Universidade Aberta

&

Universidade de Trás-os-Montes e Alto Douro

Electric Vehicules Cyber-attack Detection

Master Dissertation in Computer Engineering and Web Technology

Abderrazak Mahi

Advisor: Professor António Cunha

Co-Advisor: Professor Pedro Mestre

Portugal, 2025

Universidade Aberta

&

Universidade de Trás-os-Montes e Alto Douro

Electric Vehicules Cyber-attack Detection

Master Dissertation in Computer Engineering and Web Technology

Abderrazak Mahi

Advisor: Professor António Cunha

Co-Advisor: Professor Pedro Mestre

Jury composition

Jury Presidency

Portugal, 2025

"The duty of the man who investigates the writings of scientists, if learning the truth is his goal, is to make himself an enemy of all that he reads." - *Ibn al-Haytham (Alhazen, 965–1040) – Father of Optics, Scientific Method*

Scientific Advisor

António Cunha

Associate Professor (Universidade de Trás-os-Montes e Alto Douro, Portugal)

Scientific Co-Advisor

Pedro Mestre
Assistant Professor (Stirling School, China)

Acknowledgments

This dissertation is the culmination of intense dedication, perseverance, and continuous learning throughout the course of my academic journey. It reflects not only my individual efforts but also the support and encouragement of those who accompanied me along the way.

First and foremost, I thank God for granting me the strength, patience, and resilience to overcome the various challenges I faced during this process. Without His guidance and blessings, none of this would have been possible.

I extend my heartfelt gratitude to my family for their unconditional support throughout this journey. Their encouragement, understanding, and love have been my foundation.

I would like to express my sincere gratitude to Tiago Silva for his constructive feedback, support and reviews. His insights greatly contributed to the optimization and refinement of my work, and his dedication was instrumental in enhancing its overall quality.

I would also like to thank Ahmed Walid Belhadj, a dear friend and data analyst, who kindly answered several of my technical questions and offered valuable insights that contributed to the development of this work.

My sincere thanks go to Professor António Manuel Cunha, my scientific advisor, for his dedicated guidance and constructive feedback. His honest advice, thought-provoking suggestions, and constant support played a fundamental role in shaping this dissertation.

I would also like to express my profound appreciation to Professor Pedro Mestre, my co-advisor, for all his guidance, contributions and suggestions throughout this project. His knowledge, valuable advice and constant support were truly appreciated and contributed immensely to the development of this dissertation.

In addition, I wish to express my appreciation to Rodrigo Abrantes, a PhD student and cybersecurity expert, whose expertise and helpful discussions on the security aspects of my work were extremely valuable.

I am especially thankful to the authors of the CICEVSE2024 dataset—E. D. Buedi, A. A. Ghorbani, S. Dadkhah, and R. Ferreira—for granting me access to the dataset, which formed the backbone of my experimental analysis. Without their contribution, this research would not have been possible.

Finally, to everyone who, in one way or another, contributed to this academic journey: thank you.

Abstract

This thesis explores ML and DL techniques for improved cybersecurity of EVSE as part of smart grid systems. As the adoption of electric vehicles increases, their related infrastructure is in jeopardy of cyber-attack, which can directly affect the users' safety, data privacy, and overall stability of the grid. The overall objective of this research was to prototype, implement, test and evaluate, and compare several classification-based intrusion detection systems that could detect harmful activity in an EVSE environment.

This work builds upon the published CICEVSE2024 dataset, which is a multilayer, multimodal dataset that provides telemetry data from a simulated EV charging environment. At this time, there was limited research on the use of telemetry measurement, and the potential value add to an intrusion detection space. The dataset contained network level, power level, and kernel level telemetry data which, when used in conjunction, can support extensive analysis for intrusion detection purposes. Given the breadth and diversity of the data attributes, it is well suited for creating realistic, enforceable detection accuracy models.

There are two types of classification tasks in this study: Binary classification task of normal vs. attack packets, and multi-class classification task of identifying the type of attack.

One goal of this work was to achieve and maintain high detection accuracy, which we achieved by developing a pipeline for our analysis which included: Data cleaning and data normalization; Feature selection and correlation analysis and Implementation of scenario-based and class-based balancing strategies to eliminate data imbalance. The use of both mature ML algorithms (Random Forest, Gradient Boosting, Support Vector Machine, K-Nearest Neighbors, Logistic Regression) and DL algorithms (LSTM, GRU based architectures).

The performance of each model was evaluated by measures of overall accuracy, precision, recall, F1-Score, and confusion metrics and validated in a train-test setting. The study's findings suggested classical ML models have rapid search speed and provide interpretable outputs, however DL models are more suited to detecting intrusions in the time series and context by moving beyond independent features in the telemetry data demonstrating high accuracy, especially where the sequential model could capture the temporal dependencies of LSTM and GRU models.

The thesis also examined implications of using ML/DL hybrids for intrusions detection and discussed several main principles to consider in the context of specificity in modeling, the value of a realistic dataset, limitations in detecting zero-day attacks, and stealth attacks. Some of the challenges to developing robust, and importantly adaptable and scalable intrusion detection systems in real world CPS contexts of EVSE were acknowledged.

Overall, we can say that the research analysis confirmed the value and significance of continued study of multimodal datasets, and provided suggestions for future research based on detection, certainly online detection, but also transfer learning strategies while accounting for the need to deploy models in edge contexts, is warranted. Overall, the study made methodological contributions and validated its experimental analysis providing new perspectives and possibilities for securing and monitoring the next evolution of smart EV charging infrastructure.

Key-words: Electric Vehicle Supply Equipment (EVSE), Intrusion Detection System (IDS), Multimodal Telemetry, Network Security, Power-Level Data, Kernel-Level Data, Temporal Dependency, Sequential Modeling, Zero-Day Attacks, Stealth Attacks, Edge Computing, Transfer Learning, Cyber-Physical Systems (CPS)

Contents

List of Figures

List of Tables

List of Abbreviations

EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
IDS	Intrusion Detection System
ML	Machine Learning
DL	Deep Learning
CPS	Cyber-Physical Systems
RF	Random Forest
SVM	Support Vector Machine
KNN	K-Nearest Neighbors
LR	Logistic Regression
GBT	Gradient Boosting Trees
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
GNN	Graph Neural Network
OCPP	Open Charge Point Protocol
ISO 15118	International Standard for Vehicle-to-Grid Communication Interface
SMOTE	Synthetic Minority Over-sampling Technique
AUC	Area Under the Curve
F1-Score	Harmonic Mean of Precision and Recall
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative

DoS Denial of Service

DDoS Distributed Denial of Service

MitM Man-in-the-Middle

CSV Comma-Separated Values

API Application Programming Interface

Chapter 1 – General Introduction

IoT has become one of the quickest developing technological sectors, impacting on a large scale of organizations, homes and urban infrastructure by 2030. The number of IoT-related devices is anticipated to exceed 125 billion globally. Thanks to the fast adoption of 5G and other related technology. Advancements in communication generation, affordability, there is also the invention of efficient sensors. IoT programs variety from business automation to healthcare analytic and smart cities solutions, which assist improve operations and aid usage. However, this rapid improvement has also increased the complexity of the IoT surroundings, which includes devices with specific capabilities. With the enormous network and interconnection, ensuring the security and reliability of IoT systems has become increasingly hard. As IoT continues to expand, addressing these vulnerabilities in the network is prime to guarding vital infrastructure and consumer information. [1][2]

The worldwide transition toward sustainable transportation has brought a large increase within the electric vehicles market. In 2023, over 14 million EVs had been sold worldwide, accounting for 18% of total vehicle income, a dramatic growth compared to 4% in 2020. This increase is fueled through improvements in battery generation, declining expenses, and authorities' incentives to sell green energy adoption. Leading markets, which include China, Europe, and America, have seen a speedy increase in EV sales, with China by itself contributing to greater than 50% of the global income. As the number of EVs on the road grows, the infrastructure for charging those motors is also expanding, with millions of charging stations deployed globally. Despite the tremendous environmental effect, this growth provides challenges, inclusive of the need for scalable infrastructure and cybersecurity measures to defend this crucial sector. Ensuring the resilience of EV ecosystems is important as they grow to be critical to attaining worldwide sustainability dreams. [3][4]

While electric powered vehicles offer environmental and monetary benefits, the charging infrastructure supporting them is increasingly threatened by cyberattacks. Many EVSE structures lack solid protection, making them susceptible to several cyber threats, consisting of ransomware attacks, DoS, and unauthorized access. For example, research indicates that 84% of EV chargers surveyed didn't put into effect proper encryption protocols, exposing people and operational statistics to interception. A 2023 study highlighted how hackers could take advantage of open ports in charging networks to disrupt services or maybe manipulate electricity pricing. Additionally, compromised charging stations could function as entry points to larger power grids, doubtlessly leading to big disruptions. These vulnerabilities are now a big risk, threatening the reliability of EV, however, they also jeopardize user safety and data privacy. Addressing those challenges calls for sturdy cybersecurity measures tailored to EVSE structures, focusing on real-time detection and mitigation. [5][6]

This thesis targets to cope with the urgent need for stronger cybersecurity in EV charging infrastructures by developing strong detection models. The proposed models utilize the CICEVSE2024 dataset, which gives a comprehensive collection of real-global attack eventualities precise to EV charging structures. The main contribution of this study is the design and implementation of several architectures based on trial and error and coming up with the most suitable architecture for this subject. This method ensures accurate identification of each acknowledged and novel attack styles even as maintaining scalability for actual-global deployment. Additionally, the thesis evaluates the overall performance of the model throughout a couple of attack sorts, benchmarking its effectiveness against existing answers. By providing a sensible and progressive solution, this study contributes to the developing area of cybersecurity for EV infrastructures and presents actionable insights for producers and policymakers to enhance the resilience of EVSE structures. [7]

The CICEVSE2024 dataset, developed by the Canadian Institute for Cybersecurity, is a complete and meticulously curated dataset designed to simulate real-world cyberattack eventualities focused on EVSE. It consists of various records on network traffic, communication protocols, and system logs from EV charging infrastructure under various regular and malicious conditions. The dataset captures a wide variety of kinds of attacks, inclusive of ransomware, DoS, phishing, and zero-day exploits, permitting researchers to look at both recognized and rising threats. Its designated labeling and representation of actual-international EVSE operations make it a treasured aid for training and testing superior intrusion detection models. By leveraging the CICEVSE2024 dataset, this thesis guarantees that the proposed AI-pushed detection model is validated on practical and enterprise-relevant attack situations, bridging the gap among theoretical strategies and practical programs in EVSE cybersecurity. Figure 1 illustrates a comparative sales trend between Classic Vehicles over the period from 2020 to 2030, with the sales measured in millions of units.

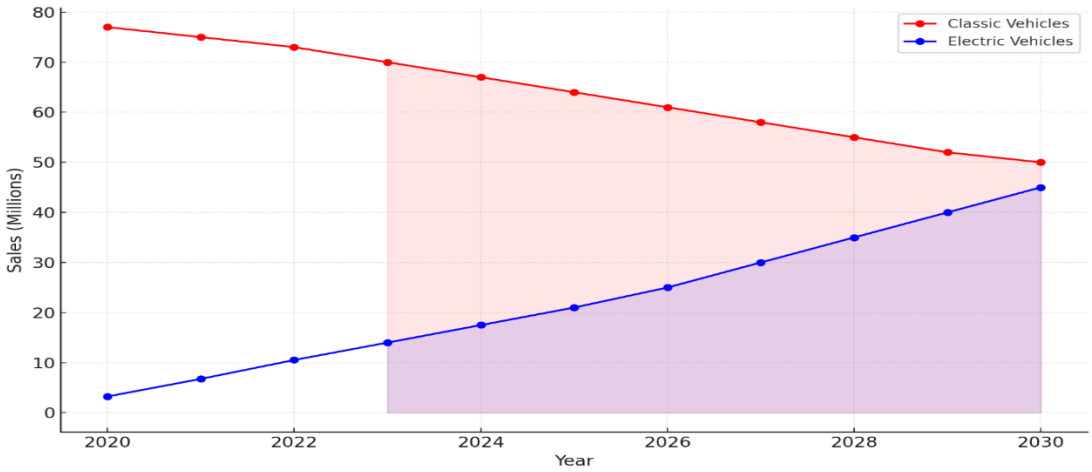

Figure 1: Global Vehicle Sales: Electric vs. Classic Vehicles (2020-2030)

1.1 Motivation

The growth of EV adoption and charging infrastructure has opened new attack vectors for cybercriminals. Recent reports highlight real-global attacks that divulge the vulnerabilities of EVSE. For example, a documented ransomware attack in Europe disrupted an entire EV charging network, rendering stations inoperable for hours and causing financial losses to companies. Similarly, researchers have validated that poorly secured EV chargers may be exploited to govern charging schedules, energy intake, or at the same time as entry factors into related power grids. In an excellent experiment, Zhang et al. (2024) demonstrated how open communication ports in charging systems allow unauthorized attacks to enter the network too, and to cause grid instability. These examples emphasize that cyber threats focused on EV charging infrastructure aren't simply theoretical, However, they present impending dangers to operational reliability, user safety, and power grid safety.

Current cybersecurity techniques employed in IoT environments, inclusive of EVSE structures, basically depend upon traditional strategies like signature-based totally IDS, anomaly detection models, and rule-primarily based processes. While these methods, such as Random Forest and SVM, have shown effectiveness in detecting acknowledged attack styles, their applicability to EV charging structures stays underexplored. In the broader IoT context, deep learning tactics, inclusive of CNNs and hybrid models, have received traction due to their capability to investigate massive-scale, complex network traffic. For instance, Cheng et al. (2023) used AI-driven IDS to become aware of anomalies inside smart grid structures, achieving excessive accuracy. However, these solutions are frequently designed for usual IoT devices and are hardly ever tailored to the precise exchange protocols, electricity constraints, and real-time operational needs of EV chargers.

Despite advances in cybersecurity and IoT attack detection, several vital gaps remain while carried out to EVSE systems. First, there may be a great number of studies that especially cope with

cyberattacks on EV chargers. Most existing research focuses on general IoT devices or smart grids, overlooking the great vulnerabilities of EV charging infrastructure. Second, traditional detection strategies warfare to increase evolving threats, consisting of zero-day attacks, which make the most formerly unknown vulnerabilities. Moreover, modern-day techniques regularly fail to reap actual-time detection while balancing accuracy and computational performance, that is critical for resource-limited EVSE devices. These gaps highlight a pressing want for specialized and scalable detection answers tailor-made to EV charging networks to ensure resilience against both regarded and emerging threats.

ML and DL strategies give a promising solution to deal with the prevailing gaps in EV charger security. AI models like CNNs, RNNs, and hybrid approaches excel in figuring out complex patterns and anomalies in network site traffic information. For instance, the mixing of AI allows for the detection of both acknowledged and novel attacks by analyzing subtle deviations in conversation protocols and system behavior. Deep learning, a rising AI approach, can in addition optimize protection with the aid of allowing decentralized training of detection models throughout disbursed EVSE networks while keeping consumers data private. By leveraging AI, it is possible to increase sturdy, actual-time, and scalable attack detection systems which are specially optimized for EV charging environments. This thesis goal is to bridge this studies gap through providing innovative AI-driven detection models tailor-made to protect EV charging infrastructure against cyber threats.

1.2 Objectives

The primary aim of this thesis is to design and enforce an AI-based attack detection models for EVC infrastructure to address current cybersecurity vulnerabilities. By using advanced machines and deep learning techniques, this study ambitions to bridge the gap between theoretical methods and possible solutions, ensuring the safety, reliability, and scalability of EV charging structures.

To reach this aim, the thesis sets out the following specific goals:

- Investigate and examine existing cyberattack detection strategies applied to IoT and smart grid structures, figuring out their strengths, barriers, and applicability to EVC infrastructure.
- Identify gaps in cutting-edge research, emphasizing the lack of specialized detection models for EV charging networks and the challenges of real-time detection and scalability.
- Develop a strong AI-based detection model after trial and error capable of identifying recognized and novel attack patterns precise to EV chargers with the usage of the CICEVSE2024 dataset.
- Evaluate the proposed detection model's overall performance by benchmarking it in opposition to modern solutions with the use of key metrics which include accuracy, precision recall and more.

- Provide actionable recommendations for producers, researchers, and policymakers to enhance the cybersecurity resilience of EVSE.

1.3 Contributions

This thesis provides a widespread contribution to addressing cybersecurity demanding situations in EVC structures through the development of a sturdy AI-driven attack detection model. The key contributions are as follows:

- **Comprehensive Literature Analysis:** Conducted an in-intensity systematic assessment of present IoT attack detection techniques, highlighting their strengths, weaknesses, and applicability to EVSE. Identified crucial study gaps, particularly the shortage of specialized research specializing in EV charger cybersecurity and the challenges of real-time detection, scalability, and handling zero-day attacks.
- **Development of a Novel Detection Model:** Designed and applied AI-primarily based models to analyze traffic styles and hit upon anomalies in EVC structures. The proposed model is particularly skilled and evaluated on the CICEVSE2024 dataset, which incorporates many attack scenarios concentrated on EV chargers.
- **Performance Benchmarking and Validation:** Conducted rigorous evaluation of the proposed model of the usage of key metrics including accuracy, precision, recall, and F1-rating to make sure of its effectiveness in detecting each recognized and emerging cyberattacks. Benchmarked the overall performance of the model in contrast to cutting-edge solutions to demonstrate improvements in detection accuracy and real-time applicability.
- **Practical Recommendations for EVSE Security:** Provided actionable insights and suggestions for manufacturers, researchers, and policymakers to strengthen the cybersecurity resilience of EVSE infrastructure. Highlighted the significance of integrating lightweight and scalable AI solutions into real-world EV charging.

Chapter 2 – Contextualization

EVSE represents a crucial component inside the sustainable transition of transportation systems. While EVSE allows seamless integration of electrical motors into the energy grid, its increasing connectivity to the IoT introduces new cybersecurity demanding situations. These challenges, inclusive of vulnerabilities in encryption, scalability problems, and difficulties in detecting zero-day attacks, spotlight the urgent need for targeted studies. This chapter delves into the technicalities of EVSE structure, its cybersecurity demanding situations, and potential AI-driven solutions to address these gaps, culminating in a comprehensive exploration of the CICEVSE2024 dataset as a transformative tool in EVSE studies.

2.1 EV Charging Infrastructure: Technical Overview

The EVC infrastructure paperwork is a crucial issue of the worldwide transition to sustainable transportation. Its architecture encompasses both hardware and software components, operating in unison to ensure green and steady energy delivery. This section affords an in-depth assessment of the technical factors of EV charging stations, that specialize in hardware additives, software program systems, and communication protocols along with the OCPP and ISO 15118.

A. Architecture of EV Charging Stations

EVCS are designed as integrated systems comprising multiple hardware and software components as shown in Figure 2.

1. Hardware Components:

- **Electric Vehicle Supply Equipment (EVSE):** The EVSE provides the physical interface between the EV and the electrical grid. It includes charging connectors, cables, and protective enclosures.
- **Power Conversion Unit (PCU):** Converts AC grid electricity to DC for DCFC or modulates the AC supply for AC charging. Advanced systems use bidirectional inverters to enable V2G functionalities.
- **Charge Controllers:** These devices manage the power flow and control the charging process. They interface with both the vehicle and the central management system to ensure appropriate power delivery.
- **Energy Meters and Sensors:** Energy meters measure the electricity consumed, while sensors monitor environmental conditions and equipment status to enhance safety and operational efficiency.

- **Cooling Systems:** High-power chargers (e.g., ultra-fast chargers) often require advanced cooling systems to prevent overheating and ensure consistent performance.

2. Software Aspects:

- **Central Management System (CMS):** A cloud-based system that oversees station operations, user authentication, billing, and real-time monitoring.
- **Firmware and Embedded Software:** Embedded systems in the EVSE and charge controllers manage real-time communication and execution of charging protocols.
- **User Interfaces:** These include mobile applications, touchscreen kiosks, or RFID card readers for user interaction, session management, and payment processing.

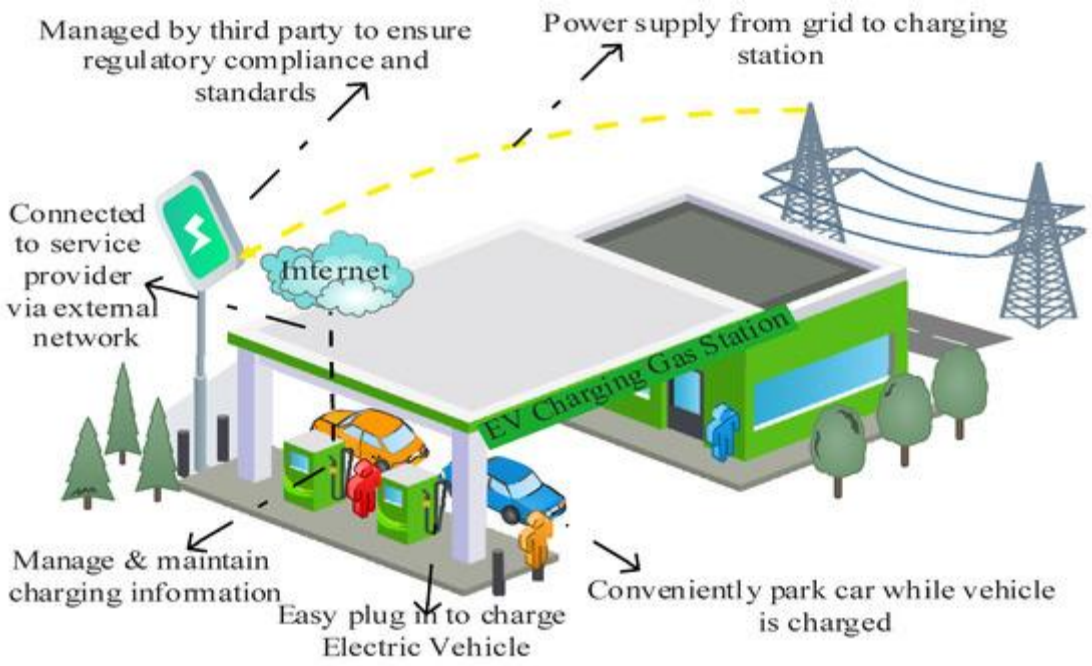

Figure 2: EV public Charging use case scenario.

B. Communication Protocols

Efficient communication between the various entities within the EV charging ecosystem is crucial for seamless operation. Protocols like OCPP and ISO 15118 play a pivotal role in achieving this.

- **2.1 Open Charge Point Protocol (OCPP):** OCPP is an open-source communication standard developed to facilitate interoperability between EV chargers and central management systems. The latest version, OCPP 2.0.1, provides several enhancements:
    - **Bidirectional Communication:** OCPP allows chargers and CMS to exchange real-time data on session status, fault notifications, and energy usage.

    - **Smart Charging Capabilities:** The protocol enables dynamic load management by adjusting charging power based on grid constraints and user preferences.
    - **Security Features:** OCPP 2.0.1 incorporates TLS for encrypted communication and secure firmware updates to mitigate cyber threats.
    - **Integration with V2G Systems:** OCPP supports bidirectional energy flow communication, enabling vehicles to discharge energy back to the grid.
- **2.2 ISO 15118:** ISO 15118 is a suite of international standards governing the communication between EVs and EVSEs. Its objectives include enhancing user convenience, enabling secure operations, and supporting future-proof functionalities:
    - **Plug-and-Charge (PnC):** ISO 15118 allows EVs to automatically authenticate and initiate charging sessions upon connection, eliminating the need for manual input.
    - **Security Mechanisms:** The protocol employs cryptographic techniques to ensure data integrity, confidentiality, and mutual authentication between the vehicle and the charging station.
    - **Energy Management:** It supports advanced energy management features such as scheduled charging, demand-response integration, and V2G communications.
    - **Scalability:** The standard is adaptable to evolving technologies, ensuring compatibility with next-generation EVs and charging stations.

