time to implement this method effectively. Additionally, SPA may not be compatible with all operating systems and applications, and it may fail to inspect benign protocol behaviors, resulting in a high false alarm rate. This means that SPA may trigger false positives, or alarms for attacks that are not actually happening, which can distract system administrators from more pressing security issues.

1.1 Security Threats on EVCS

In a scenario where an intruder wants to gain access to confidential information within an EVCS, they may try to sneak into the system from an external location. The goal of the

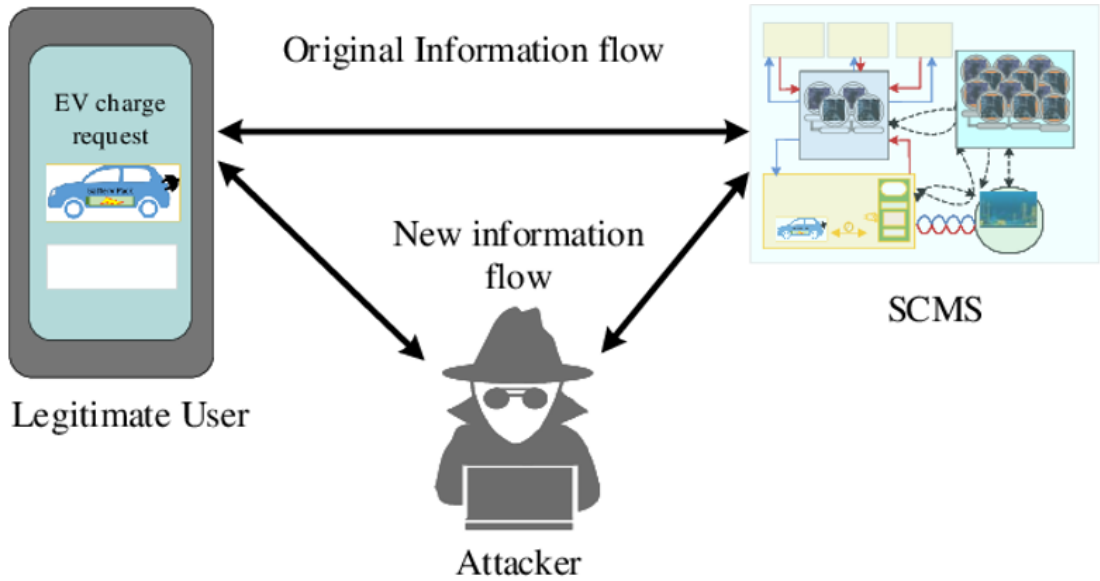


Figure 1.1: Attack Scenario in EVCS

attacker is to modify the data and cause disruption within the EVCS. This can be done by changing the original information about the scheduling of electric vehicles, resulting in congestion at the charging stations. The attacker's actions are depicted in Fig. 1.1, which shows the steps involved in the attack and how the intruder may modify and make intrusions into the EVCS. It is important to note that the attacker's intentions are malicious and their actions can cause significant harm to the system, leading to potential data loss or financial losses for the organization managing the EVCS.

1.1.1 Types of Attackers

The attackers in the EVCS can be divided into four categories, insider vs outsider, local vs extended, active vs passive, and rational vs malicious, are given below:

Insider vs Outsider

Insider vs outsider refers to whether the attacker is a member of the organization or not. An insider attacker is an employee, contractor, or third-party with access to the system, while an outsider attacker is someone who does not have authorization to access the system.

Rational vs Malicious

Rational vs malicious refers to the motivations of the attacker. A rational attacker is motivated by financial gain, political or ideological reasons, while a malicious attacker

is motivated by a desire to cause harm or destruction.

Active vs Passive

Active vs passive refers to the level of involvement of the attacker. An active attacker actively engages in harmful actions, while a passive attacker may just monitor the system, but does not take any actions that cause harm.

Local vs Extended

Local vs extended refers to the geographical location of the attacker. A local attacker is within the proximity of the system, while an extended attacker is located remotely and may access the system through the internet.

1.1.2 DoS Attack

Denial of Service (DoS) attacks are a malicious type of cyber attack aimed at making a computer, network, or service unavailable to their intended users. The attacker achieves this goal by overwhelming the other system with an excessive amount of traffic or requests, thereby causing the system to become overloaded and unable to function properly. This can result in significant disruption to the availability of critical services, causing financial losses and inconvenience for businesses and organizations. DoS attacks can be launched from a single device or from multiple devices such as a network of infected computers known as a botnet, making them difficult to defend against. DoS attacks can be challenging to trace and can be launched from anywhere, making it harder for organizations to protect themselves from these types of attacks. To defend against DoS attacks, organizations can implement an Intrusion Detection System (IDS) that can identify and classify potential attacks, and have systems in place to mitigate the effects of an attack if one does occur. This can include implementing rate-limiting mechanisms, traffic filtering, and load balancing, among others. It is important for organizations to be proactive in protecting themselves against DoS attacks, as the consequences can be severe if they are not prepared.

1.2 Motivation

The lack of research in the field of cybersecurity for electric vehicle charging stations (EVCS) creates a critical gap in the knowledge and understanding of how to effectively

secure these systems. This lack of understanding can leave EVCS operators without the necessary tools and techniques to effectively deal with cyberattacks and protect their systems. This vulnerability can result in serious consequences such as financial losses, harm to the organization's reputation, and even physical harm to users. It is imperative for EVCS operators to recognize the potential risks and take proactive measures to secure their systems and guarantee the safety of their users. To address the technical gap, further research is required to provide EVCS operators with the resources and knowledge necessary to protect against cyber threats and ensure the security of EVCS.

1.3 Contribution

This research aims to address the lack of research and understanding in the field of cyber security for electric vehicle charging stations by proposing a machine learning-based intrusion detection system to detect and classify denial of service attacks in the network. The proposed IDS is trained using the CICID 2017 dataset, which includes both examples of DoS attacks and benign traffic in an EVCS scenario. The IDS utilizes five different machine learning algorithms: Decision Tree, Random Forest, XGBoost, Extra Tree, and KNN. The results of the evaluation show that the proposed models are highly accurate in detecting DoS attacks, with an accuracy of at least 99%. Among the models, the Random Forest-based IDS performed the best in terms of precision, recall, and F1-score, demonstrating its effectiveness in detecting and classifying DoS attacks in an EVCS network.

1.4 Thesis Organization

This research study is structured in five chapters, each providing an in-depth discussion of a specific aspect of the proposed solution for detecting and classifying denial of service attacks in electric vehicle charging stations. Chapter 1 provides an overview of the electric vehicle charging station and intrusion detection system, highlighting the possible types of attacks that can occur in the scheduling process of EVCS. Chapter 2 reviews the existing research on the topic and provides a comprehensive analysis of the relevant work in the field of intrusion detection for EVCS. Chapter 3 focuses on the design of the DoS attacks in EVCS and highlights the possible intrusion scenarios that can occur in

the system. Chapter 4 outlines the methodology of the machine learning-based intrusion detection system, including the explanation of the classification of attacks and the intrusion detection model. The chapter provides a detailed description of the algorithms used, including KNN, Extra Tree, Decision Tree, Random Forest, and XGBoost, and their applications in the proposed IDS. At last, the conclusion is discussed in Chapter 6.

CHAPTER 2

Literature Review

Electric vehicle charging stations are vulnerable to cyberattacks due to their reliance on communication with incoming electric vehicles for scheduling, charging, authentication, and authorization, as well as with the grid for efficient energy use [6]. In the future, EVCS may also need to handle the bidirectional flow of energy between the charging station and the EV, which will further increase the complexity and need for secure cyberphysical infrastructure. There are various ways in which attackers could potentially target the EVCS network, such as through the wireless link between the vehicle and the charging station, or between the vehicle and the grid or between vehicles. These attacks could disrupt the normal operation of the EVCS and potentially compromise the privacy of the system. It is important for EVCS operators to be aware of these risks and to take steps to protect their systems from potential attacks.

There has been a significant amount of research on the development of intrusion detection systems (IDS) for electric vehicle (EV) charging station in recent years. In a research [5], the authors present a new intrusion detection system based on deep learning to address DoS attacks in electric vehicle charging stations. The suggested techniques utilize two commonly effective deep learning algorithms, specifically deep neural network (DNN) and long short term memory (LSTM) to classify DoS attacks into two categories (binary classification): attack or benign, and into five categories (multiclass classification): four types of DoS attacks and one benign class, using the CICIDS 2018 dataset for the EVCS environment. The research [7] introduces a probabilistic cross-layer intrusion detection system based on machine learning techniques. The IDS's detection engine uses a cross-layer approach and a variety of ML algorithms, including Random Forest

(RF) and k-Nearest Neighbour (kNN). Both of these supervised learning techniques are well-known for their performance: k-NN is resistant to noisy training data, such as those obtained from a real-world urban setting, while RF is known for its high accuracy and low risk of overfitting.

The researchers in [8] investigate Distributed Denial of Service (D DoS) and the effects of False Data Injection (FDI) attacks on the functioning of electric vehicle charging stations (EVCS). The study simulates FDI and synchronization flood D DoS attack on a 5G enabled remote Supervisory Control and Data Acquisition (SCADA) systems which control the solar photovoltaics (PV) controller, Battery Energy Storage (BES) controller, EV controller in an EVCS. In [9], the authors propose a collaborative and resilient IDS framework dependent upon federated learning for the electric vehicle charging framework. The system allows each charging stations to contribute to IDS model training by sharing model features rather than networks traffic data. In their framework, the EVCS act as federated workers and the central service provider serves as the federated master. To defend against membership interruption attack on model features, the researchers utilize unbalanced learning with differential privacy (uLDP). To the best of their knowledge, this effort represents the initial uLDP implementation in a federated learning environment. The framework also includes an clever security obscuration mechanism based on reinforcement learning for automation of the process of allocating privacy budgets and eliminate the need for human intervention in privacy provisioning. Moreover, the study [10] proposed a solution that can effectively detect unexpected events and potential fraudulent activity during charging sessions at charging stations using machine learning algorithms. However, these algorithms can struggle to perform well in large networks and often produce a high number of false positives and negatives, particularly due to shifts in data distribution across time. To address this problem, the researches suggest a Collaborative Anomaly Detection System for Charging Stations as a means of optimization. The authors [11] in examines current threats to communication networks, establishes a framework for information security protection, and suggests a comprehensive information security protection structure. The firewalls element uses the SHA-1 cryptographic hash technique and the ELGamal public key encryption scheme to authenticate digital signatures. The firewall also includes interactive modules for an intrusion detection systems (IDS) and host information security protection module. This work provides a foundation for building the security of the communication channel

in the charging stations. The study of the paper [12] focuses on the cybersecurity risks facing EVCS and presents mitigation and detection measures to tackle coordinate cyber attack. The core contributions of this paper is a cyber security mechanism that includes STRIDE-based threat modeling for the identification of potential vulnerabilities in an EVCS, a weighed attacks defence tree that describe many situations of cyberattack, a hidden Markov model to forecast the greatest probable route of a multi-stage attacks, and a partially observable Markov decision process method to steer the attacker away from the intended attack route.

CHAPTER 3

Attacks Design in EVCSs

Hackers often follow a set of steps when attempting to compromise an electric vehicle charging station (EVCS) or any other IT system. These steps, known as the cyberattack lifecycle, include reconnaissance, scanning, exploitation, and installation of backdoors or other malicious software.

During the reconnaissance phase, hackers use various techniques to gather information about the target system. These may include social engineering, where the hacker uses communication and persuasive skills to gain the trust of a legitimate user and obtain important information such as pins or passwords, or traffic analysis, where The attacker monitors network components to examine the network-connected devices.

In the scanning phase, the hacker identifies weaknesses in the system by monitoring services that runs on every open port, open ports, and the IP address. This may involve using tools to scan for vulnerabilities or manually probing the system for weaknesses.