The architecture of EV charging stations, integrating robust hardware and sophisticated software systems, is critical for reliable and efficient EV adoption. Communication protocols like OCPP and ISO 15118 enable seamless interoperability, secure operations, and smart energy management, aligning with the future demands of sustainable transportation.

2.2 Cybersecurity in IoT and EVSE: Challenges and Implications

EVSE serves as an important interface between EVs and the strength grid, facilitating the charging system. However, the mixing of EVSE into the wider IoT environment introduces unique cybersecurity demanding situations that can have extensive implications for the EV surroundings.

A. Technical Vulnerabilities in EVSE

- **Weak Encryption:** Many EVSE systems make use of old or insufficient encryption protocols, making them prone to facts interception and unauthorized get admission to. For instance, inadequate encryption can allow attackers to interexchange between the EV and the charging station, doubtlessly leading to record breaches or unauthorized manipulation.
- **Open Ports:** EVSE units frequently have open network ports to facilitate further management and firmware updates. However, these open ports can function as entry points for cyber attackers if they are not nicely secured, permitting unauthorized entry to the tool's internal systems.

- **Firmware Flaws:** Firmware vulnerabilities within EVSE may be exploited to execute malicious code, disrupt charging operations, or maybe damage the vehicle's battery. Unpatched firmware flaws may permit attackers to take advantage of managing the charging process, leading to capacity safety dangers.

B. Impact on the EV Ecosystem

The vulnerabilities in EVSE have distinct implications for the EV ecosystem, differentiating them from general IoT devices:

- **Grid Stability Risks**: Compromised EVSE can be manipulated to create sudden spikes in electricity demand, potentially destabilizing the power grid. Coordinated attacks on multiple charging stations could lead to grid overloads and widespread outages.
- **Vehicle Safety Concerns**: Exploiting EVSE vulnerabilities can directly affect vehicle operations. For example, altering charging parameters through a compromised station could damage the vehicle's battery or electrical systems, posing safety risks to users.
- **Data Privacy Issues**: EVSE units collect and transmit user data, including charging habits and payment information. Weak security measures can lead to unauthorized access to this sensitive information, resulting in privacy breaches.

Case Study: Ransomware Attack on EV Charging Infrastructure

An excellent example illustrating the technical demanding situations and implications of EVSE vulnerabilities is the potential for ransomware attacks concentrated on the SCADA structures that manipulate EV charging stations. In such an attack, malicious actors should encrypt the management systems, rendering the charging stations inoperative and disturbing a ransom to restore functionality. This scenario does not disrupt the supply of charging offerings, however, additionally poses full-size operational challenges for EV customers and service providers.

Addressing the cybersecurity vulnerabilities in EVSE is important for the safe and reliable operation of the EV atmosphere. Implementing strong encryption protocols, securing network interfaces, and often updating firmware are essential steps to mitigate those dangers. Additionally, developing comprehensive protection frameworks tailor-made to the unique components of EVSE can assist guard against capacity cyber threats and ensure the integrity of both the vehicles and the helping infrastructure.

2.3 Current State of Cyberattack Detection in IoT

IDS are critical for figuring out and mitigating cyber threats inside IoT environments, such as EVSE. Traditional detection mechanisms inclusive of signature-primarily based IDS, anomaly detection,

and rule-based structures have been widely applied. However, their utility to EVSE gives specific demanding situations, in handling zero-day attacks.

A. Traditional Detection Mechanisms

1. **Signature-Based IDS**: This method relies on a database of known threat signatures to detect malicious activities. It is effective against previously identified threats but struggles with new, unknown attacks. For instance, in smart energy grids, signature-based systems offer higher detection rates for known attacks but require extensive manual configuration and cannot learn from new attack patterns.

2. **Anomaly Detection**: Anomaly-based IDS establishes a baseline of normal network behavior and flags deviations as potential threats. This approach can identify unknown attacks but often results in higher false positive rates. In IoT applications, anomaly detection systems can detect novel attacks by identifying deviations from normal behavior patterns.

3. **Rule-Based Systems**: These systems use predefined rules to detect suspicious activities. While they can be effective for specific scenarios, they may not adapt well to the dynamic nature of IoT environments. In IoT networks, rule-based deep learning models have been proposed to detect and classify novel attacks, achieving high accuracy rates.

B. Weaknesses in Application to EVSE

Applying these traditional detection mechanisms to EVSE introduces specific challenges:

1. **Cyberattacks**: EVCS may interfere with operations. Network attacks, such as service or port scanning, deny target communication systems, making them inaccessible or impaired. Host -based attacks, such as cryptos, compromise internal systems, while non -conformity with electrical consumption manipulates energy use, potentially damaging harmful infrastructure. These attacks affect both EVC functionality and security, causing physical damage to blocking service, data violations and components.
2. **Resource Constraints:** EVSE devices often have limited computational resources, making it challenging to implement complex detection algorithms without impacting performance.
3. **Dynamic Network Behavior:** The operational environment of EVSE is dynamic, with varying charging patterns and user behaviors, complicating the establishment of a stable baseline for anomaly detection.
4. **Maintenance and Updates:** Rule-based systems require regular updates to remain effective, which can be logistically challenging for widespread EVSE deployments.

Case Study: Delayed Charging Attack (DCA) on Electric Shared Mobility Systems

A study on DCA in electric shared mobility systems highlights the limitations of traditional IDS in EVSE contexts. The DCA exploits vulnerabilities in the communication between shared electric vehicles and EVSE, causing charging delays that lead to operational disruptions. The study found that traditional anomaly detection algorithms were insufficient to detect DCA, emphasizing the need for more robust detection mechanisms tailored to EVSE environments.

While traditional intrusion detection mechanisms provide a foundation for securing IoT devices, their application to EVSE requires careful consideration of the unique challenges present in these environments. Developing adaptive, resource-efficient, and context-aware detection systems is essential to effectively safeguard EVSE against both known and emerging cyber threats.

2.4 Cyberattacks on Electric Vehicle Charging Stations

EVC are an important infrastructure, and their safety is important for reliable service. Attacks on EVC can be divided into three categories: network attacks, host -based attacks and non -conformity with electrical consumption.

- **Network attacks:** These interfere with the communication between EVC components, which have unauthorized access to rejections of service or unauthorized data.
- **Host -based attacks:** These are aimed at EVC's internal systems, such as cryptojacking and back door attacks.
- **Deviations in electrical consumption:** The use of abnormal electricity may indicate malicious activity, such as electrical load manipulation or service through electrical fluctuations.

A. Network Attacks

- **DOS:** An attack aimed at overwhelming a network resource, which is not available for legitimate users. "Rejection of service" occurs when the attackers flooded a system with extra traffic. [8]
- **TCP port scanning:** This includes sending requests to different gates to find weaknesses, which can later be utilized. [9]

B. Host-Based attacks

- **Cryptojacking:** This attack uses the victim's machine resources without authority for Cryptocurrency. [10][11]
- **BACK DEPORT:** Used to achieve remote access in malicious software systems, socializing (Shin et al., 2014).

C. Electrical consumption anomalies

- **Electrical load manipulation:** attackers change power consumption to disrupt the normal function of EVCs.
- **Power Daniel-Off-Service (DOS):** A form of DOS that affects the physical components of EVC by manipulating power levels, leading to power failure.

EVC security is important and understanding these attacks helps to reduce risk. Proper security measures are required to prevent disruption in both networks and physical composition as shown in Figure 3.

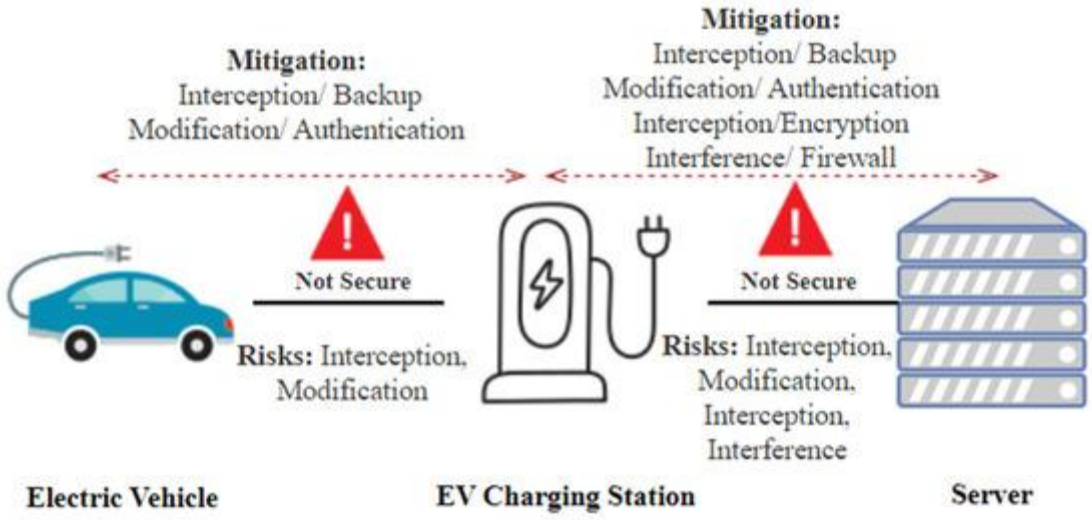

Figure 3: Cyberattacks on EV charging networks.

2.5 AI-Driven Solutions for EVSE Cybersecurity

AI techniques have become pivotal in enhancing cybersecurity measures for EVSE. Advanced models such as CNNs, RNNs, and hybrid architectures offer significant advantages in detecting and mitigating cyber threats which can overcome the lack of standardization issue as shown in Figure 4.

A. AI Techniques in Cybersecurity

- **CNNs:** Primarily used for spatial information evaluation, CNNs are a good choice at figuring out patterns within statistics, making them appropriate for picture and signal processing. In cybersecurity, CNNs were hired to stumble on anomalies in network site visitors by recognizing malicious styles. For instance, a study applied CNNs to identify cyber-attacks in electric vehicle charging infrastructure, demonstrating excessive accuracy in detection.
- **RNNs:** RNNs are designed for sequential statistics evaluation, capturing temporal dependencies within datasets. This functionality is beneficial for tracking time-series

statistics in EVSE systems to coming across irregular sports over the years. Research has shown that RNNs can efficiently model and predict sequential patterns, assisting in the identification of potential safety breaches.

- **Hybrid Models (example: CNN-LSTM):** Combining CNNs and LSTM networks leverages each spatial and temporal statistics evaluation strengths. Such hybrid models have been developed to expect key performance metrics in complicated systems, improving the accuracy of cyber-attack detection in EVSE with the aid of taking pictures of intricate styles in records.

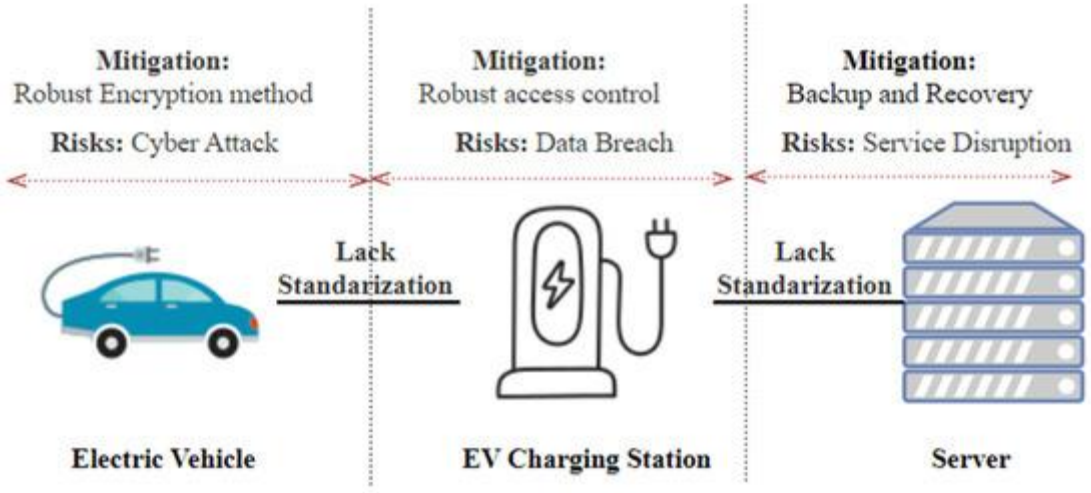

Figure 4: Lack of Standardization issue in EV charging network architecture.

B. Challenges in Real-Time AI-Based Detection for EVSE

Implementing AI-driven cybersecurity solutions in EVSE presents several demanding situations:

- **Computational Constraints:** EVSE devices often have constrained processing capabilities, making it hard to deploy useful resource-extensive AI models. Techniques like TinyML aim to address those barriers by using optimizing models for low-electricity devices, however challenges which include energy consumption and constrained memory persist.
- **Scalability:** As the range of EVSE units increases, ensuring constant and green security monitoring throughout all devices will become complicated. Scalable AI solutions need to be advanced to handle big-scale deployments without compromising performance. Deep learning tactics have been proposed to strengthen scalability by way of training models across decentralized systems, lowering the need for centralized data processing.
- **Real-Time Processing:** Detecting and responding to cyber threats in real-time is critical for EVSE safety. However, achieving low-latency processing with AI models is hard because of the computational needs and the desire to spark off decision-making. Edge-primarily based

detection techniques have been explored to facilitate real-time analysis with the aid of processing records towards the source, thereby lowering latency.

Addressing those challenges calls for ongoing studies and improvement to optimize AI models for the unique constraints of EVSE environments, ensuring sturdy and green cybersecurity measures.

2.6 The CICEVSE2024 Dataset: A Game-Changer

The CICEVSE2024 dataset represents a huge development within the field of EVSE cybersecurity research. Developed to address the shortage of comprehensive datasets in this domain, it gives a multi-dimensional series of statistics encompassing power intake metrics, network visitors' logs, and host activity data below each benign and malicious conditions.

A. Composition of the Dataset

The dataset is meticulously dependent to facilitate diverse cybersecurity analyses:

- **Power Consumption Data:** Captures the electrical utilization patterns of the EVSE throughout preferred operations and points out various attack eventualities.
- **Network Traffic Data:** Includes precise packet captures reflecting the exchange between the EVSE and external entities, presenting insights into network-based threats.
- **Host Activity Logs:** Document the internal procedures and system logs within the EVSE, essential for detecting anomalies at the host stage.

This comprehensive composition permits researchers to perform baseline behavioral profiling, class, and anomaly detection responsibilities correctly.

B. Real-World Applicability

The CICEVSE2024 dataset is designed to reflect actual-global scenarios, improving its applicability in cybersecurity studies:

- **Simulated Scenarios:** The dataset encompasses each idle and active charging states of the EVSE, subjected to many attack vectors which include network-based reconnaissance, DoS attacks, and host-primarily based intrusions including backdoor entries and cryptojacking.
- **Data Labeling Methodology:** Each information point inside the dataset is meticulously classified to indicate whether it represents normal operation or a specific attack. This labeling enables supervised learning programs to enhance the accuracy of ML models evolved for intrusion detection.

By imparting a dataset that carefully replicates the operational environment of EVSEs, CICEVSE2024 serves as an important aid for developing and trying out robust cybersecurity

measures tailored to the precise demanding situations of electric automobile charging infrastructure.

2.7 Gaps in Research and the Need for Innovation

The integration of EVSE into the broader IoT surroundings has introduced specific cybersecurity demanding situations that are not directly addressed by existing research. This segment synthesizes the recognized gaps and establishes the research area of interest that this thesis objectives to cope with.

A. Identified Research Gaps

- **Lack of EVSE-Specific Cybersecurity Models:** While widespread IoT cybersecurity frameworks exist, they regularly fail to account for the unique operational characteristics and chance vectors related to EVSE. The absence of tailored models leaves EVSE prone to attacks that exploit those unique aspects.
- **Scalability Issues in Detection Mechanisms:** Traditional IDS and anomaly detection methods may not scale effectively in the EVSE context, particularly as the range of connected devices increases. This problem hampers the ability to maintain robust safety throughout increasing EVSE networks.
- **Challenges in Real-Time Detection:** Implementing real-time detection mechanisms in EVSE is complex by computational constraints inherent to these devices. The excessive useful resource needs of superior detection algorithms, along with those based totally on deep learning, often exceed the processing abilities of EVSE hardware, main to capacity delays in danger identification and reaction.
- **Inconsistent Cybersecurity Standards:** The EVSE industry lacks a unified cybersecurity framework, resulting in inconsistent implementation of safety features. This fragmentation will increase the threat of vulnerabilities and complicates efforts to set up complete safety across one-of-a-kind systems and manufacturers.

B. Establishing the Research Niche

Addressing these gaps necessitates the improvement of modern cybersecurity solutions tailored to the EVSE environment. This thesis goal is to contribute to this endeavor through specializing in the following regions:

- **Development of EVSE-Specific Detection Models:** Designing and enforcing intrusion detection models that account for the precise operational patterns and risk landscapes of EVSE. These models will leverage domain-specific data to enhance detection accuracy and reliability.

- **Enhancement of Scalability in Security Solutions:** Exploring methods to optimize detection algorithms for scalability, ensuring that security features remain effective as the number of connected EVSE devices grows. This consists of investigating light-weight algorithms appropriate for deployment in resource-limited environments.
- **Real-Time Detection within Computational Constraints:** Developing tactics to put into effect real-time chance detection that functions correctly within the constrained computational assets of EVSE hardware. This might also involve the use of optimized machine learning models or side computing strategies to balance performance with useful resource availability.
- **Advocacy for Standardized Cybersecurity Frameworks:** Contributing to the establishment of standardized cybersecurity protocols for the EVSE enterprise by identifying best practices that can be followed universally. This standardization is essential for ensuring a cohesive and steady EVSE infrastructure.

By addressing these issues, these studies try to bridge the existing gaps and lay the foundation for extra stable and resilient EVSE structures. The insights received will clear the method bankruptcy, in which specific techniques for developing and enforcing those solutions could be mentioned.

The growing occurrence of EVSE in IoT ecosystems demands robust cybersecurity frameworks to protect the integrity and functionality of charging infrastructures. This chapter has highlighted key vulnerabilities and restrictions of conventional answers, underscoring the vital role of AI in bridging these gaps. The CICEVSE2024 dataset emerges as a critical useful resource for advancing research, offering a realistic foundation for developing tailor-made, scalable, and efficient detection mechanisms. By addressing these gaps, this study paves the way for developing stable and resilient EVSE structures, supporting the broader intention of sustainable and secure transportation infrastructure.

Chapter 3 – Literature Review

This chapter provides an extensive literature review placing cybersecurity threats into context in EVSE systems. The chapter begins with a description of the architecture of EVC systems and protocols such as OCPP and ISO 15118, both of which help standardize EV operation. The chapter describes how merging physical infrastructure with networked control systems in EVCS increases the attack surface. Next, the chapter describes the structure and data scope of the CICEVSE2024 dataset that can be used for data-driven anomaly detection, followed by a review of the variety of IDSs focused on cyber-physical systems. The chapter compares three general detecting approaches that can be used either with classical anomaly detection or with deep learning for anomaly detection. The three detection approaches are Signature based IDS, Anomaly based IDS, and Hybrid IDS. The chapter highlights existing research in IDS demonstrating progress but also limitations, made specifically on EV infrastructure, and stresses that it is important for future research incorporating IDS for EV infrastructure to take a layered, integrated with data, and adaptable approach.

3.1. Cybersecurity in Electric Vehicle Supply Equipment (EVSE)

Three charging types influence the EV adoption experience as follows and comprise a wider range of charging opportunities: Level 1, Level 2 and DC fast charging (DCFC).

- **Level 1** utilizes a standard 110-volt outlet for a charging rate of approximately 2–5 miles of range per hourly charge. It provides a slow charging experience along with the significant flexibility for charging during the night at home, suited for electric vehicle owners who have relatively low driving needs during the day.
- **Level 2** is much different in that it charges on a dedicated charging station using a hardwire connection of 240 volts; charging rates can expect 10–60 miles of range per hourly charge depending on the vehicle being charged, and the station's listed power output. If this type is charging is mostly done during the day, then it is found in public locations, including shopping centers, public parking businesses, and workplace charging options. Level 2 charging serves daily charging types extremely well.
- **DCFC** provides the least amount of charge time and is also called fast charging. This type of charging can provide up to 80% of a vehicle's battery capacity in around 30 minutes, again depending on the vehicle and the charging station power output. Most DCFC are located along highway travel routes and act as supplemental recharge station for long distance travel.

The worldwide trend for the future of EV charging has focused on more Level 2 and DCFC public and private stations as the end user charging experience has enhanced from very slow Level 1 charging through to faster charging experiences with Level 2. The future use of DCFC charging

stations will take the concept of driving beyond the range with less anxiety for the end user. This can be explained by the representation shown in Figure 5.

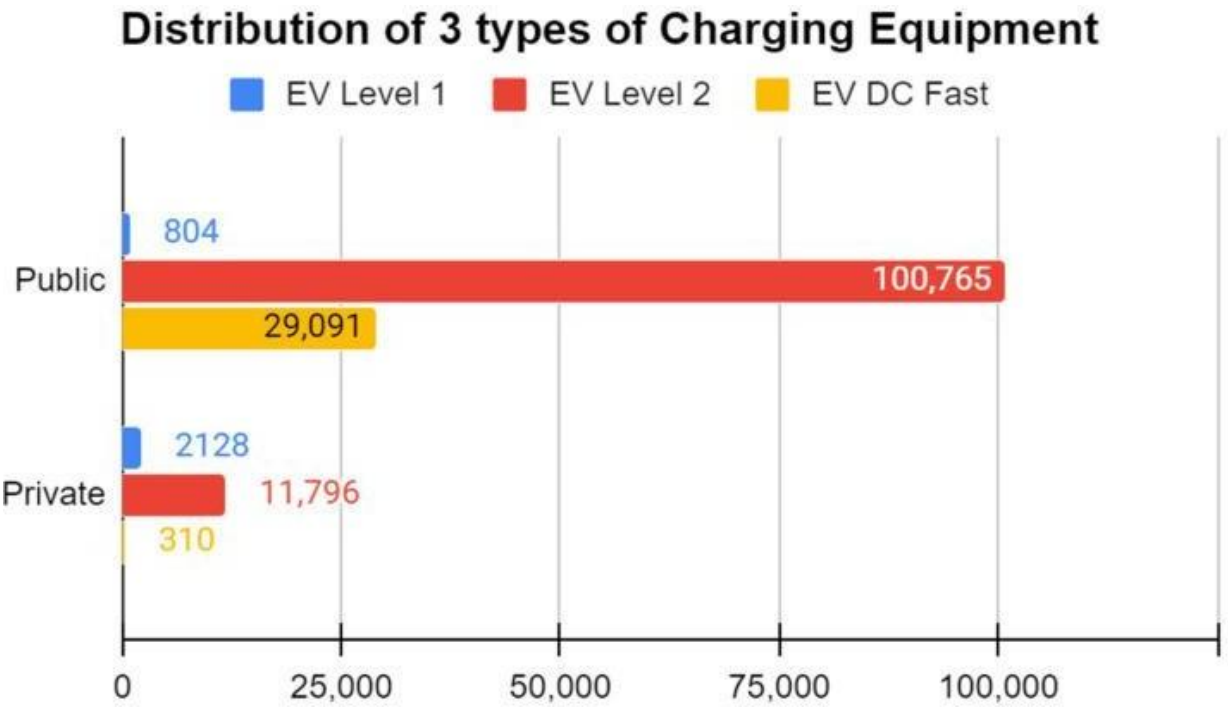

Figure 5: Factors influencing adoption of EVs.