During the exploitation phase, the attacker tries to take advantage of weaknesses in the charging station components in order to gain control over them. There are many types of attacks that can be launched at this stage, including jamming attacks, ransomware attacks, malware attacks, man-in-the-middle attacks, denial of service attacks, and replay attacks. Some of the most common types of DoS attacks include time synchronization attacks, time delay attacks, smurf attacks, puppet attacks, teardrop attacks, buffer overflow attacks, and SYN attacks.

Finally, in the last step, the hacker installs malicious software such as viruses, trojan horses, or backdoors on the target system. Backdoor attacks can be particularly damaging as they allow the hacker to gain undetectable and stealthy access to the system,

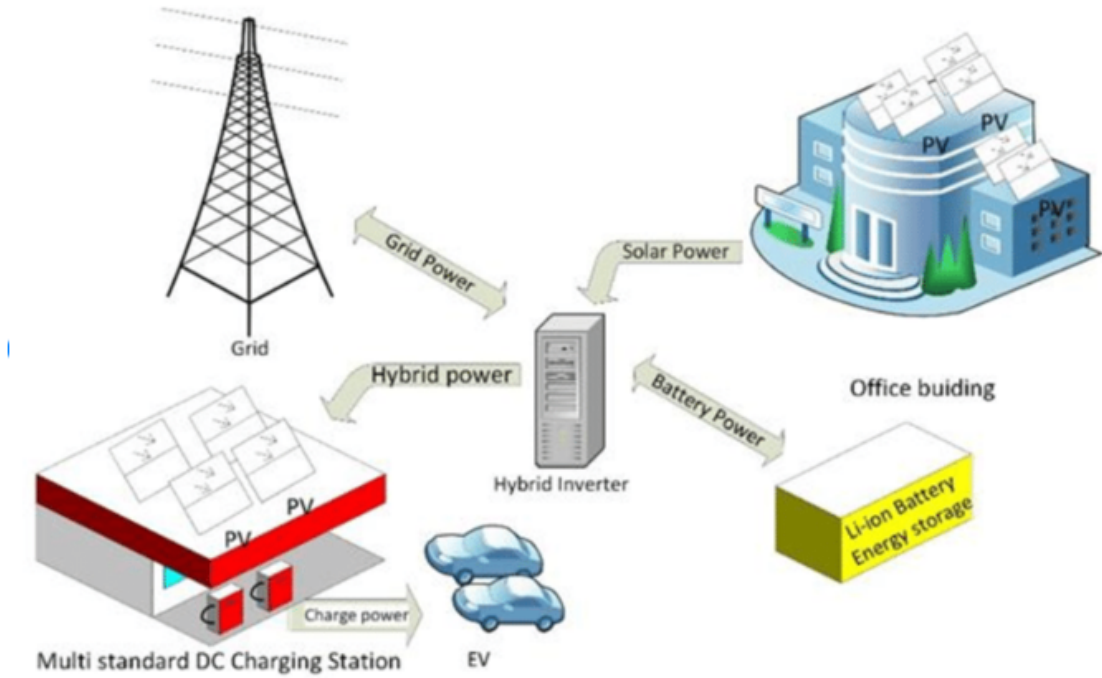


Figure 3.1: Electric Vehicle Charging Stations Components

facilitating multiple attacks on the EVCS server.

In the electric vehicle charging station architecture as shown in Fig. 3.1, there are four key components which are Grid, Solar Power, Battery Power, and Hybrid Power. These components are the building blocks of the charging station and work together to provide electric power to the vehicles. The Grid component is responsible for providing electric power from the traditional power grid. The Solar Power component utilizes solar energy to generate electric power for the charging station. The Battery Power component stores excess energy generated from either the Solar Power component or the Grid component for later use. The Hybrid Power component combines the Grid, Solar Power, and Battery Power components to provide a seamless source of power to the vehicles. These components play a crucial role in ensuring the efficient and reliable operations of the electric vehicle charging station.

DoS attacks are a type of cyberattack which aim to do a system or network unavailable to its users. DoS attacks can be launched against EVCS and can have serious consequences, including economic and reputational losses, customer dissatisfaction, and potentially even physical harm. A distributed DoS (DDoS) attack is one in which the identical Ev charging system is the target of several DoS attacks. DDoS attacks can be particularly

severe as they can involve a large number of devices or servers, making this hard to trace the origin of the attack. DDoS attack can be conducted in a variety of methods, including by using the valid third-party servers as component of an amplifying or reflecting assault, by straight syn flood offensive from a single machine, by a network of several machines, or by a machine with a fake IP address of a genuine user. It is important for EVCS operators to be aware of these risks and to take steps to protect their systems from DoS and DDoS attack.

This research study focuses on the detection of Denial of Services and Distributed Denial of Services attacks on electric vehicle charging stations. The authors acknowledge the importance of timely detection of these attacks in order to minimize the damage they can cause to the system and its users. To this end, the authors have made several assumptions about the data and the nature of the attacks. Here assume that the attackers have targeted the EVCS servers to initiate the DoS attack and that these attacks have altered the properties of network snippets. The authors also assume that they have access to network snippets both before, during and after the attacks, and that these packets can be used to extract relevant attributes that can be used in machine learning algorithms. Further it is assumed that the CICIDS 2017 DoS dataset is ideal for their study as it includes real-world examples of modern-day DoS attacks and can be used to train the intrusion detection system based on machine learning that they propose. The authors believe that these assumptions, when combined with machine learning algorithms, will allow them to accurately detect and classify DoS and DDoS attacks on electric vehicle charging stations.

Proposed Methodology

In order to use machine learning algorithms for supervised learning tasks, it is necessary to go through a series of steps. Feature engineering is an important part in machine learning which includes fetching the most important features or variables to use for training the algorithm. The aim of feature engineering is to identify and fetch features that are highly relevant and informative for the tasks at hands, while reducing the dimensionality of the data. This is important because using all of the available features may not be feasible due to limitations on computation and storage resources. There are several approaches that can be used for feature engineering, each with its own advantages and disadvantages. Principal component analysis (PCA) is one such approach that is commonly used to decrease the dimensions of the data and extract the most important feature. In PCA, the features are transformed into news variables, called principal components, that capture the most significant variability in the data. Another key step in the machine learning process is model selection, which involves choosing the best algorithm to fit the data. This step is important because the performance of the algorithm depends on the choice of model, and different models may perform better or worse on different data sets. Common machine learning models used for intrusion detection include decision trees, random forests, support vector machines (SVM), and deep neural network. The option of model will depends upon the characteristics of the data and the problem being solved, and the model must be trained on the data in order to make predictions.

Preprocessing and normalization of the data is another important part in the preparation the data for use in machine learning algorithms. This may involve converting data into

a uniform format, such as converting categorical data, integers, and floats into a single format. Min-max scaling and one-hot encoding are common techniques used for this purpose. Additionally, preprocessing may involve dealing with missing or corrupted data, and may also involve identifying and removing outliers. Normalization is a process of transforming data so that it has an average of zero and a standard deviation of one. This can help the machine learning algorithms to converge more quickly and to avoid overfitting to the training data. The results of normalization should be stored for future use so that the same transformation can be applied to the test data for accurate results. Preprocessing and normalization plays an important part in the overall performance of the machine learning algorithm, and it is important to carefully consider the techniques used to preprocess and normalize the data.

In the proposed machine learning algorithms, 50% of the data is used for training, the 30% is used for testing, and 20% is used for validation (to evaluate the model's performance with hypothetical data), and . It's indeed crucial to make assured that the data used for validation, validation, and testing are functionally incompatible, meaning that none of the data points overlap between these sets. This helps to ensure that the effectiveness of the model on the test data is an accurate representation of its ability to generalize to new, unseen data. Once the data has been properly preprocessed and normalized, the next step is the splitting of the dataset into three different sets: training, validation, and testing. The goal of this division is to assess how well machine learning models execute across various data sets and to ensure that the models generalize well to unseen data. In the proposed study, 50% of the data is used for training, 20% is used for validation, and the remaining 30% is used for testing. It is important to make sure that there is no overlap between these sets, meaning that none of the data points are used in more than one set. This helps to ensure that the performance of the model on the test data is a reliable indicator of its ability to generalize to new, unseen data. The validation data is utilized to fine-tune the hyperparameters of the algorithm and to avoid overfitting, It just happens whenever a methodology is too tuned to the training set and behaves badly on brand-new, untainted data.

The three primary objectives of the machine learning algorithms proposed in this research are as follows:

1. Classification: To determine whether a particular data vector represents an attack

or benign data. This classification will be based on the extracted features from the network packets.

2. Attack classification: To further categorize distinct attack classifications based on the entering channel matrix, such as the six different types of DoS attacks and one benign class.
3. Comparative analysis: To perform a comprehensive comparison between the applied machine learning algorithms, such as Decision Tree, Random Forest, XG-Boost, Extra Tree and KNN, to determine which one is most effective in classifying DoS attacks in the EV CS network. The result of the comparative analysis will help to determine the best approach for detecting and classifying DoS attacks in EV CS.

In essence, the suggested machine learning algorithms aim to classify and identify different types of DoS attacks in EVCS, to help improve the overall security of these systems.

4.1 Dataset

In this research, the authors have used the CICID 2017 dataset [13], which is a commonly used dataset for evaluating the effectiveness of intrusion detection systems (IDS). This dataset involves examples of recent denial of service (DoS) attacks, and has been selected for use in this study due to its relevance and up-to-date information on DoS attacks. Other popular datasets for IDS evaluation include the KDDCUP 99 dataset [14], the NSL KDD dataset [15], the UNSW NB15 dataset [16], the WSN-DS dataset [17], and the Kyoto dataset [18]. These datasets have been used extensively in previous research and provide a benchmark for the performance of intrusion detection systems.

The extent of datasets corresponding to each form of DoS attack in the CICIDS 2017 dataset are given below.

4.1.1 Benign

Benign data plays a crucial role in machine learning and data analysis because it serves as a benchmark for the models being developed. By training machine learning models on benign data, the models can learn to recognize what normal behavior looks like and

develop an understanding of what is considered "expected" behavior. This is particularly important in the context of intrusion detection and cyber security, where machine learning models are often used to identify and classify malicious data.

In addition to training the models, benign data is also utilized to assess the effectiveness of the models. Researchers often compare the model's predictions on benign data to the actual results to see how well the model is able to handle normal data and make accurate predictions. This information helps researchers to fine-tune the model and improve its performance.

4.1.2 DoS Hulk

A Hulk DoS attack is a type of denial of service (DoS) attack that uses the Hulk tool to generate a large number of HTTP requests and send them to a targeted website or server. The goal of the attack is to saturate the server's resources and prevent it from responding to legitimate requests, thereby disrupting the normal operation of the website or service.

Hulk works by generating a high volume of traffic that can mimic different types of legitimate traffic, making it difficult for the targeted site to distinguish it from normal traffic. The tool allows the attacker to customize the type and volume of traffic, as well as the rate at which it is sent.

DoS attacks like Hulk can have serious consequences for the targeted website or service, and they are illegal in many countries. It is important to protect against such attacks by implementing appropriate security measures, such as rate limiting and network intrusion detection systems.

4.1.3 DDoS

A distributed denial-of-service attack (DDoS) is a form of cyberattack which seeks to stop a website or other online service from operating normally by flooding them more information from several origins.

A botnet, or network of hacked computers, is what the perpetrator utilizes in a DDoS assault, to flood the targeted website or server with traffic. This traffic can come from a huge number of different sources, making it hard for the targeted site to differentiate

between legitimate and malicious traffic. The goal of the attack is to saturate the server's resources and prevent it from responding to legitimate requests, thereby rendering the website or service unavailable to its users.

DDoS attacks can have serious consequences for the targeted website or service, and they are illegal in many countries. It is important to protect against DDoS attacks by implementing appropriate security measures, such as DDoS protection services, network intrusion detection systems, rate limiting.