3.1.1. Architecture of EV Charging Systems

EVCS architecture is fundamental to the incorporation of EVs into the existing energy infrastructure. The adoption rate of EVs is climbing internationally, so it is essential to fully grasp the architectural variations and operational characteristics of each type of EV charging station (level 1, level 2, DC fast charging) as they relate to EVCS systems and their V2G EVCS operational and future prospective systems. This relates primarily to the EVCS evolution from rudimentary power supply units to advanced cyber-physical systems that allow for smart charging, grid integration, and V2G incorporation. Each of these architectural components, such as common communication standards and component construction of the distinct charging equipment, introduces a wide range of cybersecurity vulnerabilities and considerations. An appreciation of the cybersecurity framework allows for successful implementation of security protections, which provide maximum security protection for EVCS, and their components.

EVSE, commonly called charging stations, plays a crucial position within the evolving smart grid ecosystem. These stations are usually classified into Level 1, Level 2, and DC Fast Charging units. Level 1 chargers provide slow charging via standard 120V outlets, while Level 2 stations use 240V AC and are familiar in public or business locations. DC Fast Chargers allow rapid charging by

delivering direct high- voltage current, making them ideal for highway and fleet operations. The shift toward smart charging, grid integration, and V2G abilities introduces both operational advantages and cybersecurity demanding situations. In the CICEVSE2024 infrastructure, two distinct EVSE types were used: EVSE-A, an industrial-degree charger, and EVSE-B, a custom low-value Raspberry Pi-based implementation designed for designated statistics tracking and experimentation.

3.1.2. Communication Protocols and Standards

The experimental testbed used to collect data for this study simulates practical EVCS surroundings with each smart and embedded device. EVSE-A represents a manufacturing-grade Level 2 charging station communicating with a remote CSMS over OCPP. In assessment, EVSE-B is a flexible, studies-orientated prototype built on a Raspberry Pi interfacing with an EVCC through ISO 15118 and connects to a nearby CSMS via OCPP. This dual-station configuration allows for the collection of rich, synchronized data from both business and experimental perspectives, presenting numerous cyber-physical situations for evaluation.

The growing complexity and connectivity of EV charging infrastructure inherently enlarges its vulnerability to cyber threats. As EVSE systems integrate with cloud-primarily based CSMS structures, utilize open conversation protocols, and function inside shared or public networks, they become prone to a huge spectrum of attack vectors. Notable threats consist of DoS attacks targeting EVSE nodes or the CSMS, efficaciously disabling charging operations and disrupting service availability. MitM attacks exploit unsecured ISO 15118 sessions, potentially permitting adversaries to intercept or control crucial parameters such as charging fees, authentication tokens, or maybe consumer identification. Beyond network-based threats, EVSE hardware—specifically systems like EVSE-B built on general-purpose operating systems—may be exploited at the firmware level, allowing backdoor set up, cryptojacking, or lateral motion into related smart grid systems.

These host-level threats are particularly risky as they could continue to be hidden from traditional network monitoring equipment, emphasizing the need for multi-layered intrusion detection processes. The CICEVSE2024 dataset displays these risks via simulating diverse attack types across network, physical, and OS layers, capturing rich data to help strong threat modeling. The advanced capabilities and interconnectivity of modern EVSEs are being realized through standardized protocols, most notably OCPP and ISO 15118. The OCPP protocol primarily supports communications between the charging station and the CSMS which coordinates the start of the session that initiates charging, authorizing the APIs, recording the session, enabling remote monitoring, and managing firmware updates. Because OCPP is implemented as a lightweight JSON-over-WebSocket protocol, it is extensible and scales well. However, it is susceptible to

spoofing or hijacking sessions if security features are poorly implemented, such as a lack of support for mutual TLS, or message integrity. ISO 15118 is a protocol standardized for V2G communications, that supports critical operations including Plug C Charge, contract authentication, and load negotiation.

Unfortunately, because ISO 15118 leverages digital certificates and signature exchanges, it is open to security weaknesses if certificate chains are compromised, or messages are intercepted. Within the CICEVSE2024 testbed, EVSE-A and EVSE-B were configured close to commercial implementations demonstrating both protocols and enabled a richness of behaviors at the protocol layer against typical operations and cyber- attacks using some degree of coordination. Because both realistic protocols were exposed against operational scenarios, it provides an authentic dataset and information for intrusion detection modeling at the protocol-level. Table 1 summarizes the EVSE communication protocols and their features.

Table 1: Summary of EVSE Communication Protocols and Cybersecurity.

Protocol	Used By	Main Functions	Vulnerabilities	Devices in Testbed
OCPP	EVSE-A & EVSE-B	- Communication with CSMS - Start/stop charging - Remote monitoring - Firmware updates	- Session hijacking - Spoofing - Lack of mutual TLS or message integrity	EVSE-A (Commercial), EVSE-B (Prototype)
ISO 15118	EVSE-B	- EVCC communication - Plug & Charge - Contract authentication - Load negotiation	- Certificate compromise - Message interception - Signature spoofing	EVSE-B only

3.1.3. CICEVSE2024 Lab Setup

The CICEVSE2024 dataset is the one utilized throughout this thesis. It was created to support machine learning-based intrusion detection specifically for EV charging environments, providing synchronized, labeled data that are structured around several layers of the system: the physical layer (power metrics), the communication layer (traffic - packet data); and the system or activity level (host log/s). The CICEVSE2024 dataset is composed of three primary data sources: the electric power consumption measurements from EVSE-B; network traffic from both EVSE-A and EVSE-B; and the host-level logs (HPCs and kernel events) from EVSE-B. Each source provides different insight into the system's behavior under benign and malicious scenarios.

The *EVSE-B-PowerCombined.csv* dataset contains power details data with time-series collected via an onboard I2C wattmeter attached to EVSE-B. Features include shunt voltage, bus voltage, current (mA), and power (mW). Data was collected while EVSE-B was on either charging or idle states. These time-series features show signals from the underlying electrical behavior of the system. Physical-layer anomalies such as sudden drops in current or spikes in power consumption may herald attacks, especially in network traffic attacks. The physical signatures in these logs allow for machine learning models to detect cyber- attacks without always relying on network traffic to do so as shown in Figure 6.

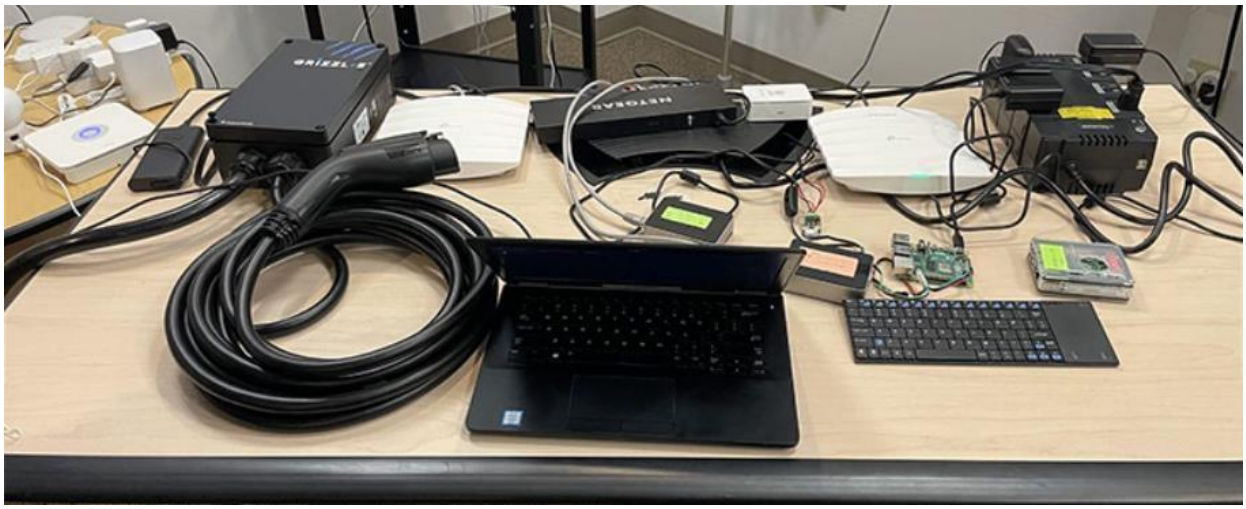

Figure 6: CICVESE 2024 EV charging station lab setup

Another key part of the dataset is *Network-Traffic-Combined.csv*, which compiles network communication logs from both EVSE-A and EVSE-B as shown in Figure 7. The logs are based on packet capture (PCAP) files and include communications over OCPP and ISO 15118 protocols. The dataset simulates several types of network attacks such as reconnaissance, SYN floods, ICMP floods, among others. The traffic patterns differ in the two logs in that EVSE-A contained typical request-response interactions with the cloud-based CSMS, while EVSE-B had local interactions and different forms of anomalous traffic associated with the attack scripts. The logs will allow classifiers to be trained to distinguish normal protocol behavior from malicious protocol behavior at the communication layer.

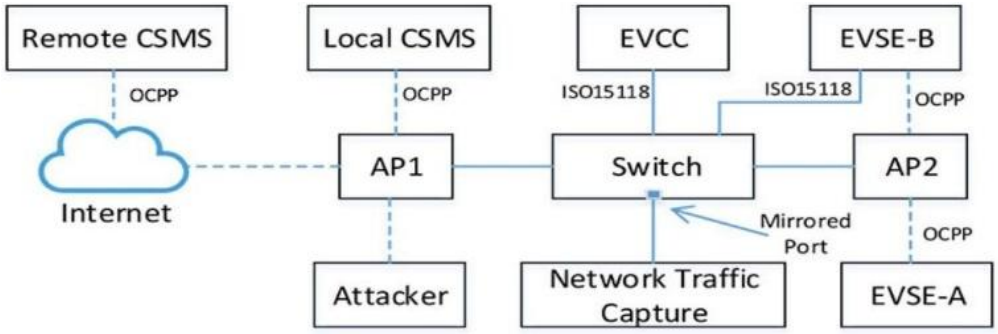

Figure 7: Network traffic for EVSE-A and EVSE-B

A distinguishing aspect of the CICEVSE2024 dataset is the low-level host activity monitoring for the EVSE-B. The `*Kernel_Events_1.csv*` file contains logs of Hardware Performance Counters (HPCs) and kernel events that were recorded directly from the Raspberry Pi hardware. Such events included process executions, memory operations, context switches, and CPU usage, each of which has valuable details related to system-level attacks such as backdoor activity or more subtle attacks such as cryptojacking. As such events are not apparent from the network, they appreciably offer an orthogonal detection layer. Therefore, host-based IDS models can now complement the prior efforts on both network-based models and power-based anomaly detection systems.

The dataset is labeled and organized to facilitate binary and multi-class classification tasks. For each data point, labels are applied to state (charging or idle), Scenario (possible types include benign, DoS, recon, and backdoor), Attack type, Interface (OCPP or ISO 15118), and Label (benign or attack). This level of labeling enables robust supervised learning methods. The text corpus also includes metadata to link the attack events across various data modalities to ensure consistency when training, validating and benchmarking detection models' performance.

The EVSE-A charger is representative of a realistic, pre-commercial deployment of a charger, with closed firmware, hardened networking stacks, and standard OCPP functionality. This charger operates in a similar way to a commercial charger normally deployed with CSMS, which will keep only logging for recent charge sessions. EVSE-B is a highly instrumented, research-grade charger. It is based on a Raspberry Pi and installed with a Linux-based operating system. This type of charger can log kernel-level events in real-time, collect sensor data via I2C interfaces, and build custom communication routines using the OCPP and ISO 15118 specifications. The EVCC mimics an EV requesting charging from the charging station, allowing the research team to control and simulate a variety of physical and protocol-level attack scenarios. Overall, these two systems work together to provide a nice collection of possibilities to conduct intrusion detection research with operational and experimental features. Figure 8 represents the technological enhancements in

EVs over the last decade.

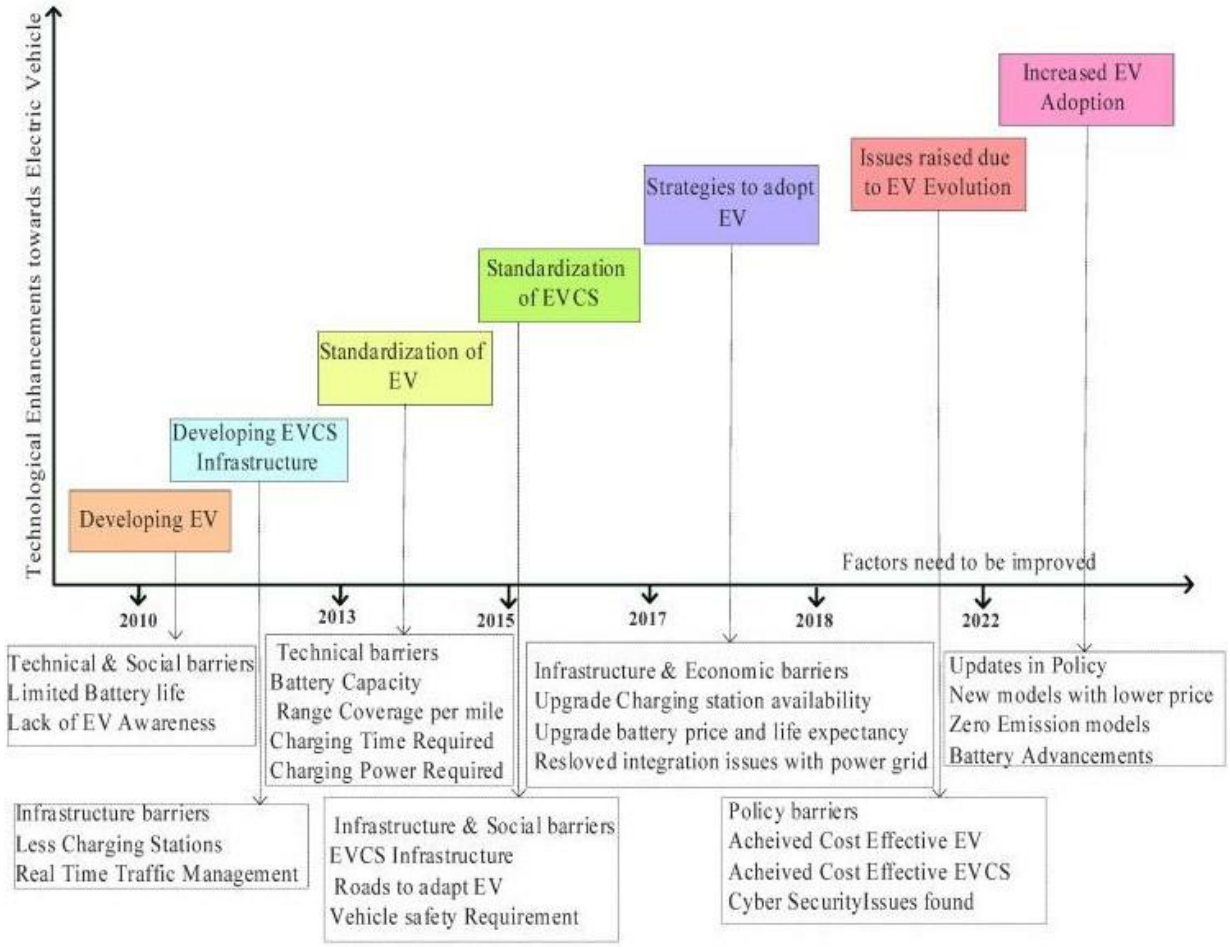

Figure 8: Roadmap of Technological Enhancements in EVs.

3.1.4. EVC stations Vulnerabilities: Real World Incidents

Within the realm of EVCS there is a growing convergence in risk of attack due to both distinct applications for EVCS in IoT as well as consistent internet connection (Rahman et al., 2024). Real-world vulnerabilities include, but are not limited to, DoS, reconnaissance, MitM, and injection attacks. Various testbed scenarios highlighted potential threats to the behavior of the communications protocols ISO 15118 and OCPP communications between EVSE and the management system. The authors examined these potential attack scenarios in real-world situations using a dataset CICEVSE2024 from real world EVCS testbeds to show potential attacks in detail. The authors also highlighted the significance of robust intrusion detection systems to ensure the stability of charging infrastructures and the security of charging infrastructures.

Hamdare et al. (2023) performs a detailed evaluation of the cybersecurity threats associated with EVCS. Their review identifies a variety of vulnerabilities relevant to the real-world application of

EVCS, including the exposure of EVCS to DDoS attacks that could disrupt charging services and cause knock-on disruption to grid resilience. The authors also identified the risk of unauthorized access to EVCS due to weak authentication, the authors deem EVCS as possible incursion points for cyber attackers. Furthermore, the lack of standardized security practices and protocols among different EVCS manufacturers lead to inconsistent security practices that increase vulnerability to cyber threats. These studies show the urgent need for enhanced cybersecurity measures such as cybersecurity frameworks and intrusion detection systems to protect EVCS architecture. [12]

Johnson et al. (2022) provides an extensive review of cybersecurity threats related to EVSE. The authors documented actual instances in which EV chargers were misused to announce unauthorized content such as anti-government messaging or pornography. These incidents demonstrate that there are little to no security mechanisms against unauthorized access. The paper also reported vulnerabilities in OVPP and ISO 15118 communication protocols that were potentially exploitable to interrupt services, cancel the sessions, or access user data unknown to the user. These examples lead the reader to understand that there is a critical need for cybersecurity in EVSE to ensure the integrity of the infrastructure as well as the privacy of the user. [13]

3.2. IDS for Cyber-Physical Systems

IDS of CPS, including smart grids and EV charging systems, are vital in preserving the integrity of such critical infrastructures by detecting threats or anomalies that can impact the operation of CPS. Because CPS entails the integration of computational resources with physical processes, disruptions created by malicious cyber events can potentially have a far greater impact than cyber events in a traditional information technology space since a change to the CPS could result in subsequent physical damage or disruptions. For example, IDS can help protect the communication networks and the control systems of smart grids that reliably transmit electricity, ensuring the stable delivery of electricity and the prevention of significant outages. Similarly, ids are necessary for EV charging stations and networks to detect potential threats that can impact the safety of the vehicle as well as the security of user data. Thus, IDS help ensure the resilience, reliability, and trustworthiness of critical infrastructure supporting the rising of societal dependence on such technologies.

The application of IDS in a cyber-physical context faces a challenge that is quite unique because of the inherent complexity and operational characteristics of CPS. The real-time operational requirement of CPS means that IDS need to detect and respond to threats instantly, avoiding latency, to not cause a disruption in operations that could have catastrophic physical impacts. Moreover, the nature of CPS applications such as smart grids and EV systems being safety-critical means that an IDS providing unacceptably high false positive and/or false negative rates, even at

a low percentage would have significant consequences, requiring high accuracy and reliability in domains like IDS implementations.

In addition, because of the convergence of IT and OT in CPS environments, the result is a hybrid security landscape that is challenging to secure completely in all instances. The mix of IT and OT means that CPS could have legacy components, systems, and protocols handling the processes, while other components, systems, and protocols handle different parts, creating a heterogeneous system that complicates the effort of applying IDS in a consistent manner, if that event is possible. Enabling IDS tools to operate readily within an existing infrastructure with insufficiently compatible components and systems in real-time is another monumental hurdle. Not only can addressing these challenges result in novel, compensatory and adaptive IDS solutions that can not only monitor, but rapidly process, synchronize, and scatter the threat detection process out in the environments multi-level and multi-layer architecture.

3.2.1. Signature-Based Intrusion Detection Systems (SIDS)

Method: Signature-Based IDS identify threats by analyzing network traffic in relation to a database of known attack signatures. In the context of IDS, signatures are known patterns associated with making attacks.

Benefits

- Very accurate at identifying known threats.
- Few false positives, due to exact matches.

Drawbacks

- Incapacitated towards new or zero-day attacks where a signature is not known.
- Signature-based databases require frequent updates.

Díaz-Verdejo et al. (2022) performed a thorough evaluation of Signature-Based Intrusion Detection Systems (SIDS) regarding their capability of detecting web-based attacks. Their research article was designed to assess the real environment of SIDS including the detection of HTTP based attacks such as SQL injection, cross site scripting (XSS), and directory traversal. [14]

The researchers established a controlled experimental setting to create both benign and malicious web traffic. This traffic was then assessed by referencing popular open-source SIDS solutions such as Snort or Suricata that utilize static rule sets for malware identification purposes. The research evaluated detection accuracy, false-positive rates, and the impact of signature rule updates.

The results underscored the established strength of SIDS detection capabilities for known attack vectors- but only if the signatures were current. The systems identified classical web attacks with minimal false positives most of the time. However, the authors observed one important limitation- SIDS cannot detect unknown attack vectors or variant attacks that develop over time (i.e., zero-days). This weakness emphasizes the need for continual signature updates and the greater need to develop hybrid and/or adaptive intrusion detection systems to surmount the weaknesses of purely signature-based systems.

3.2.2. Anomaly-Based Intrusion Detection Systems (AIDS)

Method: Anomaly-based IDS utilizes historical data to establish 'normal' behavior in the system and to detect anything outside of that established baseline. When the IDS detects behavior that deviates from the baseline this would represent a possible threat. Anomaly- based IDS are useful for detecting previously unknown attacks.

Benefits

- Can detect unknown or zero-day threats.
- Can model evolving patterns of attacks.

Drawbacks

- Higher false-positive rate, anomaly may be flagged when activity may be benign.
- Typically, a significant amount of training data is needed to form an accurate model of normal behavior.

Alsoufi et al. (2021) conducted a systematic literature review to synthesize research on the use of deep learning for AIDS in IoT contexts. The study reports the imminent need for the secure IoT ecosystem, a concern which is being increasingly scrutinized as threats to IoT ecosystems are varied and thus traditional signature-based methods are ineffective due to the heterogeneous behaviors and scale of IoT devices in addition to the digital interconnectivity of these devices. [15]

The authors systematically analyzed and synthesized the findings from more than 100 peer- reviewed studies, which covered multiple models of deep learning including CNNs, RNNs, LSTM networks, Autoencoders, and Deep Neural Networks. They evaluated these models based on metrics, used for modeling anomaly detection, which reflected their ability to accurately detect intrusions, and the false positive rates, scalability, and generalization of unknown attack patterns.

The authors report that the most feasible aspect drawn from the review is deep learning's capability for dealing with complex heterogeneous data, which importantly is high- dimensional data of IoT

anomalies. The ways in which LSTM and Autoencoders represent spatial and temporal development and patterns of detection when monitoring deviations of IoT devices from 'normal', were highlighted in study as being advantageous to any occasions of anomaly detection.

The authors also pointed out the challenges faced in the conclusion of the paper such as the following:

- The lack of quality training or defined labeled datasets from previous studies.
- The computing capacity that IoT devices can adhere to.
- The compromise is that a developer would have to consider between deep learning model complexity versus what would be necessary as a solution for real-time detection.

At the conclusion of the review, the authors give suggestions about hybrid approaches and lightweight deep learning models for successful deployment of AIDS on IoT devices in the future.

3.2.3. Hybrid Intrusion Detection Systems

Method: Hybrid IDS combine elements of both signature- based and anomaly-based systems to take advantage of both systems. The purpose of this integration is to maximize detection capabilities while overcoming the limitations of either technique.

Benefits

- Improved detection capabilities for both known and unknown threats.
- False-positive rate validations through cross-checking alerts.

Drawbacks

- Increased Complexity.
- Cost and Resource Investment.