4.1.4 DoS GoldenEye

GoldenEye is a tool that is sometimes used to perform a denial of service (DoS) attack. A DoS attack is a type of cyberattack which seeks to destabilize the normal operation of a website or other online service through clogging it up with congestion, rendering it inaccessible to legitimate users.

GoldenEye works by generating a high volume of HTTP GET and POST requests and sending them to a targeted website or server. These requests can be customized to mimic different types of legitimate traffic, making it difficult for the targeted site to distinguish them from normal traffic. The goal of the attack is to saturate the server's resources and prevent it from responding to legitimate requests.

DoS attacks like GoldenEye can have serious consequences for the targeted website or service, and they are illegal in many countries. It is important to protect against such attacks by implementing appropriate security measures, such as rate limiting and network intrusion detection systems.

4.1.5 DoS slowloris

Slowloris is a tool that is sometimes used to perform a denial of service (DoS) attack. A DoS attack is a type of cyberattack which seeks to destabilize the normal operation of a website or other online service through clogging it up with congestion, rendering it inaccessible to legitimate users.

Slowloris works by sending a high volume of HTTP requests to a targeted website or server, but deliberately sending them at a slow rate. This can cause the server to become overwhelmed and unable to process legitimate requests, effectively shutting down the

website or service.

Slowloris attacks are particularly effective against servers that use the Apache HTTP Server software, as it is vulnerable to this type of attack. However, other types of servers may also be vulnerable to Slowloris attacks, depending on how they are configured.

DoS attacks like Slowloris can have serious consequences for the targeted website or service, and they are illegal in many countries. It is important to protect against such attacks by implementing appropriate security measures, such as rate limiting and network intrusion detection systems.

4.1.6 DoS Slowhttptest

Slowhttptest is a tool that is sometimes used to perform a denial of service (DoS) attack. A DoS attack is a type of cyberattack which seeks to destabilize the normal operation of a website or other online service through clogging it up with congestion, rendering it inaccessible to legitimate users.

Slowhttptest works by sending a high volume of HTTP requests to a targeted website or server, but deliberately sending them at a slow rate. This can cause the server to become overwhelmed and unable to process legitimate requests, effectively shutting down the website or service. Slowhttptest can also be used to test the resilience of a server to slow HTTP attacks.

Slowhttptest attacks are particularly effective against servers that use the Apache HTTP Server software, as it is vulnerable to this type of attack. However, other types of servers may also be vulnerable to Slowhttptest attacks, depending on how they are configured.

DoS attacks like Slowhttptest can have serious consequences for the targeted website or service, and they are illegal in many countries. It is important to protect against such attacks by implementing appropriate security measures, such as rate limiting and network intrusion detection systems.

4.1.7 Heartbleed

Heartbleed is a security vulnerability that was discovered in the open-source cryptographic software library OpenSSL in 2014. The vulnerability is caused by a flaw in the OpenSSL code that allows attackers to access sensitive information, such as passwords

and private keys, from the memory of affected systems.

Heartbleed affects versions 1.0.1 to 1.0.1f of OpenSSL, and it permits attackers to extract data from the memory of affected systems without leaving any trace of the attack. This makes it particularly difficult to detect and mitigate, as there is no way to determine if an attack has occurred or what data may have been compromised.

The vulnerability was quickly patched after it was discovered, but it is estimated that it may have affected as many as 500,000 servers worldwide. It is important to ensure that systems are updated with the latest patches to protect against known vulnerabilities like Heartbleed.

Implementation and Results

5.1 System Specification

In this research project, the authors have chosen to use Python 3.10.7 as the programming language and Jupyter Lab (version 3.6.0) as the development environment. Jupyter Lab is a user-friendly tool that provides an interactive and collaborative platform for conducting data analysis and scientific computing. It enables the sharing and creation of documents that contain a mixture of text, visualizations, equations, and live code, making it a best choice for this research project. The simulations and coding were performed on a computer with a high-performance Intel® Core™ i7-9670 processor, running at a speed of 3.20 GHz. The computer was equipped with 8.00 GB of RAM, ensuring that there would be sufficient memory for running the simulations and coding tasks. Additionally, a 64-bit edition of Windows 10 being installed on the device, which provides a stable and reliable platform for the simulations and coding tasks.

5.2 Evaluation Metrics

To evaluate the effectiveness of a machine learning model, the metrics of accuracy, precision, recall, and F1 score are frequently employed. These metrics are widely used in classification tasks to gauge the model's ability to correctly identify the class of a specific input data point.

5.2.1 Accuracy

Accuracy is a broadly utilized metric to evaluate the performance of a machine learning models, particularly in classification tasks. It measures the percentage of data points that the models has truly classified. To calculate accuracy, the amount of truly classified data points is divided by the overall amount of data points in the dataset. For example, if a model has correctly classified 90 out of 100 data points, its accuracy is 90%. The accuracy metric provides an overall measure of how good the models are capable to truly classify the data points. However, it may not always provide an accurate picture of the performance of the model, especially in imbalanced datasets where one class of data points is much more prevalent than the other. Mathematically, it is represented as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.2.1)$$

True positives (TP) and true negatives (TN) are two key metrics used to evaluate the performance of a machine learning model in a binary classification task. TP is the number of data points that the model correctly identifies as belonging to the positive class, while TN is the number of data points that the model correctly identifies as belonging to the negative class. On the other hand, false positives (FP) and false negatives (FN) are metrics used to measure the errors made by the model. FP is the number of data points that the model incorrectly identifies as belonging to the positive class, while FN is the number of data points that the model incorrectly identifies as belonging to the negative class. These metrics are often used together to evaluate the overall accuracy, precision, recall, and F1 score of a machine learning model.