Khraisat et al. (2020) developed a HIDS that integrates signature-based and anomaly-based detection methods into a single stacking ensemble model. The authors of the article developed the hybrid system to improve the accuracy and generalization of IDSs while maintaining low false positive rates, which are often a significant disadvantage for anomaly-based systems. [16]

Hybrid system's components incorporate the following:

- **A C5.0 decision tree classifier:** which was effective in classifying known attacks (signature-based) due to its high accuracy and interpretability, and

- **A One-Class Support Vector Machine (OCSVM):** to detect new or zero-day attacks (anomaly-based). The SVM learns each normal pattern and flags anything that deviates from the model.

The C5.0 and OCSVM components in the framework are composed as a stacking ensemble so that a meta-classifier, either a decision tree or logistic regression, then learns to screen the predicted outcomes of both a classifier and reports the final classification. In varying attack scenarios, the hybrid system can prioritize the identification of known threats or select previously unrecognized anomalies.

The system was tested on standard benchmark datasets. The hybrid IDS was validated using perpetually evolving and highly dynamic IDS datasets, including the NSL-KDD dataset. Additionally, evidence showing that the authors found when they developed their stack classification model were:

- A significant improvement with respect to detection accuracy was when standard classifiers were used.
- A reduction in individual classifiers with false positives due to the collective decision making of the ensemble effective.
- A significant increase in adaptability in classification across DoS, Probe, R2L, and U2R attack scenarios.

The authors emphasize the applicability and viability of highly scalable hybrid models to tackle unique IDS practice problems as more modern computing environments become comprised of cyber-physical and/or IoTs infrastructures with very dynamic threat vectors.

In securing CPS that encompass smart grids or EVSE, IDS become an essential part of the security stack. The convergence of digital infrastructures with physical systems creates greater risks of cyber-attack, while also creating new detection challenges associated with heterogeneous systems, real-time requirements, and the safety-critical nature of CPS.

3.3. Machine Learning for Network and Sensor-Based Anomaly Detection

Emerging in cybersecurity for CPS, there are increasing amounts of data-driven approaches which augment traditional methods for intrusion detection systems. As mentioned in the previous section, IDS can fundamentally be described in three categories: signature-based, anomaly-based, and hybrid. Signature-based detection systems use a specified list of rules to detect known threats. Hybrid approaches gather the outputs of different modalities to detect threats; however, anomaly-based detection systems are closely associated with ML approaches. As can

be expected, the ability to determine trends in large amounts of heterogeneous data means that anomaly-based systems are better suited for indicating the likelihood of an attack. The ability to model "normal" replies create detection for trend deviations indicating abnormal activity. Abnormal behavior can be derived from zero-day attacks or an attack based on a growth or strategic position that would have not been thought of as an attack.

As indicative of common 'plain vanilla' statistical learning models such as Random Forest, SVM, KNN, Logistic Regression, Gradient Boosting, among others, showcases just a few systems which might be considered. That is, in systems such as EVSE and smart grids, vulnerabilities are introduced when the physical systems/facilities that have digitally controlled mechanism do not have a similar form of protection. Hence, utilizing the Classical (relatively simple) ML models that can detect threats in as efficient a manner as possible (timeliness, resource efficiency (interpretability), and are easy to implement). More importantly in contrast to deep learning models, which require high computational (expensive) procedures involving longer training time, classical ML models require comparatively fewer resources including shorter training times and provide accessible returns for a 'user' to understand how they arrive at the result and indication. This is particularly useful with deployed systems that are acting on the behavioral alert in real time or near real time in an embedded or resource constrained system.

Moreover, CPS behaviors such as cyber-physical monitoring systems use telemetry from networks and sensors from all environments to act on structured and semi-structured datasets where possible. Further, where CPS has this advantage in employing classical supervised ML, the supervised aspect of labelled datasets on the networks and the devices and records of events logged from the hardware reports that focus on power usage or consumption trends, can all assist in training their models - to produce reports on abnormal behavior trends alerting them. These techniques reinforced one another to support the first line of sustainment strategies for the detections of threats given it encompasses consideration of the downstream risks and ensures that all layers to the defense mode maintain resilience.

3.3.1. Preprocessing Techniques

Successful preprocessing is a fundamental process in developing trustworthy, robust, ML models when dealing with any kind of anomaly detection in CPS with network or sensor-based data. Such raw data is frequently noisy, unbalanced, heterogeneous, and disorganized. If the required preprocessing is not properly conducted, the performance of the ML pipeline can be poor, biased and inaccurately generalizing to cases not included in the training sets. Three major types of preprocessing are demonstrated in this section, all of which focus on data normalization, dataset balancing, and encoding categorical variables, which could all involve data used in

cybersecurity-related ML workflows. Each of the processes listed serves a function in ensuring the input data is organized, scaled, and fits the problem space at hand.

A. Normalization

Normalization is the act of changing the numerical features to have a common scale for all features, without sacrificing the differences in value ranges. Many classical ML algorithms (KNN, SVM) depend on measuring distances between those features and therefore, features with larger magnitudes can affect the model predictions disproportionately. If there are many different variables in a CPS dataset, and the variables voltage, current, and CPU usage are all scaled differently, this should be a more relevant point.

Two of the most popular ways to normalize data is through:

- Min-max scaling, where the values are scaled to a range of values (a specified range of size, typically [0, 1])
- Z-score standardization (*StandardScaler*) where data is centered around the mean and has a unit standard deviation.

In the context of network traffic or power sensor datasets, and when normalizing the data, this ensures that packet size and current (mA) are treated equally during training, thus the model learns from patterns instead of the scale of those values.

B. Dataset Balancing

A considerable issue with intrusion detection problems is class imbalance. If the events categorizing a given class are much rarer than the other class events, a lot of the data used to build the classifier will belong to the benign events. Subsequent imbalance in data will lead the model to prioritize the majority classes while providing bad detection rates for minority classes.

- **Under sampling:** a method of matching the number of instances for the majority class to the size of minority classes to equalize the class distribution. Many will use under sampling to avoid additional time complexity for training, as well as avoiding additional memory for more instances. While there are advantages to under sampling, the actual data learning benefits of it can be disregarding potentially important information. In a CPS setting or others where attack patterns may not be as obvious or drastic, unsurprisingly removing data points from the benign group may not allow the model to learn the greater context of normal behavior.
- **Oversampling and SMOTE:** Oversampling, in contrast, is a method of copying minority class instances to retain their proportion in relation to majority class

instances. More advanced forms of oversampling are methods like SMOTE (Synthetic Minority Over-sampling Technique) that create synthetic examples of minority classes by interpolating between the instances it generates. It was recently adopted in IDS tasks due to its propensity to lower the chances of overfitting from naive oversampling.

- **Advanced Methods - AI-Based Sample Generation:** We also see the possibility of advanced methods of data augmentation, such as a few different AI-based methods. These include groups of models such as autoencoders and GAN. They can learn latent representations of the data and create high quality artificial samples, by simulating complex data distributions. In anomaly detection, for example, an Encoder-Decoder model can be trained on benign data and reconstruct normal behavior and to be able to construct relevant samples or other, generating purported instances will help the model define better other residual behavior that the model does not categorize as being normal behavior.

Although these methods carry complexities and must be validated carefully to make sure it doesn't leak data or create unrealistic samples, they offer a potential frontier to travel down in data augmentation activities for cybersecurity dataset-related tasks.

C. Encoding Categorical Variables

Categorical variables, such as protocol type (e.g., HTTP, OCPP, ISO 15118), state labels (e.g., "charging", "idle"), or attack type, must be encoded numerically to be used in most ML algorithms. The decision on how to encode a categorical variable will depend on the algorithm and the semantics of the variable.

To clarify, there are a variety of encoding techniques. The two major techniques are:

- **Label Encoding:** Each unique value of the category represented as an integer. This is useful for ordinal variables but can suggest a false ordinal relationship for nominal variables.
- **One-hot Encoding:** A separate binary column is created for each category - No ordinal relationship is implied. This is ideal for nominal data with low cardinality.
- **Binary or Hash Encoding:** This technique can be applied to categorical features that have high cardinality, so that one-hot encoding would create an unmanageable number of dimensions.

These encoding techniques should be undertaken, where applicable, to ensure that the model can leverage categorical information correctly without inferring relationships that do not exist or

introducing bias in the model. For example, when selecting and encoding protocol types from the EVSE network logs, if the protocol types are encoded with increasing integers, the model will likely assume that increasing ordered numeric values imply that the protocols have an ordinal relationship.

Preprocessing is a vital step that can influence the effectiveness, reliability, and explainability of the machine learning models of anomaly detection. The normalization of numerical features guarantees that all numerical features are treated equally, while balancing the number of classes in the dataset resolves any disparities in distributions that could negatively affect the learning. Encoding categorical variables converts abstract identities into usable input features. Normalization, balance, and encoding steps enable detecting patterns and predictable classifications. These can be the basis for effective and efficient anomaly detection systems in cyber-physical infrastructures such as smart grid and electric vehicle charging systems. Table 2 summarizes the deep learning architectures for IDS in EVSE systems.

Table 2: Summary of Deep Learning Architectures for IDS in EVSE Systems.

Technique	Purpose	Methods	Key Considerations
Normalization	Ensure uniform scale of numerical features	Min-Max Scaling 0,10,1 - Z-score Standardization (StandardScaler)	Essential for algorithms sensitive to magnitude (e.g., KNN, SVM); prevents bias.
Class Balancing	Address class imbalance to improve learning and fairness	Undersampling Oversampling SMOTE AI-based augmentation (GANs, AEs)	Balancing attack and benign classes reduce bias toward majority class and improves recall.
Encoding Categorical Variables	Convert categorical features to numerical formats	Label Encoding One-hot Encoding Hash/Binary Encoding	Encoding type depends on cardinality and variable semantics (ordinal vs. nominal).

3.3.2. Classical ML Algorithms

A. Logistic Regression

Mechanism: LR is a statistical model, commonly used for binary classification problems. It will

calculate the probability that a given input belongs to a specific category using the logistic (sigmoid) function. Logistic regression -as shown in Figure 9- calculates a weighted sum of the input features and applies logistics to map it to a probability between 0 and 1.

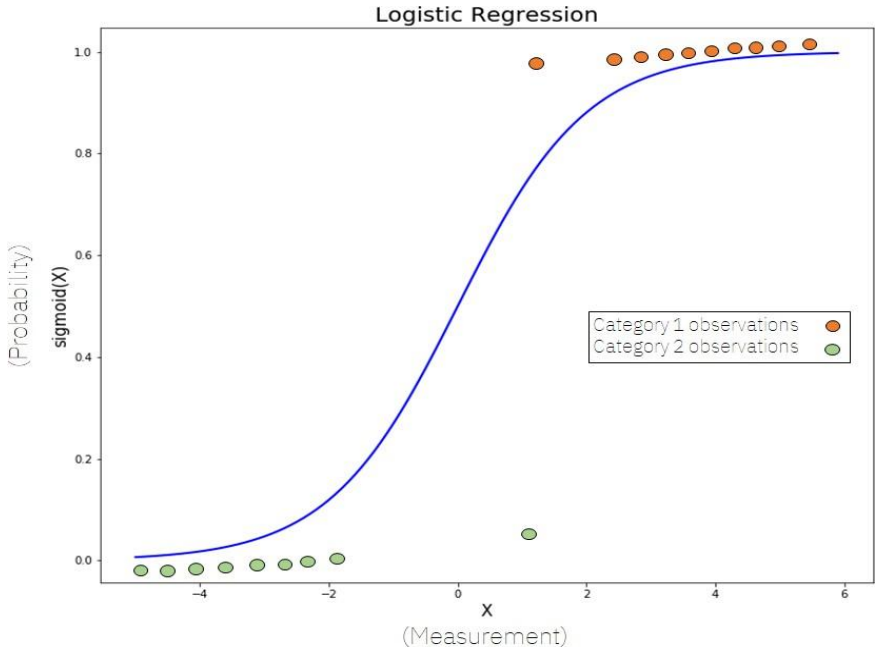

Figure 9: Linear Regression Diagram.

Advantages

- Easy to implement and simple.
- Efficient in computation.
- Output probabilities.
- Performs well with linearly separable data.

Disadvantages

- Assumes linear relationship between input variables and log odds.
- Not capable of modelling more complex relationships without feature engineering.
- Sensitive to multicollinearity among features.

Performance Context

- **Performs Nicely:** Any scenario with linearly separable data and where interpretability is important.
- **Performs Poorly:** Non-linear data patterns, or when variable interactions are complex.

Due to ease of use and interpretability, LR has been successfully deployed in detecting cyberattacks on electric vehicle (EV) charging stations. In the research conducted by Janwiri (2024), LR was executed using the CICEVSE2024 dataset to classify normal and malicious activities in EV charging systems. The model delivered a high accuracy of 94.5%, demonstrating how LR can perform binary classification applicable to cybersecurity in EV infrastructure. The research mentioned how LR performs well with linearly separable data, and LR provides probabilistic outputs which facilitate a decision-making process. [17]

B. *K-Nearest Neighbors (KNN)*

Mechanism: KNN is a non-parametric, instance-based supervised learning algorithm for classification and regression -as shown in Figure 10-. It classifies a point based on how the neighbors are classified in that space, looking at the 'k' closest training examples in the feature space provided.

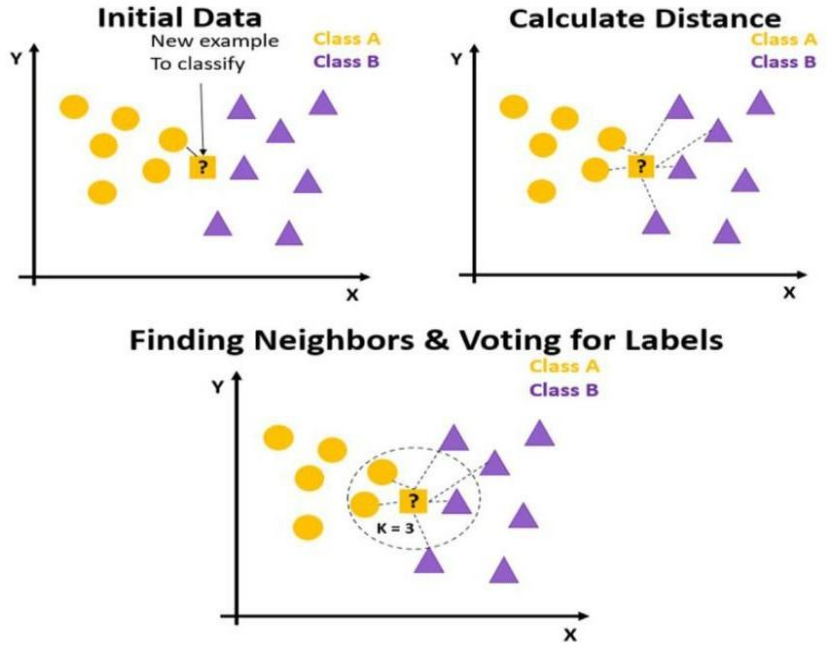

Figure 10: KNN Diagram.

Advantages

- Easy to understand and implement.
- No training phase: it is based only on the dataset at hand.
- It has multi-class classification inherently built in.

Disadvantages

- Very slow (computationally expensive) during the prediction.
- It is sensitive to the value for k and the distance metric chosen.
- It is sensitive to the curse of dimensionality during higher dimensional data.

Performance Context

- **Performs Nicely:** With smaller datasets and if the decision boundary is irregular.
- **Performs Poorly:** With larger datasets due to the time requirements for computing and in dealing with higher dimensional data.

The method K-Nearest Neighbors (KNN) has been used in EV charging stations to identify anomalies and cyber hazards. Janwiri (2024) used KNN to identify patterns in the CICEVSE2024 dataset to identify changes from the norm that reflected a possible cyber- attack. The model was able to achieve an accuracy of 92.3% indicate that it is effective when it can be assumed that similar examples can be found in proximity in the feature space. Notably, it was highlighted that while KNN is simple to establish, it is sensitive to the selection of 'k' and distance metric and requires careful tuning to achieve effective results.

C. *Support Vector Machine (SVM)*

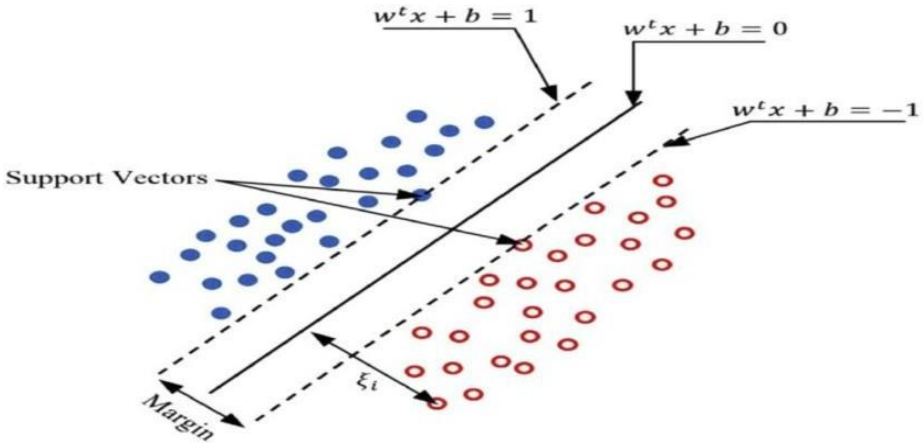

Figure 11: SVM Diagram.

Mechanism: SVM is a type of supervised learning model that uses the method of finding the best hyperplane which is the maximal separation of differing classes of data points in feature space as shown in Figure 11-. The SVM can be used for linear and non-linear classification through using different kernel functions for data.

Advantages

- Good performance in high-dimensional space.
- Good resistance to overfitting, particularly with a clear margin between classes.
- Flexibility through using different kernel functions.

Disadvantages

- Poor performance with a larger dataset, as it has longer trained time.
- Poor performance in a population that has a clear overlap between classes.
- Requires careful tuning of the parameters and the kernel function.

Performance Context

- **Performs Nicely:** high-dimensional space, or with more features than samples.
- **Perform Poorly:** with larger datasets, and datasets that have a noise and/or an overlap in the data.

SVM has demonstrated to be a powerful model for classifying cyberattacks in EV charging infrastructures with the study by Janwiri (2024) which used the CICEVSE2024 dataset to identify normal and malicious actions. The model achieved 92.9% accuracy which shows its power for identifying when high dimensional data is involved and separating classes when some separation exists. The study also showed that SVM is suitable for many areas of cybersecurity applications for EV charging stations, especially when using complex datasets that do not yield a clear linear separation of input variables.

D. *Random Forest*

Mechanism: Random forest constructs several decision trees in training time and uses the mode of all the trees' classifications in test time for classification of an observation -as shown in Figure 12-. Random forest provides a certain amount of randomness with its approach to modelling when synthesizing its trees. Instead of looking at all features, random forest selects a random set of features to build each tree, and they also randomly sample data.

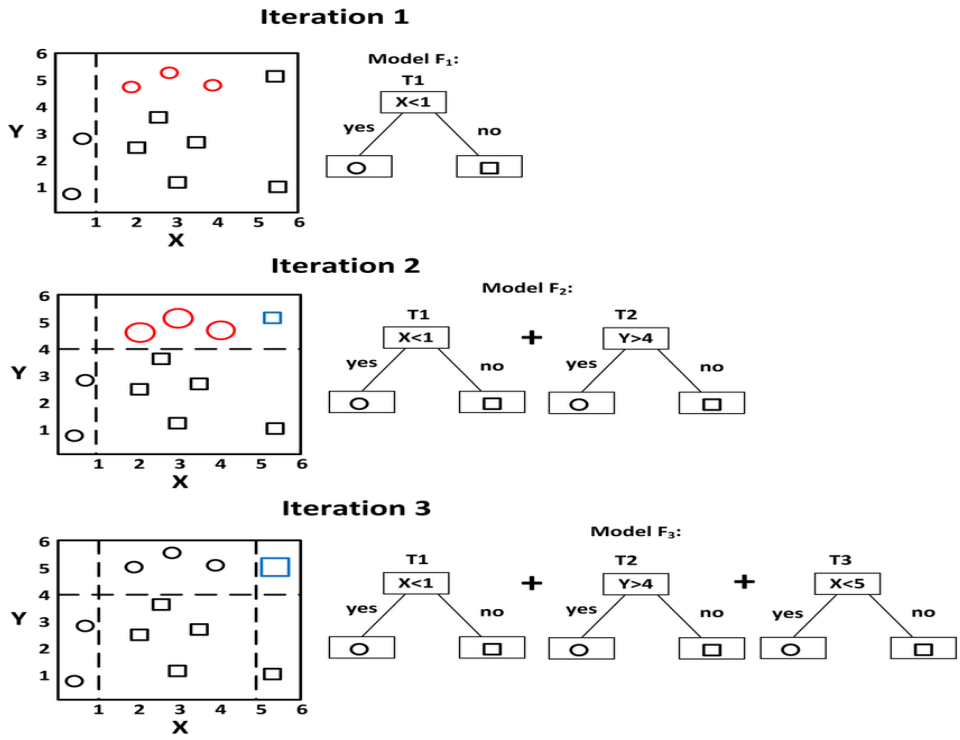

Figure 12: Random Forest Diagram.

Advantages

- Can manage large datasets, both in size and dimensionality. Overfitting could be mitigated by averaging all the interesting trees.
- It provides a measure of variable importance.

Disadvantages

- It is less interpretable than a single decision tree.
- When many trees fit, it can be computationally intensive.
- If the dataset has sparse features, it can perform poorly.

Performance Context

- **Performs Nicely:** When datasets have complexity (e.g., unbalanced datasets), then when estimating variable importance.
- **Performs Poorly:** When model interpretability is of utmost importance.

Random Forest (RF), which is an ensemble learning algorithm, has been successfully used for intrusion detection in EV charging systems. The study conducted by Janwiri (2024) applied RF to the CICEVSE2024 dataset to detect cyberattacks, scoring 95.1% accuracy. The strengths of RF, including its ability to handle large datasets with higher dimensionality and resistance to overfitting,

influenced the results of the study. Moreover, Janwiri (2024) also highlighted the utility of RF to produce feature importance estimates, as this was beneficial in establishing the factors that contribute to cybersecurity threats in EV infrastructures.

E. Gradient Boosting

Mechanism: Gradient Boosting is an ensemble learning method that is built in an additive model - as shown in Figure 13-, where each new model attempts to correct the errors made by the previous ones. This technique of Gradient Boosting builds a predictive model by bringing together several weak learners, typically decision trees, and optimizing a loss function.

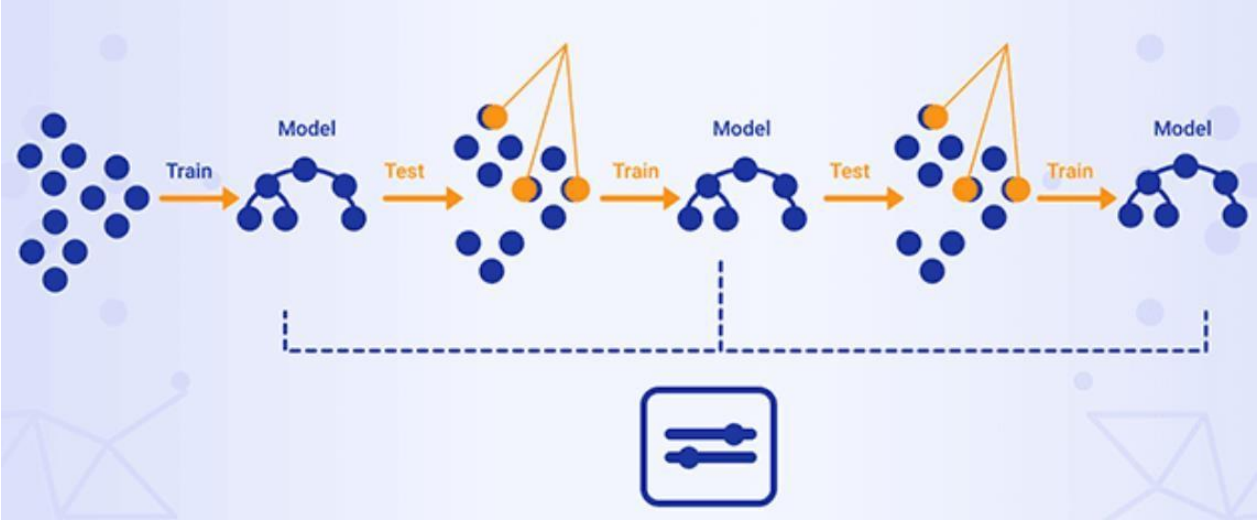

Figure 13: Gradient Boosting Diagram.

Advantages

- Great predictive performance.
- It can work with a variety of data types and loss functions.
- Can learn complex patterns.

Disadvantages

- An ill-tuned model can be easily over-fitted.
- Can be computationally expensive.
- Need proper tuning and selection of parameters.

Performance Context

- **Performs Nicely:** Gradient boosting performs well when high accuracy is beneficial, especially when complex data patterns can be leveraged.
- **Performs Poorly:** Gradient boosting performs poorly in the presence of noisy data,

especially when computing resources are limited.

GBMs have been used to improve the detection of cyber threats in EV charging stations. In the study by Janwiri (2024), GBMs were invoked on the CICEVSE2024 dataset achieving an accuracy of 93.1%. The capability for GBMs to process different types of data and loss functions and its ability to find complex variability contributed to this level of performance. The focus of the study was on parameter tuning, as it was important to reduce overfitting and obtain the best predictive accuracy in cybersecurity applications for EV charging systems. Table 3 summarizes the studied classical ML Models on CICEVSE2024 according to Janwiri's study.

Table 3: Accuracy Comparison of Classical ML Models on CICEVSE2024.

Model	Accuracy (CICEVSE2024)
Logistic Regression	94.50%
KNN	92.30%
SVM	92.90%
Random Forest	95.10%
Gradient Boosting	93.10%

After applying all five classical machine learning models - Logistic Regression, KNN, SVM, Random Forest, and Gradient Boosting - to the same dataset, CICEVSE2024, which is representative of more realistic multi-layered EV charging station data, Random Forest performed the best yielding an accuracy of 95.1%. Given that Random Forest can handle large and high-dimensional datasets and is robust to overfitting, it performed admirably in this predictive scenario.

Regarding logistic regression, Logistic Regression performed very well (94.5%) for a simple and interpretable model. Gradient Boosting (93.1%) produced strong predictive power, albeit required greater computational resources to train than the simpler models like Logistic Regression and KNN. Support Vector Machine (92.9%) provided effective separation on the high-dimensional data.

However, this comes at a cost of training complexity as it requires sufficient samples for reasonable separation across the hyperplane - it also had some noise sensitivity that could lead the model to being trained towards certain features more than others. KNN (92.3%) is a basic, conceptual model to implement, however its performance was not far off, but it was likely influenced through model location sensitivity across the distance metrics.

Nonetheless, the results show that even though models like LR provided strong performance, it is often the case that ensemble methods like RF and GB often do even better in complex cybersecurity scenarios due to their ability to capture non-linear aspects of the data and feature interaction.

3.3.3. Evaluation Metrics

A thorough evaluation of machine learning models is critical for assessing their usefulness in real-world situations and when applied to a cybersecurity context it is critical to assess the cost of misclassification. The cost of misclassification in an IDS domain for CPS including EVSE, may not be appropriately portrayed by traditional tests of a model performance. In addressing an unbalanced dataset, the best expression of model performance is typically portrayed through a balance of metrics of accuracy combined with other measures such as loss, recall, F1-score, and AUC- ROC, in training/validation, in testing, and real-world testing and computer generalization.

A. *Test and Validation Accuracy*

Accuracy is the proportion of instances that were labeled correctly with respect to the total number of instances. Accuracy is a useful overall measure of model performance for a balanced dataset, but if the cybersecurity dataset has serious imbalance, the accuracy of the model can be misleading, particularly if the majority class is where the model spends most of its time predicting.

- **Validation Accuracy** indicates how well the model generalizes to unseen data throughout training and can be useful in determining when overfitting occurs.
- **Test Accuracy** assesses the final model's ability to predict the correct class of new, real-world data.

Formula:

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

The concept of loss captures the difference between actual label and predicted output using a certain loss function (i.e. binary cross-entropy for binary classification, categorical cross- entropy for multi-class problems).

- **Validation Loss** will increase if the model begins to overfit the training data
- **Test Loss** allows the trained model to be tested for whether it can still perform on data it has not seen before.

Lower loss values generally indicate better performance with respect to predictions but should be interpreted with accuracy, and other metrics together to avoid improper conclusions.

B. *Recall (Sensitivity or True Positive Rate)*

Recall metrics the number of true positives that the model correctly classified as a positive. Recall is critically important in the cybersecurity domain and capturing most attacks is more important than

capturing benign predictions in error.

Formula:

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

A high enough recall value would indicate that most (if not all) attacks are being captured.

C. *F1-Score*

The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between detecting true positives and avoiding false positives. This is particularly relevant for IDS tasks, where both false alarms and missed detections are costly.

Formula:

$$F1 = 2 * \frac{Precision * Recall}{\text{Precision + Recall}}$$

F1-score applies to imbalanced datasets, where accuracy could be misleading because of the majority class.

D. AUC – Area Under the ROC Curve

AUC-ROC assesses the model's class discrimination ability across all threshold values. The ROC curve simply plots True Positive Rate (Recall) versus the False Positive Rate (FPR) across thresholds.

- AUC would range from a value of 0.5, where there is no class discrimination, to a value of 1.0 where there is perfect class discrimination.
- A higher AUC value will indicate a model's better ability to discriminate between types of traffic (attack vs benign traffic).

AUC is an extremely useful tool to compare several models together and consistently choose the most discriminative model for deployment into safety-critical systems (e.g., EVSE). Table 4 below summarizes the evaluation metrics described in the current section.

Table 4: Evaluation Metrics Summary.

Metric	Measures	Best For
Accuracy	Overall correctness	Balanced datasets
Loss	Model error on predictions	Detecting overfitting or underfitting
Recall	True attack detection rate	High-risk environments (safety-critical)
F1-Score	Balance between precision and recall	Imbalanced datasets
AUC-ROC	Discriminative ability	Model comparison across thresholds

Chapter 4 – Methods and Materials

Chapter 4 discusses the experimental design and methodology used in the study to measure the performance of different ML and DL approaches to detect cyber-attacks on EVSE networks. In this chapter, the reader will find a complete explanation of the datasets used—mainly the CICEVSE2024 and its various forms—as well as the data pre-processing, classification algorithms, and performance metrics. This study utilizes a combination of both classical ML models and advanced DL models to provide comparative analysis for both binary and multiclass classification problem settings. The implementation and testing protocols are provided in a way that supports consumers of the research material as well as the researchers themselves.

4.1. CICEVSE2024 dataset

4.1.1. Dataset Overview: CICEVSE2024

The CICEVSE2024 dataset, developed by the Canadian Institute for Cybersecurity at the University of New Brunswick in Canada, is a complete aid designed to develop cybersecurity research in EVCS. It features a huge variety of eventualities, shooting both benign operations and diverse cyber-attacks, thereby imparting a realistic basis for growing and comparing IDS in EVCS environments.

4.1.2. Objectives and Contributions

The primary goals of the CICEVSE2024 dataset are:

- **Comprehensive Data Collection:** To acquire a large amount of data reflecting the operational behavior of EVCS under normal and malicious conditions.
- **Multi-Dimensional Analysis:** To provide various statistics of different kinds, along with electricity intake, network visitors, and host events, facilitating a holistic evaluation of EVCS safety.
- **Support for Machine Learning Applications:** To permit the development and testing of devices by studying models for detecting and classifying cyber threats in EVCS.

4.1.3. Experimental Testbed Configuration

The dataset was generated by using a meticulously designed testbed that simulates realistic EVCS surroundings. Key additives encompass:

- **EVSE-A:** An operational Level 2 charging station speaking with a faraway CSMS via the OCPP.
- **EVSE-B:** A Raspberry Pi-based charger interfacing with an Electric Vehicle Communication Controller using ISO 15118 and with a neighborhood CSMS via OCPP.

- **Power Monitoring:** Implemented the usage of an I2C wattmeter linked to EVSE-B to file power consumption metrics.
- **Host Monitoring:** Utilization of Hardware Performance Counters (HPC) and kernel event logging to capture system-stage sports on EVSE-B. This setup guarantees the gathering of synchronized information across a couple of layers, reflecting the complicated interactions inside an EVCS as shown in figure 14.

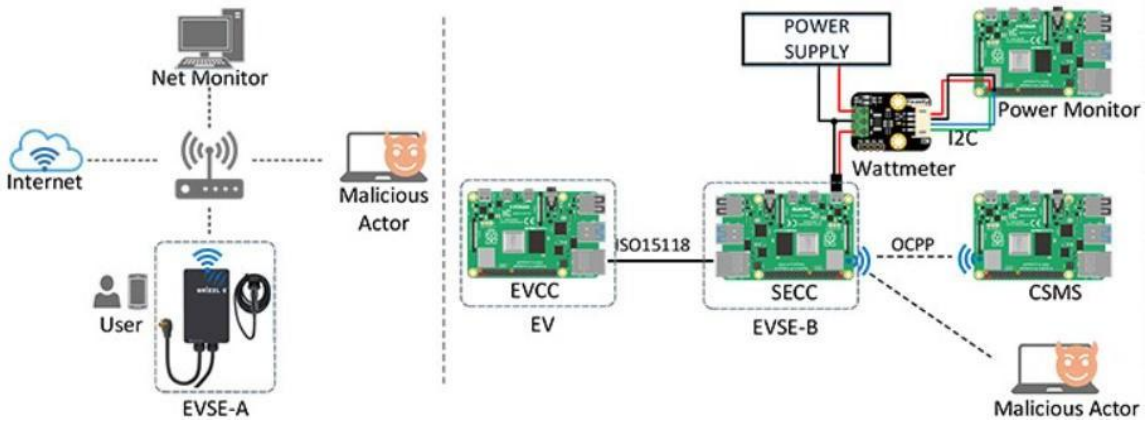

Figure 14: Overview of the EV Charging Infrastructure and Threat Model in the CICEVSE2024 Dataset.

4.1.4. Dataset Composition

The CICEVSE2024 dataset contains three number one information resources, also summarized in table 5:

- **Power Consumption Data:** Captured from EVSE-B, this consists of metrics including shunt voltage, bus voltage, contemporary, and power intake. These measurements can imply anomalies in power usage patterns associated with cyber-attacks.
- **Network Traffic Data:** Full packet captures of network communications regarding EVSE-A and EVSE-B. These records encompass each benign visitors and diverse attack vectors, imparting insights into community-level threats.
- **Host System Events:** Logs of HPC and kernel events from EVSE-B, detailing low-stage system operations. This data is vital for detecting host-primarily based attacks and systems behave under extraordinary situations.

Table 5: Dataset Summary.

Dataset Name	Source	Files Merged	Rows	Columns	Size	Classification Type

EVSE-B-PowerCombined.csv	CICEVSE2024	1	~6,700	~230	~21.78MB	Binary, Multiclass
Kernel_Events_1.csv	CICEVSE2024	1	~12,000	~220	~22MB	Binary, Multiclass
Network-Traffic-Combined.csv	EVSE-A/B	~44 CSVs	~2.7M	87	~1.1GB	Multiclass

4.2. Attack Scenarios and Labeling

The dataset includes a diverse set of attack scenarios, classified as follows:

- **Network Attacks:** Such as reconnaissance (e.g., TCP port scanning, OS fingerprinting) and DoS attacks (e.g., SYN flood, ICMP flood).
- **Host-Based Attacks:** Including backdoor installations and cryptojacking activities.

Each records instance is classified with the following attributes:

- **State:** Indicates whether the EVSE becomes idle or actively charging during information series.
- **Scenario:** Specifies interest (e.g., benign, recon, DoS, cryptojacking, backdoor).
- **Attack:** Details of the specific attack approach employed.
- **Label:** Classifies the instance as 'Attack' or 'Benign'.
- **Interface:** Denotes the conversation protocol (OCPP or ISO 15118).

This labeling allows supervised learning tactics for attack detection and classification.

4.3. Power Combined Detectors

The *EVSE-B-PowerCombined.csv* file is one of the core components of the CICEVSE2024 dataset, specifically designed to capture power consumption behavior of an EVSE machine. This record aggregates time-series measurements from the electricity monitoring module embedded in the EVSE-B testbed, which includes statistics amassed through an I2C wattmeter. Key recorded metrics consist of shunt voltage, bus voltage, cutting-edge, and power consumption, along metadata such as state (idle or charging), interface (OCPP or ISO 15118), and diverse attack state of attack scenario labels. Each row on this dataset reflects a snapshot of the EVSE's physical-layer behavior under each benign and malicious running situation, making it a vital aid for reading side-channel signatures of cyber-attacks.

The primary aim of working with the *EVSE-B-PowerCombined.csv* record on this thesis is to preprocess and remodel the raw sensor and metadata into based enter capabilities suitable for

supervised system learning obligations. This involves appearing each binary classification (distinguishing between benign and attack times) and multi-class classification (figuring out specific sorts of attacks which includes cryptojacking, DoS, or reconnaissance). By leveraging the nuanced physical behaviors recorded in this dataset, the aim is to construct and evaluate detection models capable of figuring out malicious activity primarily based totally on energy consumption patterns, a method that adds a treasured layer of protection tracking in real-world EVCS deployments.

4.3.1. Dataset Overview

The *EVSE-B-PowerCombined.csv* file, acquired from the CICEVSE2024 dataset, serves as a complete time-series log of energy-related conduct in an electric car charging station under both benign and attack situations. The document incorporates 115,298 rows and 10 columns prior to any cleaning. The dataset occupies approximately 7.94 MB on disk and is stored in a trendy CSV layout with UTF-8 encoding. Upon inspection, the dataset was located to include no lacking (null) values, making it immediately suitable for preprocessing without imputation. However, it contains 1,100 rows, which have been ultimately removed, lowering the final row count to 114,198. This raw structure presents a stable basis for supervised learning tasks, especially in domain like cyber-physical anomaly detection.

The dataset features a combination of numerical and categorical columns. As shown in table 6, all play a vital function in model development. The numerical capabilities encompass shunt_voltage, bus_voltage_V, current_mA, and power_mW, all of which can be derived from direct electric measurements. These values are continuous and vary across charging and idle states, imparting rich temporal variance suitable for category tasks. The categorical capabilities include State (with values idle and charging), interface (e.g: ocpp, none, any), Attack (seven unique attack names and none for benign), Attack-Group (3 major classes: DoS, recon, host-attack), and Label (binary: benign or attack).

Table 6: Dataset Features.

Feature	Description
Time	Timestamp of sample
Shunt_voltage (mV)	Voltage drop that occurs across a shunt resistor of I2C Wattmeter
Bus_voltage	DC Voltage supply
Current_mA	EVSE-B Current consumption
Power_mw	EVSE-B Power consumption

4.3.2. Binary classification

In this study, we focus on binary classification of EVSE network traffic, aiming to distinguish between benign communication and malicious cyberattacks. The dataset, *EVSE-B-PowerCombined.csv*, includes many electric measurements including voltage, current, power, and EVSE operational states. Our goal is to evaluate various ML and DL models for the correct detection of cyberattacks in this setting. The technique includes dataset preparation, balanced sampling, characteristic scaling, and model training with the use of classical ML algorithms including Random Forest and K-Nearest Neighbors, alongside a deep learning model built with a neural network structure. Each model is strictly evaluated using industry-standard classification metrics.

A. Dataset preparation

The dataset initially exhibited a class imbalance. Since this imbalance can critically bias classifiers, a based balancing method was applied. Attacks have been first grouped through their respective Attack-Group, and a uniform range of samples had been undersampled from each class to make sure all attack kinds had been similarly represented. The benign samples had been then undersampled to fit the full needs of attack samples, ensuing in a balanced dataset conducive to truthful model assessment.

Feature engineering involved the elimination of non-informative or identified columns along with timestamps and interface identifiers. Categorical columns like State had been numerically encoded, and numerical capabilities (e.g: shunt_voltage, bus_voltage_V, current_mA, power_mW) were standardized using z-rating normalization. The very last dataset was then split into training and testing sets with the usage of an 80/20 ratio, ensuring that stratification changed in respect where pattern sizes were allowed.

B. Machine and Deep Learning Models

- **Random Forest**

The Random Forest classifier was selected for its robustness to overfitting and its potential to deal with each numerical and specific feature efficiently. Within this dataset, its ensemble nature becomes especially valuable due to the range between attack kinds. Since every tree inside the forest evaluates a random subset of capabilities, the model captured numerous patterns of electrical conduct associated with extraordinary attack situations. Moreover, function importance evaluation provides perception into which measurements most impact attack detection—a sensible advantage in real-global diagnostics.

- **KNN**

KNN gives a non-parametric method that is based on proximity in characteristic area to make predictions. This model is incredibly intuitive and was anticipated to perform properly given that the dataset becomes normalized and balanced— crucial conditions for KNN to be effective. Because attack and benign times were represented equally, KNN had the opportunity to leverage the geometry of the feature space to split lessons.

- **Deep learning model**

The deep learning structure comprises multiple dense layers, dropout regularization, and batch normalization, making it suitable for capturing complicated, non-linear interactions in the information. Unlike conventional ML models, the neural network had the capability to examine deeper representations of the input functions. With enough training epochs and regularization strategies, the model generalized nicely across each benign and attack classes. In this case, its layered structure allowed the model to regularly abstract the electric behavior styles that are probably too subtle for less difficult classifiers to capture. The fundamental obstacle remains the call for greater training information and compute power. However, inside the scope of this balanced dataset, it presented a strong capability for nuanced discrimination.

C. Evaluation Metrics

To assess the model's overall performance objectively, we employed a set of standard classification metrics:

- **Accuracy**: This furnished a truthful measure of usual correctness. However, in binary classification with previously imbalanced classes, accuracy on its own can be misleading, that is why balancing has changed into a prerequisite.
- **Classification Report (Precision, Recall, F1-Score): This** supplies an extra granular examine of the model’s strengths. Precision indicated how nicely the model prevented false positives—essential for heading off mislabeling benign activity as malicious. Recall, conversely, highlighted the model's potential to discover all attack instances, which is paramount in cybersecurity contexts. The F1-rating balanced

these two concerns.

- **Confusion Matrix:** This allows us to visualize the true positives, false positives, true negatives, and false negatives. It was beneficial in observing model behavior in both classes and diagnosing which types of mistakes have been especially common.

These metrics collectively supply a comprehensive picture of each model's performance, permitting now not just comparative evaluation but additionally for expertise their sensible implications in a real-time EVSE monitoring system. Table 7 summarizes the preprocessing steps for binary classification.

Table 7: Summary of Preprocessing Steps and Model Pipelines for Binary Classification of the EVSE-B-PowerCombined.csv dataset.

Stage	Description
1. Cleaning	Removed redundant table and 1,100 duplicate rows to ensure integrity of the dataset.
2. Label Encoding	Transformed multiclass labels into binary categories: *Benign* vs. *Attack*.
3. Feature Scaling	Applied StandardScaler normalizes input features for better convergence of algorithms.
4. Balancing	Performed under-sampling of majority class (Benign) to match attack samples, achieving class balance.
5. Splitting	Divided data into train/test sets (80/20) using stratified sampling to preserve class distribution.
6. Random Forest	Trained an ensemble of decision trees leveraging feature importance and robustness to noise.
7. K-Nearest Neighbors (KNN)	Utilized normalized feature space to identify patterns based on proximity; effective due to balanced input distribution.
8. Deep Learning Model	Implemented a feedforward neural network with dropout and batch normalization for modeling nonlinear attack behaviors.
G. Evaluation	Used metrics such as Accuracy, Precision, Recall, F1-Score, and Confusion Matrix to evaluate each model's performance on unseen data.

4.3.3. Multi classification

The dataset used for this study, *EVSE-B-PowerCombined.csv*, captures electrical behavior information from EVSE, recorded under both benign and attack scenarios. The number one objective of this class task is to correctly become aware of the sort of cyberattack class—or confirm its absence—based totally on physical- layer measurements. The target variable, Attack-Group,

consists of four distinct classes: DoS, host-attack, recon, and none, making this a multiclass class problem. To approach this, we carried out and compared traditional machine learning models—Random Forest and Gradient Boosting—as well as a deep learning model based totally on LSTM networks. The methodology emphasizes a truthful and balanced evaluation of each model's classification abilities using carefully organized data and comprehensive metrics.

A. Dataset preparation

The initial dataset contained over 115,000 entries, with time-series information, voltage and current measurements, and categorical annotations concerning system state and attack. As a first step, we eliminated 1,100 duplicate rows to ensure statistical integrity. We additionally dropped non-informative columns which include timestamps, unique attack labels (Attack, Label), and interface identifiers (interface), which both introduced noise or redundancy.

The column State, indicating whether the system was idle or charging, becomes encoded into a binary layout. The target column Attack-Group changed into label-encoded for compatibility with machine learning algorithms. To ensure a truthful contrast throughout training and to mitigate any imbalance bias, we applied under sampling to equalize the variety of samples in line with attack labels. As a result, the final dataset was perfectly balanced with 14,300 samples for each class.

A stratified 80/20 train-check break up was implemented to keep class distribution throughout both sets. This step became vital to avoid overfitting and to keep unreliability throughout unseen records. Lastly, functions have been shuffled to do away with any ordering bias and make sure that sequence models could awareness purely on signal rather than chronological ordering, except explicitly designed to accomplish that.

B. Machine and Deep Learning Models

- **Random Forest Classifier**

The Random Forest model was selected as a robust, ensemble-primarily based baseline. Its suitability for tabular records made it a powerful candidate for early modeling tiers. In this study, it leveraged the five numerical and encoded features (shunt_voltage, bus_voltage_V, current_mA, power_mW, and State) to create a set of decor-related decision-make trees trained on bootstrapped subsets of the statistics. This architecture enabled the version to capture nonlinear interactions among electric metrics and attack categories efficaciously. A key strength of this version became its potential to offer characteristic significance, supplying insights into which physical indicators are most predictive. However, the model's reliance on begging without temporal awareness limits its ability to capture subtle, time- based adjustments that can sign sophisticated

attacks.

- **Gradient Boosting Classifier**

The Gradient Boosting model was selected to complement Random Forest with its sequential method to minimize errors. Unlike Random Forest, which builds trees in parallel, Gradient Boosting iteratively trains new trees to correct the mistakes made with the aid of prior ones. In the context of this dataset, this allowed the model to quality-tune its decision barriers in regions where training overlaps such as distinguishing among subtle recon activities and benign states. This model was especially attentive to nuanced styles in voltage and current fluctuations that easy bagging methods might overlook. While powerful, its susceptibility to overfitting—particularly in the presence of noise—necessitated careful regularization and parameter tuning.