5.2.2 Precision

Precision is a measure of the model's ability to correctly identify positive instances in the dataset. It measures the proportion of positive classifications that are actually correct. Precision is calculated by dividing the number of true positive data points by the sum of true positive and false positive data points. This metric gives an insight into the quality of positive predictions made by the model, specifically how many of the positive predictions are actually correct. A high precision value indicates that the model has a low false positive rate, which means that it is less likely to classify a benign data point as an attack. However, a high precision value does not guarantee that the model is

making enough positive predictions to identify all the attacks, which is why precision is often used in combination with other performance metrics such as recall and F1-score. Mathematically, it is represented as:

$$Precision = \frac{TP}{FP + TP} \quad (5.2.2)$$

5.2.3 Recall

Recall is a measure of the ability of a machine learning model to correctly identify positive instances. It is calculated as the ratio of true positive instances (data points that are both positive and correctly identified as positive) to the total number of actual positive instances in the dataset. Recall represents the fraction of positive instances that the model was able to correctly identify and is an important evaluation metric for machine learning models, especially in applications such as intrusion detection, where missing a positive instance could have significant consequences. High recall indicates that the model is able to correctly identify a large number of positive instances, while low recall suggests that the model is missing a significant number of positive instances. Mathematically, it is represented as:

$$Recall = \frac{TP}{TFP + FN} \quad (5.2.3)$$

5.2.4 F1 - Score

The F1 score is a single measure that considers both the precision and recall of a machine learning model. It provides a more comprehensive view of the model's performance by taking into account both the false negatives and false positives. The F1 score is calculated by taking the harmonic mean of precision and recall. The harmonic mean is used instead of the simple average because it gives more weight to low values. In the context of machine learning, it is important to have a balanced approach to precision and recall, as low precision can result in many false positives and low recall can result in many false negatives. The F1 score helps to balance these two measures and provide a single, comprehensive score for the model's performance. Mathematically, it is represented as:

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5.2.4)$$

5.3 Machine Learning

In this work, the authors used a variety of machine learning models for intrusion detection. These models include extreme gradient boost, decision tree, extra tree, random forest, K-nearest neighbors (KNN), and stacking.

5.3.1 Decision Tree

A decision tree is a type of machine learning algorithm that is used to classify data points based on their features. It works by constructing a tree-like model of decisions based on the features of the data. At each internal node of the tree, a decision is made based on a feature of the data, and the data is split into different branches based on the value of that feature. This process is repeated at each node until the data is fully classified, resulting in a tree-like structure with branches representing different decisions and leaf nodes representing the final classification of the data.

Decision trees can be used for both classification and regression tasks, and are often used in areas such as data mining, machine learning, and artificial intelligence. They are easy to understand and interpret, and can handle both continuous and categorical data. However, they can be prone to overfitting, meaning that they may perform poorly on new, unseen data. To mitigate this risk, it is often necessary to prune the tree or use other techniques to prevent overfitting. The classification of the model is given in the Table 5.1.

5.3.2 Random Forest

A random forest is an ensemble machine learning algorithm that is used for both classification and regression tasks. It works by constructing a large number of decision trees and then combining the predictions of these trees to make a final prediction. The decision trees in a random forest are constructed using a random subset of the features and a random subset of the training data. This means that each tree in the forest is slightly

Table 5.1: Classification report of Decision Tree

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	1.00	0.99	1.00	4546
DoS Hulk	0.99	0.98	0.98	393
DDoS	0.99	1.00	1.00	554
DoS Golden Eye	1.00	1.00	1.00	3807
DoS slow loris	0.86	0.86	0.86	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	0.99	0.98	0.99	436

different from the others, and the combination of these different trees leads to a more robust and accurate model.

Random forests are widely used in many areas of machine learning and data analysis due to their ability to handle a wide range of data types and their high accuracy. They are particularly useful for dealing with large datasets and for handling high-dimensional data. One of the main advantages of random forests is that they are less prone to overfitting than individual decision trees, as the combination of multiple trees helps to smooth out the predictions and reduce the risk of overfitting. However, they can be computationally expensive to train and may require a large amount of memory to store. The classification of the model is given in the Table 5.2.

5.3.3 Extra Tree

An extra tree is a type of decision tree that is used for both classification and regression tasks. It works by constructing a tree-like model of decisions based on the features of the data, similar to a standard decision tree. However, extra trees differ from standard decision trees in that they use random thresholds for each feature rather than the optimal thresholds that would be found using traditional decision tree algorithms. This makes extra trees more resistant to overfitting, as they do not rely on finding the optimal split points for each feature.

Extra trees are often used in conjunction with other machine learning algorithms, such

Table 5.2: Classification report of Random Forest

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	1.00	1.00	1.00	4546
DoS Hulk	0.99	0.98	0.98	393
DDoS	1.00	1.00	1.00	554
DoS Golden Eye	1.00	1.00	1.00	3807
DoS slow loris	0.83	0.71	0.77	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	1.00	0.98	0.99	436

as random forests, as they can provide additional information about the data and help to improve the overall accuracy of the model. They are particularly useful for handling high-dimensional data and for dealing with datasets that have a large number of features. One of the main advantages of extra trees is that they are fast to train and do not require a lot of computation, making them well suited for use in large-scale machine learning systems. The classification of the model is given in the Table 5.3.

Table 5.3: Classification report of Extra Tree

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	1.00	0.99	0.99	4546
DoS Hulk	0.97	0.98	0.98	393
DDoS	1.00	1.00	1.00	554
DoS Golden Eye	1.00	1.00	1.00	3807
DoS slow loris	1.00	0.71	0.83	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	1.00	0.99	0.99	436

5.3.4 Extreme Gradient Boost

XGBoost is an open-source machine learning library that is widely used for classification, regression, and ranking tasks. It stands for "eXtreme Gradient Boosting" and is based on the gradient boosting algorithm, which is a machine learning technique that involves training a series of decision trees and combining their predictions to make a final prediction.

XGBoost is known for its high performance and accuracy, and has been used to win many data science competitions. It is particularly well suited for dealing with large datasets and for handling high-dimensional data. It can handle missing values in the data and has built-in support for parallel processing, which makes it efficient to train even on large datasets. XGBoost has a number of hyperparameters that can be adjusted to fine-tune the model and improve its performance, and it includes a number of techniques for regularization and early stopping to prevent overfitting. The classification of the model is given in the Table 5.4.