C. Evaluation Metrics

To strictly examine the model's overall performance in this multiclass setting, we followed numerous complementary metrics:

- **Accuracy:** Represents the overall share of efficiently predicted samples. While intuitive, it can obscure overall performance disparities throughout classes.
- **Precision (consistent with magnificence):** Measures the ratio of genuine positives to overall expected positives for each class. This is especially crucial when false positives (for example: falsely predicting an attack) must be minimized.
- **Recall (according to class):** Captures the ratio of actual positives to real occurrences of each class. High recall is essential when failing to come across a real attack is unacceptable.
- **F1-Score (per class and weighted average):** The harmonic mean of precision and, F1-rating offers a balanced view and is mainly beneficial in the presence of sophistication imbalance—even though we balanced the dataset, this metric stays valuable due to subtle prediction biases.
- **Confusion Matrix:** Visual representation of actual versus predicted instructions, offering insight into unusual misclassification patterns between attack scenarios.

These metrics were computed both per class and in aggregate to evaluate the consistency and robustness of each model across the full spectrum of attack scenarios.

This study investigated the utility of machine deep learning strategies to classify cyber- physical behavior in EVSE structures, the use of each binary and multiclass class framework. The binary classification mission focused on distinguishing between benign and malicious activities, while the multiclass hassle extended this to figuring out the specific attack organization accountable for the

anomalous behavior.

In each setting, traditional machine learning models—Random Forest and Gradient Boosting—achieved a strong baseline performance, especially due to their robustness on dependent, tabular information. These models provide fast training instances and interpretable outputs, that are good for real-time monitoring systems. Their number one challenge lies in their incapability to leverage sequential facts, which in a few attack situations, consists of essential temporal cues.

To address this, an LSTM-based deep learning model was developed and optimized for each class assignment. In the binary case, the model leveraged temporal patterns to distinguish between benign and attack behaviors with high precision. In the multiclass scenario, it similarly exploited the evolving nature of attack alerts to efficiently disambiguate among nuanced classes such as reconnaissance and host-degree intrusions. While computationally more extensive, the LSTM architecture proved valuable in shooting dynamic, actual-global conduct that static models may also overlook.

Across both tasks, rigorous data preprocessing, class balancing, and assessment using complete metrics ensured the reliability and unreliability of results. The use of metrics such as precision, recollect, F1-score, and confusion matrices enabled a detailed performance evaluation that highlighted no longer just accuracy, but also the operational strengths and weaknesses of every model.

In conclusion, the integration of both traditional and deep learning techniques provided a powerful analytical framework for EVSE anomaly detection. For real-world deployments, a hybrid approach can be the handiest—leveraging system studying models for immediate, low-useful resource part detection, while utilizing deep learning models in cloud or centralized environments for extra granular and temporal risk analysis. Table 8 summarizes the preprocessing steps for multiclass classification.

Table 8: Summary of Preprocessing Steps and Model Pipelines for Multiclass Classification of the EVSE-B-Powerombined.csv dataset.

Stage	Description
1. Data Cleaning	Removed redundant second table and columns with constant values; dropped irrelevant identifiers (e.g., timestamps, interface names).
2. Label Encoding	Categorical labels (Scenario, State) were encoded numerically using LabelEncoder.

3. Feature Normalization	Applied StandardScaler to normalize numerical features, improving model convergence and comparability across features.
4. Class Balancing	Equalized samples across all classes using targeted sampling per scenario and state (Idle vs. Charging) to avoid class imbalance.
5. Dataset Splitting	Stratified split (80/20) ensured consistent class representation in training and testing sets across multiple attack scenarios.
6. Random Forest	Constructed multiple decision trees to handle multiclass decision boundaries, leveraging ensemble voting for prediction.
7. SVM	Applied SVM with RBF kernel to model complex class boundaries in high-dimensional feature space.
8. Logistic Regression	Used softmax-based multiclass logistic regression as a baseline linear classifier for interpretability and benchmarking.
G. Evaluation	Evaluated models using Accuracy, Precision, Recall, F1-Score, and Confusion Matrix; metrics computed for each individual class.

4.4. Kernel Events

EVSE forms an important part within the ecosystem of electrical mobility, ensuring the reliable and stable charging of electric automobiles. As clever infrastructure evolves, ensuring the cybersecurity of EVSE systems becomes important. The CICEVSE2024 dataset, presents a comprehensive benchmark for intrusion detection inside EVSE networks. This dataset captures a wide form of benign and malicious interactions throughout many layers, serving as a basis for evaluating cybersecurity solutions in EVSE infrastructures. Within this broader dataset.

The primary objective of this study is to expand and examine binary and multiclass classification models for anomaly detection the usage of the dataset. Specifically, the binary category assignment objectives to differentiate between benign and malicious activities, at the same time as the multiclass class task targets to identify the unique sort of attack. These type models serve as foundational equipment for imposing real-time, sensible intrusion detection structures capable of detecting anomalies in EVSE networks with high accuracy.

4.4.1. Dataset Overview

The dataset applied in this study has a file size of 21.78MB and includes time-series information derived from energy-related signals in EVSE systems. Upon preliminary inspection, the dataset

includes a great range of rows—representing individual samples— and 15 numerical functions capturing electrical behavior inclusive of voltage, contemporary, power factor, active and reactive power, among others. Each row is related to a label indicating whether the behavior is benign or corresponds to a particular attack type.

From the original CSV file, the following key information has been discovered

- Total records: 156,793 rows.
- Total features: 15 input features and 1 label column.
- Label distribution: The dataset consists of a couple of training, with the label column getting used to suggest either "Benign" or diverse sorts of cyberattacks.
- Data type consistency: All enter features were numerical, enabling compatibility with an extensive variety of conventional machine learning and deep learning algorithms.

This raw dataset serves as a wealthy and practical illustration of the operational traits of EVSE systems, taking pictures of an extensive variety of behavior styles vital for growing sturdy anomaly detection models.

4.4.2. Binary classification

The main purpose of applying the binary classification on this dataset is to build ML and DL models that are capable of distinguishing normal (benign) communication and malicious attacks. Designing a strong security mechanism that can detect such malicious activities as the first defense shield is an utmost emergency to ensure maximum security for millions of EVs.

A. Dataset Preprocessing

The original CSV report contained two appended tables. The first table contained the complete and diverse dataset used in this study, while the second table was determined to be redundant and unbalanced. Specifically, it contained a single attack class repeated across all rows, presenting no variety or value to this study. Consequently, this second phase was removed to maintain the integrity and balance of the training information.

Preprocessing steps have been completed systematically to put together the dataset for model training. The following operations were applied:

- **Cleaning:** Removal of any duplicates or inappropriate rows, particularly the second one appended table.
- **Label Encoding:** For the binary classification, the multiclass labels have been simplified to 2 classes: 'Benign' and 'Attack', where all attacks classes are grouped under a single 'Attack' label.

- **Feature Scaling:** Standardization was applied using a *StandardScaler* to normalize feature values, ensuring consistent input stages for machine learning algorithms and improving model convergence.
- **Data Splitting:** The dataset was divided into training and testing sets using an 80/20 ratio, maintaining class distributions with stratified sampling.
- **Balancing:** To address the class imbalance, we used the under-sampling technique to train the dataset. This under-sampling strategy facilitates preventing models from being biased closer to the majority class, thus improving the model's potential to detect attacks.

B. Machine and Deep Learning Models

A diverse set of machine and deep learning models were applied, each decided on for its wonderful strengths in dealing with excessive-dimensional, numerical information common in time-series cybersecurity contexts:

- **Random Forest:** An ensemble learning technique that constructs a couple of decision trees and aggregates their predictions. Random Forest is especially powerful due to its capability to handle characteristic interactions and mitigate overfitting, making it a reliable choice for structured cybersecurity data.
- **Gradient Boosting:** A powerful boosting algorithm that builds models sequentially, each correcting the errors of its predecessor. It excels at taking pictures of complicated patterns and frequently achieves excessive accuracy in tabular datasets. Its capacity to focus on tough samples makes it well-proper for detecting diffused attack signatures.
- **LSTM:** A sort of recurrent neural network especially designed to deal with sequential data and seize temporal dependencies. Given that strength alerts and EVSE behaviors are inherently time- dependent, LSTM offers the benefit of learning from past context, making it ideal for identifying evolving anomalies over time.

Each model is carefully tuned and evaluated to evaluate its capacity for generalizing well on unseen records. Selection of models is also encouraged via computational efficiency and interpretability, balancing actual-time deployment needs with overall performance.

C. Evaluation Metrics

To thoroughly determine the model's overall performance on this binary classification context, the following metrics have been employed:

1. **Accuracy:** Measures the overall proportion of correct predictions. While informative, it may

be deceptive in imbalanced datasets where the majority class dominates.

2. **Precision:** Focuses on the proportion of true positive predictions among all predicted positives. This metric is essential in cybersecurity packages, in which false positives can cause useful resource waste.
3. **Recall:** Reflects the model's potential to locate all actual attacks. A high recall is critical in this area, as lacking an attack will have excessive consequences.
4. **F1 Score:** The harmonic means of precision and recall, imparting a balanced view of model overall performance, specifically whilst managing imbalanced classes.
5. **AUC-ROC:** This threshold-unbiased metric evaluates the change-off among true positive and false negative. A better AUC suggests higher discrimination ability and robustness across numerous decision thresholds.

These metrics collectively provide a comprehensive view of each model's effectiveness in detecting attacks even as minimizing false positives, that's important for realistic deployment in EVSE cybersecurity systems. Table 9 summarizes the preprocessing steps for binary classification for the Kernel Events dataset.

Table 9: Summary of Preprocessing and Modeling Steps for Binary Classification – Kernel Events Dataset.

Stage	Description
1. Data Cleaning	Loaded the first table from the CSV file; dropped constant columns and excluded the second table, which contained only one attack type.
2. Feature Pruning	Removed irrelevant columns (time, Label, Attack, interface) that do not contribute to classification.
3. Label Encoding	Encoded State (Idle/Charging) and binary label Scenario (Benign vs. Attack) using LabelEncoder.
4. Class Balancing	Balanced samples for each scenario by equalizing the number of Idle and Charging states per class using strategic under-/oversampling.
5. Dataset Splitting	Stratified split into 80% training and 20% testing, maintaining balance across both scenarios and states.
6. Feature Scaling	Normalized numerical features using StandardScaler for consistent model input scaling.
7. Model Training	Trained classifiers: Random Forest, SVM, Logistic Regression, XGBoost, and MLP for binary scenario prediction.
8. Evaluation	Performance assessed via Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

4.4.3. Multi classification

A. Dataset Preprocessing

The primary purpose of the multi-classification task is to predict the scenario of every event primarily based on machine-stage kernel event functions. Preprocessing was important in ensuring that the dataset becomes wiped clean, converted, and balanced to facilitate fair learning by the classifiers.

The initial dataset was loaded with a CSV file containing raw kernel event data. To enhance quality, all columns with a single precise cost were discarded, as they contributed no variance to the study. Further, columns not relevant to situation prediction—inclusive of timestamps, interface data, attacks flags, and combination labels—were eliminated to prevent the model from leaking data and reduce dimensionality.

Categorical columns including State and Scenario have been label-encoded to transform string labels into numeric representations suitable for ML models. A unique recognition was given to dataset balancing. For each unique scenario, samples have been carefully selected to comprise the same range of instances where the State was Idle and Charging, based on the minimal available number of idle samples across all situations. This ensured that the models were trained on a strictly balanced dataset, decreasing class imbalance bias and allowing more robust model generalization.

After balancing, the dataset was split into training and testing subsets with 80-20 stratification per scenario, while maintaining the 50-50 distribution of Idle and Charging states within each subset. Numerical features were then standardized using *StandardScaler* to ensure all capabilities contributed equally in terms of scale, which is specifically important for distance-based totally models like SVM.

B. Machine Learning Models

To address the multiclassification task, we employed a collection of ML models known for their robustness, interpretability, and ability to seize non-linear patterns. The models selected for this study consist of:

- ***Random Forest Classifier***

The Random Forest algorithm is an ensemble of decision trees, in which every tree is trained on a bootstrap sample of the data and selections are made via majority vote. This approach is especially appropriate for excessive-dimensional and combined-type information like ours, due to its potential to address feature interactions and inappropriate functions routinely. Its inner functional mechanism also gives interpretability.

- ***SVM***

An SVM with an RBF kernel is applied to model complex selection obstacles among classes. SVMs properly seemed for their effectiveness in excessive-dimensional spaces and for maximizing the margin between class barriers. In a multiclass context, the SVM makes use of a one-vs-one strategy, building binary classifiers between each pair of classes and aggregating predictions. This method is ideal for tightly separated, linearly non- separable training like the ones in our situation-categorized kernel data.

- ***Logistic Regression (Multinomial)***

While traditionally a linear model, multinomial logistic regression is fantastically interpretable and serves as a baseline for multiclass problems. It uses SoftMax characteristics to model class probabilities and can provide calibrated performance scores. The use of the lbfgs optimizer and expanded iteration limits guarantees that convergence is accomplished even in moderately complex characteristic areas.

Each model was selected for its balance between performance, interpretability, and computational efficiency. Including a mix of linear, nonlinear, and ensemble models allows for a comprehensive assessment of the statistics' underlying structure and the models' capacity to generalize across complex styles inherent in kernel occasion sequences.

C. Evaluation Metrics

To evaluate the overall performance of our models, we employed the following metrics:

- **Accuracy:** Accuracy gives a typical indication of how often the classifier is correct. Although it is a standard metric, it may be misleading in imbalanced datasets. In our case, accuracy is important due to the strict balancing carried out during preprocessing.
- **Precision, Recall, and F1-Score:** These metrics offer a granular view of the model's overall performance for each class:
- **Precision** measures the correctness of positive predictions (i.e., the percentage of actual positives to all predicted positives), ensuring low false positives.
- **Recall** assesses the model's potential to discover all relevant instances (i.e., true positives among all real positives), helping locate under- detection problems.
- **F1-Score** is the harmonic means of precision and recall, balancing each metric and imparting a single measure of effectiveness, mainly beneficial when classes are of the same importance.
- **Macro and Weighted Averages:** The macro common treats all classes similarly and is right for comparing normal model behavior in a balanced dataset. The weighted average adjusts scores by class frequency, making sure the metric reflects the actual class distribution when evaluating real-global application.

These metrics had been selected because they align well with the classification objective and the balanced dataset setup. They permit thorough assessment of classifier behavior on both frequent and uncommon scenario types and help identify if the model overfits or neglects certain classes. Table 10 summarizes the preprocessing and modeling steps for the multiclass classification of the Kernel Events dataset.

Table 10: Summary of Preprocessing and Modeling Steps for Multiclass Classification – Kernel Events Dataset.

Stage	Description
1. Data Cleaning	Dropped second table and columns with constant or irrelevant values (time, Attack, Label, interface).
2. Label Encoding	Categorical variables State and Scenario were encoded using LabelEncoder for compatibility with models.
3. Class Balancing	Balanced all Scenario classes by sampling equal numbers of Idle and Charging rows for each class.
4. Dataset Splitting	Used stratified 80/20 train-test split, ensuring class and state balance within folds.
5. Feature Scaling	Standardized numeric features to zero mean and unit variance with *StandardScaler*.
6. Classical Models	Trained Random Forest, SVM, and Logistic Regression on the balanced multiclass dataset.
7. Deep Model (GRU)	Designed a GRU-based RNN with batch normalization, dropout, and early stopping; trained with 3-fold Stratified K-Fold cross-validation.
8. Evaluation	Evaluation used Accuracy, Precision, Recall, F1-Score, and Confusion Matrix across classes; t-tests were also applied for statistical insight.

4.5. Network Traffic

The CICEVSE2024 Network Traffic Dataset is part of the complete CICEVSE2024 dataset. This dataset captures sensible, labeled traffic flows from EVSE environments and targets to assist the development of IDS tailor-made for the smart charging infrastructure of electric vehicles. The dataset consists of packet-stage data reflecting lots of benign and malicious behaviors throughout many styles of attack vectors, inclusive of DoS, reconnaissance, and protocol-particular exploits. These flows are categorized and timestamped, enabling time-based and flow-primarily based cybersecurity modeling.

4.5.1. Dataset Overview

The dataset was initially distributed as multiple CSV files split across two directories: EVSE- A and EVSE-B, each hosting numerous varieties of traffic captures. To streamline analysis, all CSV files were downloaded programmatically using a custom script contained in the *Download_Data.ipynb* notebook. This script employed web scraping strategies through the requests and *BeautifulSoup* libraries to iterate over all document hyperlinks and download them automatically. Once collected, the documents were merged into a single comprehensive CSV

file, *Network-Traffic-Combined.csv*, which consolidated all labeled traffic flows across resources.

The merged *Network-Traffic-Combined.csv* dataset occupied about 1.1 GB and contained 2,744,700 rows across 87 columns. Each row represents a flow-level network event, with functions ranging from IP/port metadata and packet statistics to software-layer identifiers. A total of 66 unique application names and 18 software classes were determined, with the majority categorized as Unknown or Unspecified. All rows included at least one missing value, and 4 columns had a single constant value and were eliminated. The label column label_mul featured 43 attack types, closely imbalanced, with some training dominating the dataset. IP communication was restricted to 121 precise (src_ip, dst_ip) pairs, reflecting the dependent environment of EVSE communications. These insights informed the preprocessing method and justified focusing on multiclass class.

Given the multiclass nature of the labeled attacks, and the first-class granularity provided in *label_mul*, the study completely centered on multiclass classification. Binary classification is prevented since the subtleties between various attacks are essential for realistic intrusion detection. For example, distinguishing between a SYN Flood and a Slowloris experiment can substantially impact mitigation strategy.

4.5.2. Dataset Preprocessing

To prepare the data for the model's training and ensure the most beneficial memory utilization, a radical preprocessing pipeline was carried out:

a. **Column Filtering:** Columns with low variance or excessive missing values (for example: vlan_id, content_type) had been excluded to reduce noise and RAM utilization.
b. **Row Cleaning:** Rows with missing values were dropped to ensure the model's balance, mainly while training graph neural networks that require dense representations.
c. **Label Normalization:** The *label_mul* column, which in the beginning contained verbose labels (i.e., charging syn flood), was standardized into 15 steady class names which include syn-flood, competitive-scan, and vulnerability-test. Additionally, the wider Scenario column grouped labels into superclasses like Benign, DoS, and Recon.
d. **Feature Engineering:** IP and port records were merged to assemble precise conversation endpoints (src_ip: port), facilitating graph creation.
e. **Encoding:** Categorical columns like application_name and application_category_name were label-encoded for model compatibility.

The final smooth dataset was stored as CICEVSE2024_CLEAN.csv.

4.5.3. Model Architecture and Training Strategy

To cope with the complexity of flow-based network intrusion detection, we employed a memory-efficient edge classification method using the GNNs applied with PyTorch Geometric. The dataset was first represented as a heterogeneous graph, where IP address pairs (source and destination) formed the edges, and the numerical network float information acted as edge features. Node features were initialized as learnable embeddings of fixed dimensionality (32 in our configuration). These edges—every representing a network flow had been stratified and split into training and testing sets using an 80-20 split while preserving class distribution. The architecture was composed of a GCNConv layer to embed node representations, followed by an edge encoder for the flow-level features, and a completely related classifier that blended source and vacation spot node embeddings with encoded side attributes to output multiclass predictions across 15 attack kinds (including benign).The training method was carefully designed to balance performance with GPU memory efficiency:

1. **Gradient Accumulation:** Instead of updating model weights after each batch, gradients were accumulated over several small batches and then used for a single update. This significantly decreased memory utilization at the same time as permitting powerful batch-level learning.
2. **Stratified Data Splitting:** Edges were stratified by their multiclass labels to ensure proportional representation of all instructions in both training and test units, reducing the chance of class imbalance throughout model learning.
3. **Class Weighting:** To deal with label imbalance, class weights inversely proportional to class frequency have been computed and incorporated into the loss characteristic, supporting the model paying more attention to underrepresented classes.
4. **AdamW Optimizer:** We employed the AdamW optimizer with weight decay, which offers better generalization overall performance in deep learning tasks by way of decoupling weight decay from the optimization step.
5. **Loss Function:** Cross-entropy loss was used, changed with class weights to accurate for the skewed distribution of the flow labels.
6. **Batch-wise Memory Clearance:** After each forward and backward skip, intermediate tensors have been deleted, and GPU memory is cleared explicitly to keep away from memory fragmentation and overflow.
7. **Epoch-wise Monitoring:** Training metrics which include loss and accuracy were logged after every epoch, and a custom plotting application was used to visualize the learning trends across epochs.

8. **Edge-Level Supervision:** Unlike traditional node type, supervision is provided on the edge level, aligning with the flow-based nature of the dataset and making the model mainly appropriate for detecting attack styles among communicating nodes.
9. **Modular Design:** The model was encapsulated into reusable modules for clean model and experimentation with different GNN layers and training parameters.

This robust approach enabled efficient training of a deep GNN model on a big-scale, excessive-dimensional dataset with the use of limited computational sources while reaching aggressive accuracy and generalization overall performance.

4.5.4. Evaluation Metrics and Visualization

The trained GNN model is evaluated on the test dataset using a mixture of quantitative metrics and visual tools to evaluate its predictive overall performance and typical reliability. Evaluation was executed at the edge level, where every network flow was categorized into one of the predefined attack classes or categorized as benign. The category report is generated using scikit-research's utility, and a confusion matrix was plotted to offer a visual understanding of the distribution of predictions throughout genuine training. In addition to scalar metrics such as accuracy and F1-rating, training history was visualized via line plots showing the evolution of training loss and accuracy over 300 epochs. These visualizations helped identify patterns which include convergence developments, overfitting, or under-fitting, even as the confusion matrix uncovered which precise attack types have been most frequently confused by the model.

The evaluation employed the following metrics:

- **Accuracy:** To measure the portion of successfully classified flows over the total wide variety of flows.
- **Precision:** To examine how some of the expected attack times were surely correct, especially vital for minimizing false positives.
- **Recall:** To investigate what number of real attack times were correctly detected, which is critical in safety applications.
- **F1-Score:** To offer a balanced degree that bills for each precision and, recall, useful within the presence of sophistication imbalance.
- **Confusion Matrix:** To visualize and analyze according to class predictions overall performance and identify frequent misclassifications.

These metrics have been selected to provide a complete assessment of the model's classification capability, specifically in scenarios where imbalanced class distributions and the cost of misclassification vary drastically throughout attack types. Table 11 summarizes the

preprocessing and modeling steps for the multiclass classification task of the Network Traffic dataset.

Table 11: Summary of Preprocessing and Modeling Steps – Network Traffic Dataset.

Stage	Description
1. Data Acquisition	Downloaded and merged multiple CSV files from EVSE-A and EVSE-B subdirectories using a custom script (Download_Data.ipynb).
2. Initial Inspection	Verified dataset structure (87 columns, 2.7M rows, 1.1GB size), identified missing values and constant-value columns.
3. Label Construction	Extracted label_mul from file names, then mapped fine-grained attacks to generalized classes; generated label_bi and Scenario columns.
4. Data Cleaning	Removed irrelevant columns (e.g., MACs, ports, ID columns), unified IPs and ports into strings, removed columns with a single unique value.
5. Label Encoding	Encoded categorical columns such as application_name and application_category_name using LabelEncoder.
6. Graph Construction	Built a memory-efficient graph using PyTorch Geometric, with IPs as nodes and traffic events as edges. Edge features included 60+ traffic stats.
7. Feature Scaling	Standardized edge-level features using StandardScaler before feeding into the model.
8. Dataset Splitting	Performed stratified split (80/20) of edges to maintain distribution across 15 attack classes.
G. Model Architecture	Developed a memory-optimized GCN-based model using node embeddings and edge features to classify edges into 15 attack types.
10. Training Strategy	Applied class weighting, mini-batch training, gradient accumulation, dropout, and GPU memory tuning to handle large- scale graph efficiently.
11. Evaluation	Assessed with Accuracy, Precision, Recall, F1-score, and Confusion Matrix; visualized training curves and classification performance.