Table 5.4: Classification report of XGBoost

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	0.99	0.99	0.99	4546
DoS Hulk	0.99	0.98	0.99	393
DDoS	1.00	1.00	1.00	554
DoS Golden Eye	0.99	1.00	1.00	3807
DoS slow loris	0.83	0.71	0.87	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	1.00	0.98	0.99	436

5.3.5 Stakings

Stacking is a machine learning ensemble technique that involves training multiple models and combining their predictions to make a final prediction. It works by using the predictions of a set of base models as input features for a higher-level model, which is trained to make a final prediction based on the base model predictions. This approach

can improve the performance of the overall model by leveraging the strengths of different base models and by reducing the variance of the predictions.

Stacking is often used to combine the predictions of different types of models, such as decision trees, random forests, and neural networks. It can be used for both classification and regression tasks, and is particularly useful for handling high-dimensional data and for dealing with large datasets. One of the main advantages of stacking is that it can improve the overall accuracy of the model by combining the predictions of multiple models, which can be more robust and less prone to overfitting than a single model. However, it can be computationally expensive to train and may require a large amount of memory to store the base models. The classification of the model is given in the Table 5.5.

Table 5.5: Classification report of Stackings

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	1.00	0.99	0.99	4546
DoS Hulk	1.00	0.97	0.98	393
DDoS	0.99	1.00	1.00	554
DoS Golden Eye	0.99	1.00	1.00	3807
DoS slow loris	1.00	0.71	0.83	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	1.00	0.98	0.99	436

5.3.6 KNN

KNN, or k-Nearest Neighbors, is a machine learning algorithm that is used for both classification and regression tasks. It works by identifying the k data points in the training set that are most similar to a given data point (where k is a user-specified parameter), and then using the class labels or values of these neighboring points to make a prediction for the given data point.

KNN is a simple and easy-to-understand algorithm that does not require any training, as it simply uses the data points in the training set to make predictions. It is often used

for classification tasks, where it can be effective in identifying patterns in the data and classifying new data points based on these patterns.

However, it can also be used for regression tasks, where it can be used to predict continuous values based on the values of the k nearest neighbors. One of the main advantages of KNN is that it is relatively fast to predict, as it only requires a small number of calculations to determine the nearest neighbors. However, it can be computationally expensive to train, as it requires storing the entire training set in memory. The classification of the model is given in the Table 5.6.

Table 5.6: Classification report of KNN

	PRECISION	RECALL	F1 - SCORE	SUPPORT
Benign	0.99	0.99	0.99	4546
DoS Hulk	0.97	0.96	0.96	393
DDoS	0.99	1.00	0.99	554
DoS Golden Eye	1.00	1.00	1.00	3807
DoS slow loris	0.44	0.57	0.50	7
DoS Slow http test	1.00	1.00	1.00	1589
Heart bleed	0.98	0.98	0.98	436

5.4 Summary of the Results

All machine learning models function effectively because all model gives more than 99% accuracy, random forest consistently yields the most favorable results compared to the other models. The accuracy of all models are given in the Fig 5.1.

The summary of all models are given in the Table 5.7.

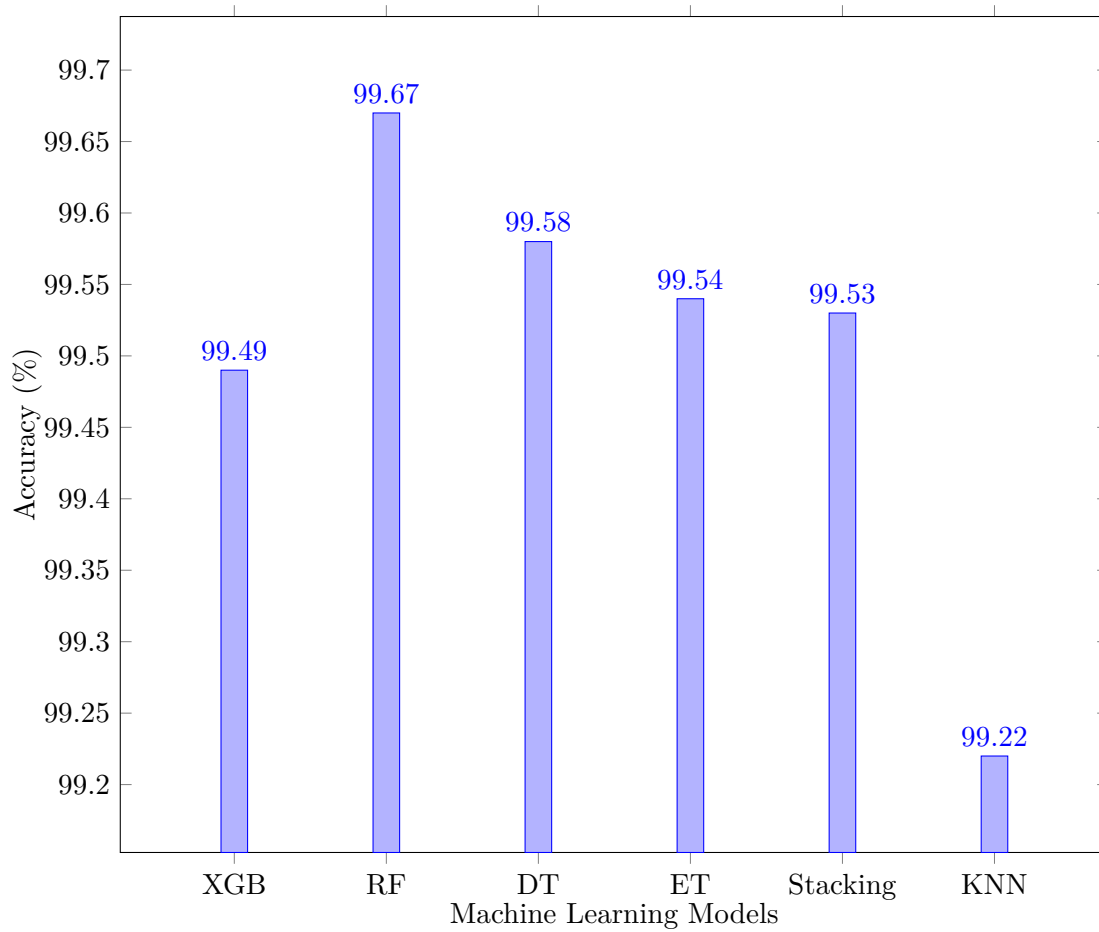


Figure 5.1: Accuracy of the Machine Learning Models

Table 5.7: Machine Learning Results

	ACCURACY	PRECISION	RECALL	F1 - SCORE
XG Boost	0.9949	0.9949	0.9949	0.9949
Random Forest	0.9967	0.9967	0.9967	0.9967
Decision Tree	0.9958	0.9958	0.9958	0.9958
Extra Tree	0.9954	0.9954	0.9954	0.9953
Stacking	0.9953	0.9953	0.9953	0.9952
SVM	0.8587	0.8644	0.8587	0.8440
KNN	0.9922	0.9923	0.9922	0.9922