The experimental portion of this research is established in Chapter 4. The systematic preprocessing of data, model development, and evaluation of multiple datasets gives the chapter a solid base for the discussion of the results in the second half of this report. This chapter tackles the customized methods required to accomplish both binary and multiclass classification, and how to achieve the essential trade-offs between classical and deep learning. The results that were shown on applicable models such as Random Forest, Gradient Boosting, and LSTM sets up for an informed discussion in the next chapter.

Chapter 5 – Results and Discussions

The fifth chapter presents the interpretation and analysis of the experiment results from the ML and DL models described in the previous chapter, from a comprehensive assessment of model performance using evaluation metrics like accuracy, F1-score, recall, precision, and confusion matrices. Comparative perspectives across the different models discussed in this chapter, assessing the experiment results across two tasks, the binary task and the multi-class task, and across datasets. The chapter also discusses the ramifications of the findings related to EV charging infrastructure in the real-world context, and highlights reflections of the challenges, gaps, and opportunities for improvement.

5.1. PowerCombined Detectors

5.1.1. Binary Classifiers

In this section, we report the performance of a binary classification task performed on the *PowerCombined* dataset using 3 models: Random Forest, KNN, and a fully connected feed-forward neural network. The models were trained in encoded and scaled raw data without any additional feature engineering. The deep learning model included drop out and batch normalization as regularization techniques. The performance results are provided below.

A. Results Comparison

The evaluation of the trained models is conducted on a balanced test set along with 5720 samples, with an equal distribution between benign and attack classes. As shown in table 12 that compares the 3 models' performances, the following metrics were recorded for every model:

Table 12: Results Comparison of the 3 models.

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	TP	FP	FN	TN
Random Forest	0.9217	0.94	0.9	0.92	2583	171	277	2689
KNN	0.9136	0.93	0.9	0.91	2564	198	296	2662
Deep Learning	0.904	0.96	0.85	0.9	2419	108	441	2752

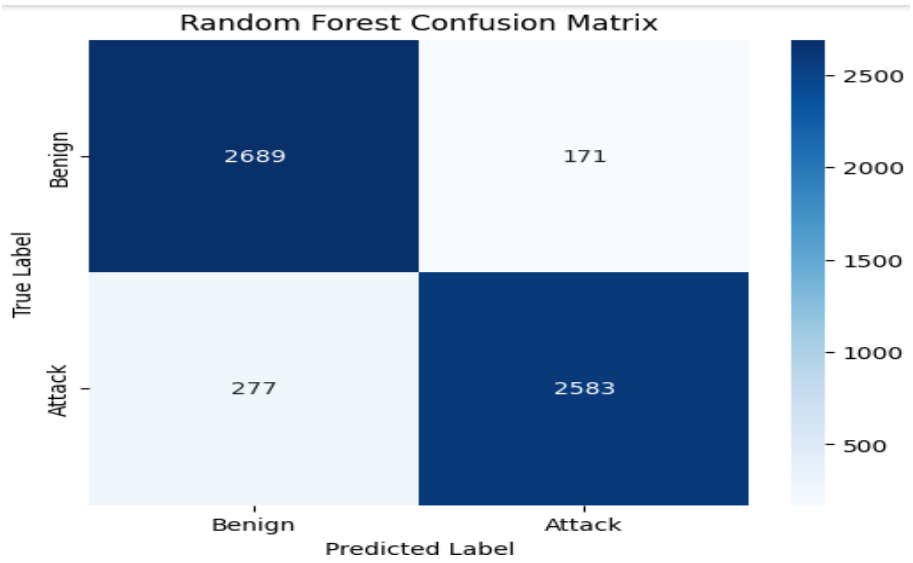

Figure 15: Random Forest Confusion Matrix.

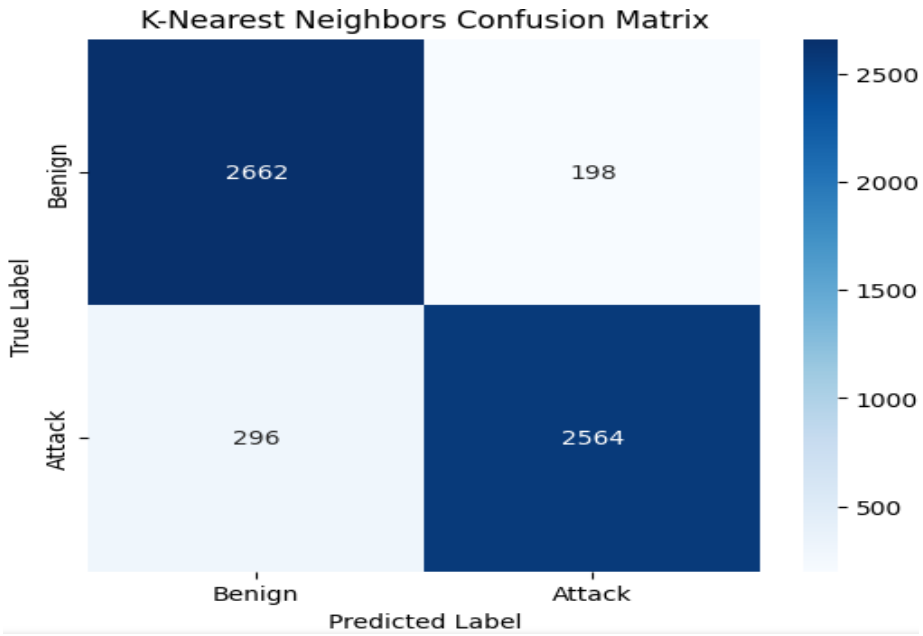

Figure 16: K-Nearest Neighbors Confusion Matrix.

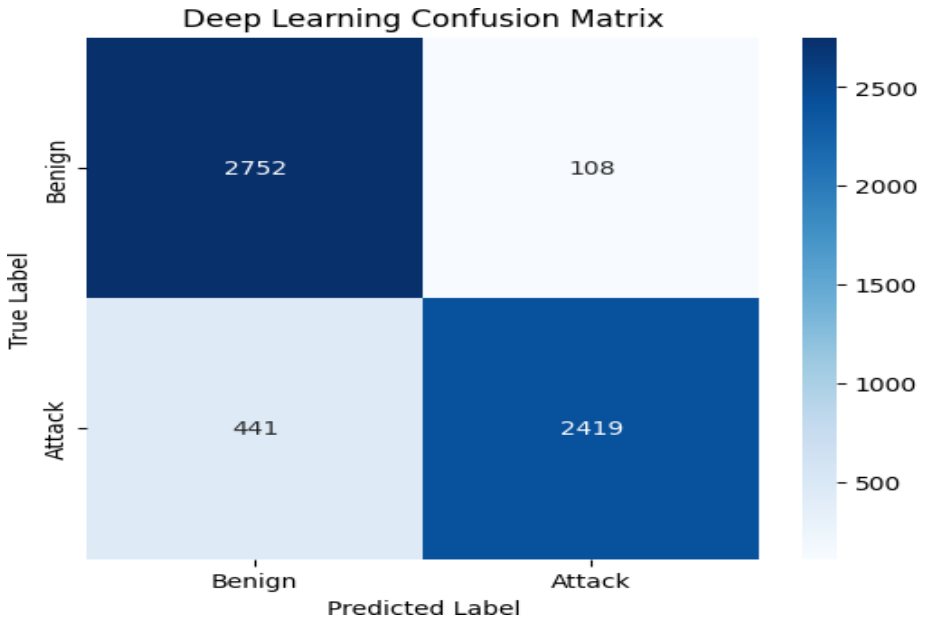

Figure 17: Deep Learning Confusion Matrix.

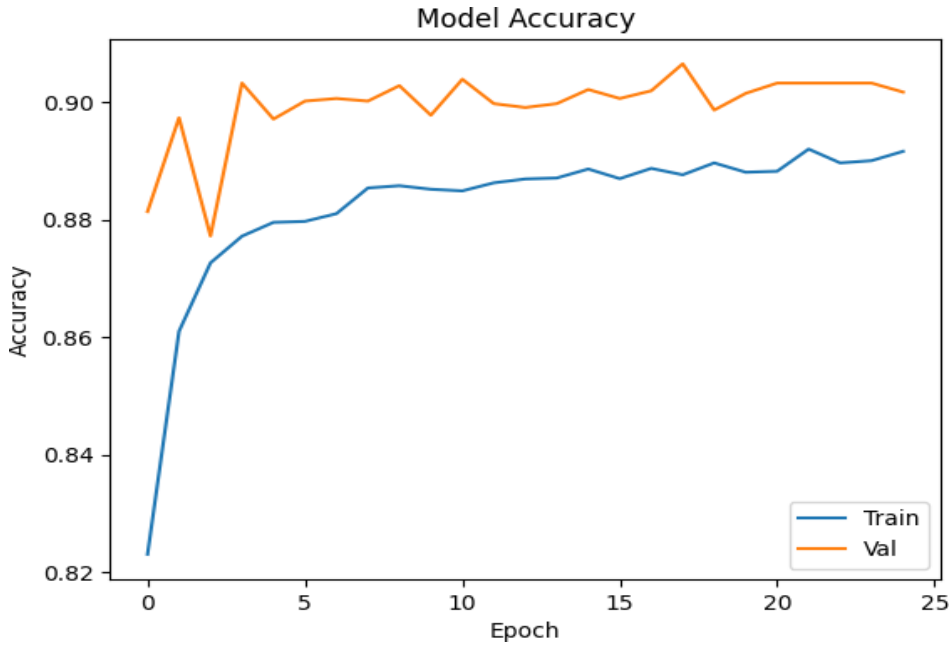

Figure 18: Deep Learning train and validation Accuracies.

Keywords:

TP: True Positive, **FP:** False Positive, **FN:** False Negative, **TN:** True Negative.

Figures 15, 16, 17 represent the confusion matrices of the 3 models, while Figure 18 compares the train and validation accuracies of the deep learning model. All three models performed high predictive overall performance, with the Random Forest classifier main in normal accuracy. The deep learning

model showed barely decrease recall, however better precision for the malicious class, indicating a more conservative but precise detection behavior.

B. Discussion

The binary classification experiments on the *PowerCombined* dataset highlight the effectiveness of numerous machine learning and deep learning approaches in identifying anomalous charging behavior. Among the three models evaluated, Random Forest, KNN, and DNN all demonstrated high predictive capability, with accuracy rankings exceeding 90%. However, every model displayed precise behavioral inclinations in precision, recall, and error distribution.

The Random Forest classifier had the highest overall accuracy (92.17%) and F1-score (0.92), suggesting a nicely balanced trade-off between sensitivity and specificity. This overall performance may be attributed to its ensemble nature, which permits it to handle non-linear relationships and resist overfitting, even with confined feature engineering. The incredibly low false positive (FP = 171) and false negative (FN = 277) counts in addition, make stronger its robustness across both benign and malicious classes.

The K-Nearest Neighbors model intently accompanied, with a barely reduced accuracy (91.36%) and a similar recall (0.90). KNN's instance-based reasoning allows it to carry out properly on established datasets where comparable samples cluster clearly. However, the slightly better false positive rate (FP = 198) can also suggest a few misclassifications of benign behavior as malicious under ambiguous conditions, a recognized problem when using KNN on high-dimensional or noisy data.

Interestingly, the Deep Neural Network model presented the very best precision (0.96) but the lowest recall (0.85). This behavior shows that the model turned into more conservative in labeling a pattern as malicious, possibly due to regularization mechanisms like dropout and batch normalization that reduce overfitting but can also increase decision boundaries' sharpness. Consequently, while DNN produced the lowest false positive count (FP = 108), it also missed a higher quantity of actual attacks (FN = 441), reflecting a trade-off that favors minimizing false alarms on the fee of a slightly better chance of undetected threats.

The confusion matrices and the training-validation accuracy plots (Figures 1–4) support those conclusions by visually illustrating model self-assurance, generalization types, and convergence behavior. Notably, the DNN maintained solid training and validation curves, indicating effective generalization regardless of its reduced recall.

Overall, Random Forest proved to be the most balanced and generalization model for this binary task, appropriate for deployment in real-time detection environments. However, DNN’s advanced precision can be great in contexts in which false positives are especially expensive, consisting of computerized system responses that would disrupt charging operations.

5.1.2. Multiclass Classifiers

The multiclass classification task around the *PowerCombined* dataset was to classify electrical charging behavior into 4 classes: DoS, host-attack, none, and recon. This task allowed for a deeper inspection of EVSE-B activity beyond binary classification. This task followed two machine learning models, Random Forest and Gradient Boosting. Pre-processing steps for the models included dataset balancing, label encoding, and feature scaling to facilitate the models with consistency across the task. The results of the multiclass classification task are shown in the next section.

A. Results Comparison

Every model was evaluated on a stratified, balanced test set of 11,440 samples (2,860 per class). Accuracy and weighted F1 rankings were used to evaluate average performance, at the same time as in step with-class precision, recall, and F1-score offered deeper insight into each classifier's behavior. The confusion matrices visually summarized the class's particular strengths and weaknesses. Table 13 compares the models' accuracies and F1-Score.

Table 13: Models Performance Comparison.

Model	Accuracy	Weighted F1 Score
Random Forest	0.7034	0.7022
Gradient Boosting	0.7173	0.7005

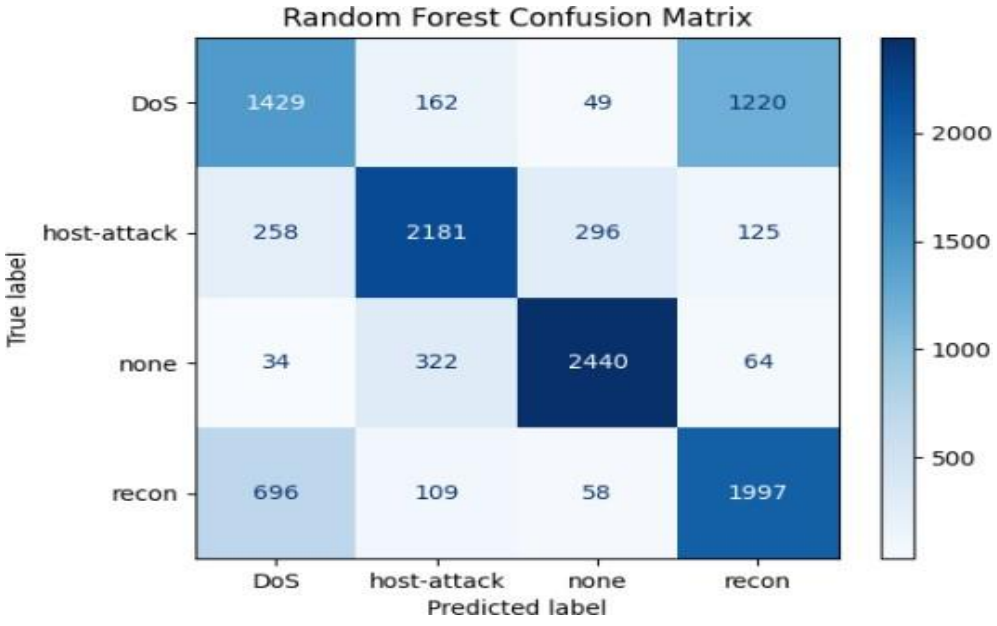

Figure 19: Random Forest Confusion Matrix.

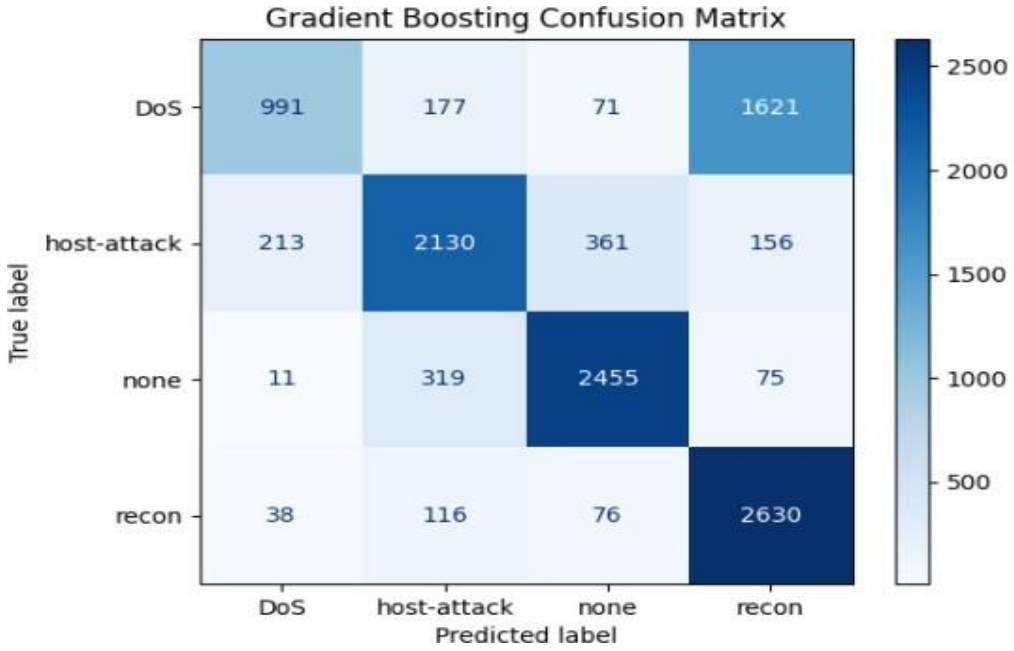

Figure 20: Gradient Boosting Confusion Matrix.

Figures 19 and 20 represent the model's confusion matrices. These metrics mirror the models' competencies to differentiate between nuanced attack patterns, with each displaying unique strength in spotting the "none" and "host-attack" classes. Minor performance trade-offs were discovered between precision and recall, depending on the classifier and class. Table 18 compares the performances of each model on each class, while Figure 21 visualizes the results of each model for comparisons purposes.

Table 13: Per-Class Metrics.

Model	Class	Precision	Recall	F1-Score
Random Forest	DoS	0.59	0.5	0.54
	host-attack	0.79	0.76	0.77
	none	0.86	0.85	0.86
	recon	0.59	0.7	0.64
Gradient Boosting	DoS	0.79	0.35	0.48
	host-attack	0.78	0.74	0.76
	none	0.83	0.86	0.84
	recon	0.59	0.92	0.72

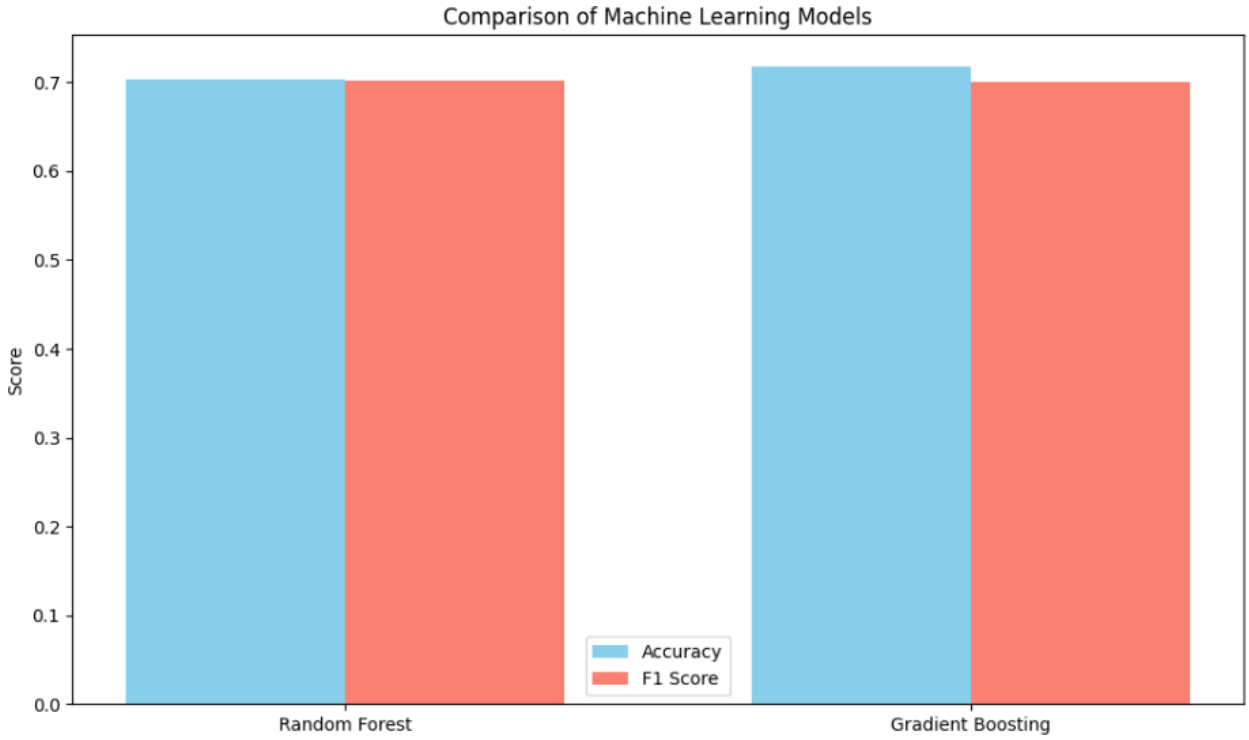

Figure 21: Comparison of the Machine Learning Models Accuracies and F1 Score.

B. Discussion

The multiclass classification results of the *PowerCombined* dataset reveal that both Random Forest and Gradient Boosting models demonstrated stable functionality in distinguishing between the 4 described behavioral classes: DoS, host-attack, none, and recon. Trained and tested on a cautiously stratified and balanced dataset (2,860 samples per class), the models' performance metrics provide significant insights into their effectiveness and trade-offs in a real-world EVSE security detection

context.

In terms of overall accuracy, Gradient Boosting barely outperformed Random Forest (71.73% vs. 70.34%), though the difference was distinctly minor. The weighted F1 rating, which accounts for class imbalance and overall performance consistency across training, was comparable among the two (0.7005 for Gradient Boosting vs. 0.7022 for Random Forest), suggesting both models are almost equivalent in combination overall performance.

However, per-class metrics provide an extra granular view of the model's strengths. For example, Random Forest exhibited extra constant performance throughout all classes, with F1-scores starting from 0.54 (DoS) to 0.86 (none). It mainly excelled at detecting none and host-attack scenarios, reflecting sturdy generalization for benign and often occurring attack types.

In comparison, Gradient Boosting confirmed better precision for the DoS class (0.79), but its recall was appreciably decreased (0.35), indicating that at the same time as it made fewer false positive DoS predictions, it additionally overlooked more real DoS samples. This pattern displays conservative bias in the model, favoring precision over recall for that class. Interestingly, Gradient Boosting excelled at identifying recon behavior, attaining the very best recall (0.92) throughout all class-model mixtures. This shows a robust sensitivity to reconnaissance patterns, which is a precious trait in early threat detection systems.

The confusion matrices visually verify these interpretations, highlighting Random Forest's greater balanced misclassification rates and Gradient Boosting's sharper class-unique differences. Furthermore, the performance assessment chart (Figure 7) demonstrates the minor, however applicable divergence in universal accuracy and F1 scores, underscoring the effect of those trade-offs.

Finally, even as each model is a viable multiclass classifier for EVSE security tasks, the selection among them must consider operational priorities. Random Forest is prime for wide-spectrum reliability throughout attack sorts, whereas Gradient Boosting can be better suited for centered precision, especially in identifying stealthy or reconnaissance-primarily based threats. These findings emphasize the class of consistent with-class evaluation whilst designing robust cyber-physical system defenses.