Conclusion Future Work

6.1 Conclusion

The integration of communication layers into the physical infrastructure of power grids poses a significant risk of potential cybersecurity threats. These threats can undermine the confidentiality, integrity, and availability of grid resources and have severe impacts on the reliability and safety of the power grid. To tackle these challenges, the use of machine learning-based intrusion detection systems (IDS) has been shown to be an effective solution for detecting and classifying potential attacks on electric vehicle charging stations. The results from this research indicate that multiple machine learning algorithms can achieve high levels of accuracy in detecting potential attacks, with the Random Forest method performing particularly well. Further research into the development of reinforcement-based IDS systems may be beneficial, as these systems can adapt to new types of threats and attacks. Additionally, it is crucial to consider implementing additional measures to ensure the security of the power grid, such as secure communication protocols and proper security protocols and policies. It is crucial to consider the potential cybersecurity threats and implement effective solutions to protect against them to ensure the reliability and safety of the power grid.

6.2 Future Works

The future work in intrusion detection systems for electric vehicle charging stations will aim to tackle the limitations of existing machine learning-based IDS. These limitations

include issues such as traffic imbalance and high false alarm rates. Traffic imbalance is a problem where the distribution of normal and attack traffic in the training dataset is uneven, leading to poor performance and biased results. High false alarm rates refer to the IDS producing many false positive results, where normal traffic is mistakenly identified as malicious. To address these issues, future research in this field will concentrate on the development of reinforcement-based IDS. Reinforcement-based methods train the IDS by using feedback from its actions to learn and adapt over time. This approach has the potential to improve the performance of IDS for EVCS by reducing traffic imbalance and the false alarm rate. By incorporating reinforcement-based methods, the goal is to create more accurate and reliable IDS that can adapt to new types of threats and attacks.

Bibliography

- [1] Raju Gottumukkala et al. “Cyber-physical system security of vehicle charging stations”. In: *2019 IEEE Green Technologies Conference (GreenTech)*. IEEE. 2019, pp. 1–5.
- [2] Mohamed A Ahmed, Mohamed R El-Sharkawy, and Young-Chon Kim. “Remote monitoring of electric vehicle charging stations in smart campus parking lot”. In: *Journal of Modern Power Systems and Clean Energy* 8.1 (2019), pp. 124–132.
- [3] Islam Safak Bayram and Ioannis Papapanagiotou. “A survey on communication technologies and requirements for internet of electric vehicles”. In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014), pp. 1–18.
- [4] S Sobin Soniya and S Maria Celestin Vigila. “Intrusion detection system: Classification and techniques”. In: *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE. 2016, pp. 1–7.
- [5] 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)*. IEEE. 2020, pp. 408–413.
- [6] Naireeta Deb et al. “A Review of Extremely Fast Charging Stations for Electric Vehicles”. In: *Energies* 14.22 (2021), p. 7566.
- [7] Dimitrios Kosmanos et al. “A novel intrusion detection system against spoofing attacks in connected electric vehicles”. In: *Array* 5 (2020), p. 100013.
- [8] Manoj Basnet and Mohd Hasan Ali. “Exploring cybersecurity issues in 5G enabled electric vehicle charging station with deep learning”. In: *IET Generation, Transmission & Distribution* 15.24 (2021), pp. 3435–3449.
- [9] Shafkat Islam et al. “An Intelligent Privacy Preservation Scheme for EV Charging Infrastructure”. In: *IEEE Transactions on Industrial Informatics* (2022).

BIBLIOGRAPHY

- [10] Jesus Cumplido, Cristina Alcaraz, and Javier Lopez. “Collaborative anomaly detection system for charging stations”. In: *European Symposium on Research in Computer Security*. Springer. 2022, pp. 716–736.
- [11] Huan Li et al. “Information security protection design of electric vehicles charging station”. In: *Applied Mechanics and Materials*. Vol. 741. Trans Tech Publ. 2015, pp. 681–686.
- [12] Mansi Girdhar et al. “Machine Learning-Enabled Cyber Attack Prediction and Mitigation for EV Charging Stations”. In: *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2022, pp. 1–5.
- [13] Deris Stiawan et al. “CICIDS-2017 dataset feature analysis with information gain for anomaly detection”. In: *IEEE Access* 8 (2020), pp. 132911–132921.
- [14] Mahbod Tavallaee et al. “A detailed analysis of the KDD CUP 99 data set”. In: *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee. 2009, pp. 1–6.
- [15] S Revathi and A Malathi. “A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection”. In: *International Journal of Engineering Research & Technology (IJERT)* 2.12 (2013), pp. 1848–1853.
- [16] Nour Moustafa and Jill Slay. “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)”. In: *2015 military communications and information systems conference (MilCIS)*. IEEE. 2015, pp. 1–6.
- [17] Iman Almomani, Bassam Al-Kasasbeh, and Mousa Al-Akhras. “WSN-DS: A dataset for intrusion detection systems in wireless sensor networks”. In: *Journal of Sensors* 2016 (2016).
- [18] Jungsuk Song et al. “Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation”. In: *Proceedings of the first workshop on building analysis datasets and gathering experience returns for security*. 2011, pp. 29–36.