5.2. Kernel Events Detectors

5.2.1. Binary Classifiers

The binary classification task for the *Kernel_Events_1* dataset was built to differentiate between regular and malicious network behavior throughout numerous experimental situations. This way of classification is vital for real-time anomaly detection in cyber-physical structures, along with EVSE. To ensure robust evaluation and equitable illustration of attack classes, the dataset underwent rigorous

preprocessing. This blanketed one-hot encoding of categorical features, z-score standardization of numerical fields, and a strategic sampling strategy to ensure the same illustration of classes throughout all unique experimental scenarios.

Two machine learning classifiers—Random Forest and Gradient Boosting—have been deployed to model this classification problem. Both are ensemble techniques, known for their effectiveness in established tabular data and their resilience to overfitting. No manual feature engineering is achieved beyond preprocessing; the models operated on a balanced dataset composed of 30,000 samples for every class.

A. Results Comparison

Model overall performance was assessed on a stratified test set with 14,000 samples (7,000 consistent with class). Binary classification evaluation metrics covered accuracy, precision, recall, and F1-score, with additional insights furnished by the confusion matrix. As shown in table 14 that compares the models' performances, these measures offer each popular and class-unique insights into detection capability and misclassification tendencies.

Table 14: Models Performances Comparison.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.9635	0.9636	0.9635	0.9634
Gradient Boosting	0.9619	0.9621	0.9619	0.9619

B. Discussion

The binary classification outcomes for the *Kernel_Events_1* dataset exhibit the strong capability of ensemble learning models in distinguishing between benign and attack behaviors in EVSE-associated kernel-stage telemetry. Both Random Forest and Gradient Boosting classifiers finished excessive standard overall performance, with Random Forest barely outperforming in nearly each metric: 96.35% accuracy vs. 96.19%, and marginal upgrades in precision, recall, and F1-rating.

Given the huge and balanced test set (14,000 samples), those effects are statistically reliable and mirror sturdy generalization. The consistency across metrics—with F1-rankings aligning tightly with precision and recalls shows balanced sensitivity and specificity. In practice, this means both models can detect attacks with minimal false positives and false negatives.

Random Forest's marginal side in performance can also stem from its tendency to higher capture non-linear interactions in specific-heavy characteristic spaces without overfitting. However, Gradient

Boosting's comparable performance, finished through sequential learning and optimization of residuals, confirms its viability as an equally equipped detector for real-time systems.
Importantly, no hyperparameter tuning was achieved; default configurations had been maintained to ensure model comparison. The high rankings, despite this constraint endorse that the preprocessing pipelines specifically class balancing and characteristic encoding— was powerful in revealing discriminative patterns between the training.

Each model exhibits high readiness for deployment in real-time detection pipelines for EVSE kernel-level activities. Random Forest may be slightly better for practitioners prioritizing performance stability with minimum tuning, while Gradient Boosting remains an effective alternative, mainly if future paintings involve pleasant-tuning or ensemble stacking.

5.2.2. Multiclass Classifiers

The multiclass classification task for the kernel activities dataset was designed to classify system behavior into one in all 4 distinct scenarios: Benign, Cryptojacking, DoS, and Recon. This method allows an in-depth expertise of system states beyond binary distinctions. Three machine learning models were employed for this project: Random Forest, Support Vector Machine, and Logistic Regression, all identified for their effectiveness in coping with structure data for classification. The preprocessing steps targeted balancing the dataset, encoding categorical labels, and scaling numerical capabilities to ensure equity and consistency across the models.

A. Results Comparison

Each model was evaluated on a stratified, balanced test set of 584 samples (146 per situation). Accuracy and weighted F1-rating were used to assess standard performance. Additionally, per-class precision, recall, and F1-rating offer an in-depth view of every classifier's effectiveness for individual scenarios. Table 16 compares the models' accuracies and F1-Score, while Table 16 displays the performances of each model on each class.

Table 15: Models Performances Comparison.

Model	Accuracy	Weighted F1 Score
Random Forest	0.9675	0.9674
SVM	0.9623	0.9621
Logistic Regression	0.9795	0.9795

The evaluation metrics highlight the sturdy overall performance of all three models in classifying the distinctive system scenarios. Logistic Regression had the very best accuracy (0.9795) and weighed F1-score (0.9795), indicating its effectiveness in distinguishing among the 4 classes. Random Forest and

SVM additionally tested high accuracy and F1-scores, suggesting sturdy type talents.

Table 16: Per-class Metrics.

Model	Class	Precision	Recall	F1-Score
Random Forest				
	Benign	0.97	1	0.99
	Cryptojacking	1	1	1
	DoS	0.93	0.95	0.94
	Recon	0.96	0.92	0.94
SVM	Benign	0.95	1	0.98
	Cryptojacking	1	1	1
	DoS	0.93	0.95	0.94
	Recon	0.96	0.9	0.93
Logistic Regression				
	Benign	0.97	1	0.98
	Cryptojacking	1	1	1
	DoS	1	0.93	0.96
	Recon	0.95	0.99	0.97

The per-class metrics display that all models typically carry out properly throughout all situations, with ideal or close to-best ratings for the 'Benign' and 'Cryptojacking' classes. There are minor variations in precision and recall for the 'DoS' and 'Recon' scenarios, suggesting a few subtle differences in how the models manage these greater complicated or potentially overlapping patterns. For instance, Logistic Regression shows a great precision for 'DoS' however barely lower recall as compared to Random Forest, indicating it is very correct while it predicts 'DoS'. However, it would possibly leave out some instances. Conversely, for 'Recon', Logistic Regression indicates an excessive recall, capturing maximum instances of this scenario, however with slightly decreased precision.

B. Discussion

The multiclass classification outcomes for the *Kernel_Events_1* dataset underscore the effectiveness of conventional machine learning models in identifying nuanced system behaviors across 4 different scenarios: Benign, Cryptojacking, DoS, and Recon. The models: Random Forest, SVM, and Logistic Regression, each demonstrated extraordinarily high overall performance, with accuracies exceeding 96% and closely aligned weighted F1-scores.

Among the models, Logistic Regression emerged as the top performer, accomplishing both the highest general accuracy and the best weighted F1-rating. Its energy lies in its capability to model probabilistic boundaries between multiple classes, which seems especially useful in distinguishing subtle differences between attack scenarios. This additionally shows that, whilst the dataset is well-preprocessed and capabilities are nicely scaled, even linear models can provide quite competitive

performance.

Random Forest and SVM additionally did a good job displaying sturdy generalization on the balanced and stratified test set. Random Forest confirmed constant F1-scores throughout all classes, with barely better managing of the Recon scenario compared to SVM. This is likely due to its ensemble nature, capturing nonlinear interactions and varied choice barriers. SVM, despite its theoretical capacity to model complicated boundaries for the RBF kernel, confirmed a slightly lower recall for Recon, which may additionally endorse a tighter margin boundary that didn't capture a few ambiguous samples.

At the per-class level, all 3 models got ideal rankings for Cryptojacking, and close-to-ideal results for Benign. This consistency confirms that the attack scenarios are distinguishable based totally on kernel event metrics, especially when the charging state is likewise encoded.

However, the 'DoS' and 'Recon' classes introduce minor variance between models, highlighting them as more challenging to the model, likely due to overlapping behavior in kernel-level features. Logistic Regression did perfect precision but slightly decreased recall for DoS, whilst its Recon classification had high recall (0.99) however, slightly lower precision. This trade-off indicates model-particular biases in coping with these eventualities. Logistic Regression is noticeably touchy to detect Recon, probable at the value of misclassifying some benign or DoS samples as such.

Finally, all three classifiers are powerful for multiclass situation detection, with Logistic Regression demonstrating top notch generalization and simplicity. These results strengthen the importance of cautious data balancing and preprocessing, which possibly contributed to the models' uniformly sturdy performance. Future work might also explore ensemble techniques or hybrid models to further fine-tune decision boundaries for borderline cases like DoS and Recon.

5.3. Network Traffic Detector

5.3.1. Multiclass Classifiers

This phase details the application of a GNN for the multiclass classification of network traffic events. The intention is to classify network flows into one of 15 distinct attacks sorts or a 'None' (benign) class. By representing network traffic as a graph, we can leverage the relationships among exclusive IP addresses to enhance category accuracy.

A. Results Comparison

The trained GNN model was evaluated on a stratified test set. Universal performance is evaluated using accuracy, and a detailed classification report gives precision, recall, and F1- rating for each of the 15 classes. Table 17 displays the accuracy and F1-Score of the GNN model, while table 18 compares the evaluation metrics results on each model.

Table 17: GNN Model Performance.

Metric	Value
Test Accuracy	0.9281
Weighted F1 Score	0.9285

Table 18: Per-Class Metrics.

Class	Precision	Recall	F1-Score	Support
None	0.6071	1	0.7556	17
aggressive scan	0.7333	0.963	0.8326	20504
icmp-flood	0.2273	0.8333	0.3571	12
icmp- fragmentation	0.0893	0.8333	0.1613	6
os-fingerpriting	0.8373	0.9685	0.8981	28055
TCP-port-scan	0.909	0.8822	0.8954	85332
pshack-flood	0.9999	0.9998	0.9999	39331
service- version- detection	0.9469	0.8999	0.9228	58571
slowloris-scan	0.9731	0.9893	0.9811	840
syn-flood	0.9998	0.8909	0.9422	52439
syn-stealth- scan	0.935	0.8766	0.9048	106189
synonymousIP- flood	0.903	0.9997	0.9489	52437
TCP-flood	0.9999	0.9974	0.9987	52439
upd-flood	0.9503	0.9982	0.9736	6495
vulnerability- scan	0.9165	0.9368	0.9266	46273
Accuracy			0.9281	548940
Macro Avg	0.8018	0.9379	0.8332	548940
Weighted Avg	0.9323	0.9281	0.9285	548940

The GNN model performed a sturdy universal test accuracy of 0.9281, with a weighted F1- rating of 0.9285, indicating powerful classification throughout the distinctive network traffic sorts as shown in Figure 22.

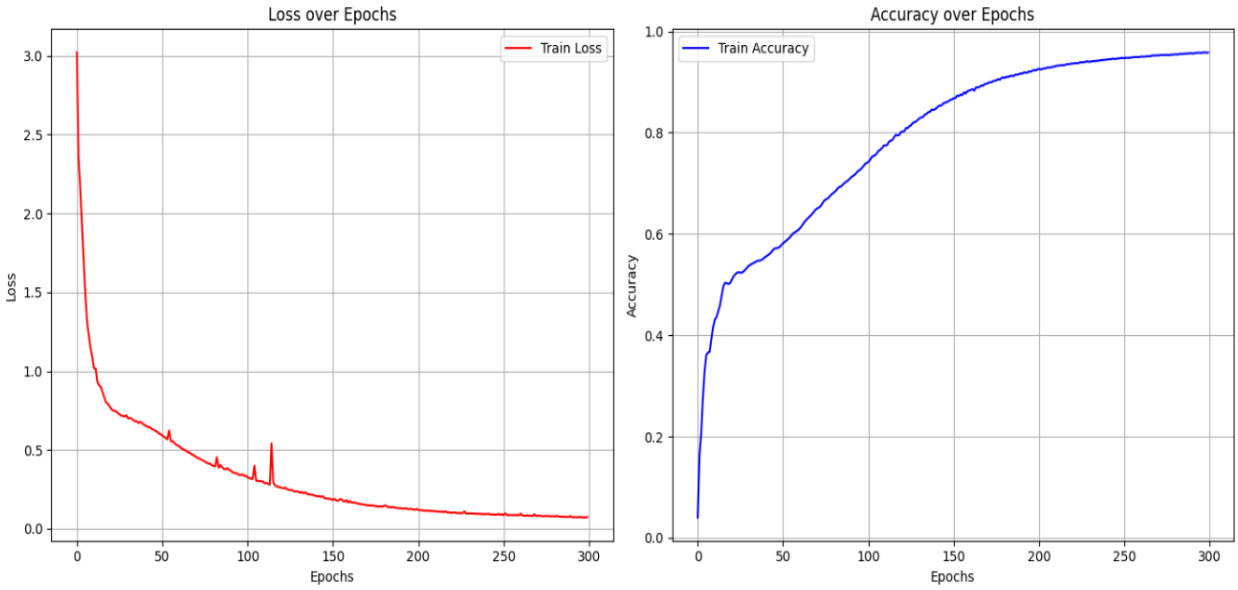

Figure 22: GNN Training Loss and Accuracy

The per-class metrics reveal significant variations in performance:

- The 'None' (benign) class, despite a perfect recall (1.0000), shows a lower precision (0.6071). This indicates that while all benign instances have been efficaciously recognized, a portion of attack traffic was misclassified as benign. The very constrained aid for this class inside the test set (17 instances) makes these metrics especially sensitive to even a few misclassifications.
- Several attack training, inclusive of 'pshack-flood', 'TCP-flood', and 'synonymousIP- flood', display close-to-ideal precision and recall, highlighting the model's high- quality ability to accurately pick out those malicious types.
- Conversely, 'icmp-flood' and 'icmp-fragmentation' show excessive recall but very low precision. This shows that while the model captures a large percentage of those attacks, it additionally generates a considerable variety of false positives. The very low guide for those classes (12 and 6 times, respectively) amplifies the impact of individual misclassifications on these metrics.
- The last attack classes, inclusive of 'aggressive-experiment', 'os-fingerpriting', 'TCP- port-test', 'provider-model-detection', 'slowloris-scan', 'syn-flood', 'syn-stealth-scan', 'upd-flood', and 'vulnerability-scan', usually exhibit proper precision and recall, contributing significantly to the excessive typical accuracy of the model.The macro common F1-score of 0.8332 affords an unweighted measure of the model's overall performance throughout all training, highlighting the challenges posed by the less frequent attack types where performance is substantially lower.

The high overall accuracy demonstrates the ability of the graph-based method for network traffic classification, successfully leveraging the relationships between network entities and the capabilities in their interactions. However, the discovered variations in per-class performance, especially the decreased precision for the 'None' class and the low precision despite high recall for a few infrequent attacks, warrant similar investigation. Analyzing the confusion matrix could provide deeper insights into the unique patterns of misclassification and guide future efforts to decorate the model's robustness and discriminatory strength across all network traffic categories.

Discussion

The consequences of the GNN-primarily based multiclass classification in the network traffic dataset show the effectiveness of graph learning techniques for detecting and distinguishing between a big selection of cyberattack types. By modeling network flows as edges among IP addresses (nodes), the model efficaciously captured each nearby feature patterns and structural dependencies inside the network traffic, achieving an outstanding test accuracy of 92.81% and a weighted F1-score of 0.9285. This high degree of overall performance highlights the ability of graph-primarily based tactics in actual international IDS.

The GNN model specifically excelled at figuring out frequent, excessive-extent attack types, consisting of pshack-flood, TCP-flood, synonymous IP-flood, and TCP-port-test, all of which achieved close to-perfect F1-rankings. These classes benefited from each high support (for example: many times, inside the dataset) and distinct feature patterns, allowing the model to generalize properly during training. For example, pshack-flood and TCP-flood got F1- rankings of 0.9999 and 0.9987, respectively, reflecting a totally low misclassification rate.

Conversely, overall performance deteriorated for uncommon attack types, maximum appreciably icmp-flood and icmp-fragmentation, both of which had fewer than 15 study samples. Despite excessive recall (for example: efficaciously figuring out maximum advantageous cases), their precision was substantially lower, with many false positives misclassified into these classes. This discrepancy is indicative of class imbalance troubles, where rare training lacks enough illustration to shape distinct, separable patterns at some point of training. In such cases, even a few misclassifications dramatically skew precision metrics.

Interestingly, the None (benign) class had a perfect recall (1.00) but especially low precision (0.6071). While this indicates that the model is a success in capturing all benign network traffic, it additionally misclassified several attacks as benign. This trade-off may be unstable in high-security applications where false negatives are costly. The very small range of benign samples showed the effect of every misclassification, similarly, emphasizing the want for statistics rebalancing or focused oversampling in future iterations.

The macro common F1-rating of 0.8332 indicates a slight variant in class-wise performance, especially for underrepresented classes. Nonetheless, the weighted F1-score of 0.9285 confirms that the model performs reliably across the dataset as an entirety, especially for the most impactful traffic flow types.

Finally, the GNN-primarily based model indicates robust potential for multiclass intrusion detection in network traffic, successfully capturing high-frequency threats even as exposing areas of development for rare or subtle attack signatures. Future paintings ought to involve addressing class imbalances via data augmentation, making use of interest- primarily based GNN layers, or leveraging fee-touchy learning techniques to strengthen model precision for underrepresented or project-crucial classes like benign traffic flows. Additionally, visual inspection of the confusion matrix and edge case analysis ought to yield sensible insights into the nature of misclassifications, guiding iterative model refinement.

Chapter 6 – Conclusion

The rapid increase in adoption of EV is putting pressure on EVSE to be more secure and resilient. EVSE is becoming more interconnected and reliant on communication protocols, thus they are more vulnerable to cyber-attacks. Taking this new threat into account, this thesis set out to investigate how ML and DL models could be used to create resilient IDS for EVSE infrastructures.

The work started with a review of the literature, where key challenges were recognized and highlighted (such as real-time detection, scalability and availability of realistic, multi-layered datasets). To address the challenges, an experimental environment known as CICEVSE2024, that simulates real-world scenarios in EV charging, was developed. Using both industrial-grade and academic-grade chargers, CICEVSE2024 was able to provide telemetry in a synchronized approach across the power level, network and kernel space of supply equipment, allowing for many varied cyber-physical attack vectors to be developed.

Then, using those datasets, the study evaluated traditional ML (Random Forest, Gradient boosting, DL approaches in binary and multi-classification problems. Results show that the traditional ML models performed reasonably well while providing visible results in ways that can be understood even for practical implementation with low resources, whereas DL models performed better when modelling temporal and complicated behaviors.

The results are promising but there are limitations to which this study must acknowledge. Under the conditions of real-time or high disparity, many models seemed to degenerate in performance equity. The issue of scalability and adaptivity in a live working environment of high-density IoT networks remains. Edge based incremental and reinforcement learning approaches are potential avenues to examine but require further exploration.

This research also identified the need for being able to detect at protocol-layer level, as well as the consideration for real-world multi-layered attacks at the network, host or physical layer. The experimental environment was tested with realistic configurations of both OCPP and ISO 15118, which provided practical applicability, beyond studies solely based on simulation alone, that could further validate the use of these methodologies.

There are many future work directions that it would be important to explore. First, large pan-industry public datasets like CICEVSE2024 at the large scale where they can be deployed at scale with many types of devices, protocol behaviors and variations of attack vectors. Second, there will be a need to move toward possible trust-aware, explainable artificial intelligence (XAI) mechanisms so operators can become more comfortable and aware of alerts generated from IDS. Lastly, with the improved system of defense, there will also be a need for adaptive and federated learning

frameworks that maintain privacy yet build on a defense mechanism as they are distributed further into an EV infrastructure.

Finally, the basic premise of the thesis has shown that the combination of classical, deep learning models based on realistic data, and evaluated from rigorous experimental methods, can positively shape the security resilience of future EVSE systems. The proposed frameworks for intelligent detection will provide scalable, adaptable responses for the evolving EV landscape which will be critical for establishing a secure smart mobility ecosystem; as EV adoption accelerates, we will need to think critically and ethically about the data involved in our everyday activities.

References

[1]: H. HaddadPajouh, A. Dehghantanha, R. M. Parizi and M. Aledhari, “A survey on Internet of Things security: Requirements, challenges, and solutions,” Internet of Things, vol. 14, 2021, Art. 100129, https://doi.org/10.1016/j.iot.2019.100129

[2]: R. Chataut, A. Phoummalayvane and R. Akl, “Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and Industry 4.0,” Sensors, vol. 23, no. 16, Art. 7194, 2023, https://doi.org/10.3390/s23167194

[3]: X. Hu, “Electric Vehicle Charging Infrastructure Security: A Survey,” Electric Power Systems Research, vol. 234, Art. 108294, 2025. DOI: 10.1016/j.epsr.2024.108294.

[4]: I. Skarga-Bandurova, I. Kotsiuba, T. Biloborodova, “Cyber Security of Electric Vehicle Charging Infrastructure: Open Issues and Recommendations,” in Proc. IEEE Big Data Conference, 2022, pp. xxx-xxx. DOI: 10.1109/BigData55660.2022.10020644.

[5]: S. Hamdare, O. Kaiwartya, M. Aljaidi, M. Jugran, Y. Cao, S. Kumar, M. Mahmud and D. Brown, “Cybersecurity Risk Analysis of Electric Vehicles Charging Stations,” Sensors, vol. 23, no. 15, art. 6716, 2023. DOI: 10.3390/s23156716.

[6]: M. Szakály, S. Köhler and I. Martinovic, “Current Affairs: A Security Measurement Study of CCS EV Charging Deployments,” in Proc. 34th USENIX Security Symposium (USENIX Security ’25), Seattle, WA, USA, Aug. 13–15 2025, 978-1-939133-52-6, pp. 7997-8009. DOI: 10.5555/3766078.3766488.

[7]: E. D. Buedi, A. A. Ghorbani, S. Dadkhah and R. L. Ferreira, “Enhancing EV Charging Station Security Using a Multi-dimensional Dataset: CICEVSE2024,” in Proc. 11th International Conference on Security of Internet of Things (SecIoT) 2024, LNCS vol. 13735, pp. 93-107, Springer, 2024. DOI: 10.1007/978-3-031-65172-4_11.

https://www.unb.ca/cic/datasets/evse-dataset-2024.html

[8]: S. Zargar, J. Joshi, and D. Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks,” IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[9]: G. Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, 2nd ed., Insecure.Org LLC, 2009.

[10]: S. Kumar, K. K. R. Choo, and A. Dehghantanha, “Cryptojacking in the Internet of Things: A New Threat to Cybersecurity,” IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8832–8845, Jun. 2021, doi: 10.1109/JIOT.2021.3059135.

[11]: A. Saad, M. Z. Khan, and S. Abdullah, “A Survey on Cryptojacking Attacks and Detection Techniques,” IEEE Access, vol. 10, pp. 60314–60332, 2022, doi: 10.1109/ACCESS.2022.3180279.

[12]: S. Hamdare, O. Kaiwartya, M. Aljaidi, M. Jugran, Y. Cao, S. Kumar, M. Mahmud, D. Brown and J. Lloret, “Cybersecurity Risk Analysis of Electric Vehicles Charging Stations,” Sensors, vol. 23, no. 15, art. 6716, Jul. 2023, doi: 10.3390/s23156716.

[13]: J. Johnson, T. Berg, B. Anderson and B. Wright, “Review of Electric Vehicle Charger Cybersecurity Vulnerabilities, Potential Impacts, and Defenses,” Energies, vol. 15, no. 11, art. 3931, 2022, doi: 10.3390/en15113931.

[14]: J. Díaz-Verdejo, J. Muñoz-Calle, A. Estepa Alonso, R. Estepa Alonso and G. Madinabeitia, “On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks,” Applied Sciences, vol. 12, no. 2, art. 852, Jan. 2022. doi: 10.3390/app12020852.

[15]: M. A. Alsoufi, S. Razak, M. M. Siraj, I. Nafea, F. A. Ghaleb, F. Saeed and M. Nasser, “Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review,” Applied Sciences, vol. 11, no. 18, art. 8383, Sep. 2021, doi: 10.3390/app11188383.

[16]: A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman and A. Alazab, “Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine,” Electronics, vol. 9, no. 1, art. 173, Jan. 2020, doi: 10.3390/electronics9010173.

[17]: A. Janwiri, EV Charging Station Attack Detection Using Machine Learning: A Comparative Study Using the CICEVSE2024 Dataset, Master’s thesis, University of Victoria, Victoria, Canada, 2024.