National College of Ireland

Enhancing Security in Electric Vehicle Charging Stations Through Advanced Anomaly Detection Systems

MSc Research Project
MSc Cybersecurity

Donnel Shinto
Student ID: X23154748

School of Computing
National College of Ireland

Supervisor: Dr Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Donnel Shinto
Student ID:	X23154748
Programme:	MSc in Cybersecurity **Year:** 2024
Module:	MSc Research Project
Supervisor:	Dr Arghir Nicolae Moldovan
Submission Due Date:	December 12, 2024
Project Title:	Enhancing Security in Electric Vehicle Charging Stations Through Advanced Anomaly Detection Systems
Word Count:	7018 **Page Count** 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Donnel Shinto
Date:	December 12, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Security in Electric Vehicle Charging Stations Through Advanced Anomaly Detection Systems

Donnel Shinto
X23154748

Abstract

In response to the increase of electric vehicles (EVs), there has been rapid deployment of electric vehicle charging stations (EVCS) in order to serve the needs of sustainable transportation. But incorporating EVCS within power grids and network infrastructures creates potentially severe cybersecurity vulnerabilities. To address these challenges, this research develops a comprehensive anomaly detection framework based on machine learning approaches for boosting the security and reliability of EVCS. The study uses power consumption, network traffic, and hosts events datasets and classifies anomalies in binary and multiclass tasks using algorithms including Random Forest and K-Nearest Neighbours (KNN). Large amounts of preprocessing and feature selection were applied on the datasets. Results demonstrate that Random Forest outperforms KNN and is the most adaptable to feature reduction. KNN performs well in binary tasks but decreases in multiclass tasks. Analysis of host events data showed near perfect accuracy, indicating that removing predictive features can improve efficiency of models. This work contributes a novel scalable anomaly detection framework, provides understanding of the performance and importance of algorithms, and presents practical applications in EVCS security. The results improve resilience of charging infrastructures and motivate further research in the anomaly detection domain for critical systems.

1 Introduction

1.1 Background

Electric vehicles (EVs), as an alternative to fuel engines, are growing rapidly. Supporting the ever increasing rate of adoption of electric vehicles is important in helping to reduce greenhouse gas emissions and fight climate change, and EV charging stations are crucial to that effort. However, with the expansion of this infrastructure the security and reliability of charging stations will have to be ensured. Although these systems are beneficial, potential vulnerabilities in the systems bring up a number of issues such as unauthorized access, energy theft, and disruption resulting in large scale collapse of power distribution networks. As such, the demand for state of the art anomaly detection solutions for EV charging stations has never been more needed. EV charging stations can experience a multitude of anomalies caused by malicious cyber-attacks, faulty components, or by unpredictable user behaviour(Root, n.d.). The existence of such anomalies can undermine its safety and reliability at the same time.

Therefore, timely detection of these anomalies is required for proper functioning and safety, and also for protecting the users from possible harms.

1.2 Objective

The objective of this research is to develop a complete anomaly detection framework for electric vehicle charging stations. The project utilises machine learning algorithms to accurately classify benign and malicious activities and improve EVCS security and reliability. The study uses Random Forest and K-Nearest Neighbours (KNN) to evaluate multiple datasets and to analyse how well these models perform in guessing anomalous events on binary and multiclass classification tasks.

To effectively detect anomalies, this research utilizes the CICEVSE2024 dataset which has three subsections: power consumption, host events, network traffic data(*EVSE Dataset 2024 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, n.d.). The datasets each give a different view of how EV charging stations are operated and them combined gives a larger threat coverage. This research uses Weka for data analysis and model development to determine which of the machine learning techniques can be used effectively to detect anomalies in various operational aspects of a charging station.

The specific research question addressed in this study is:

How can machine learning techniques be used to implement anomaly detection systems for Electric Vehicle charging stations?

In order to answer this research question, the study uses Weka Workbench to analyse power consumption, host event data and network traffic data. A GUI based tool like Weka allows us to examine pros and cons of the tool used in our study and to improve overall effectiveness of our proposed anomaly detection framework.

1.3 Contributions

- Dataset Analysis - Analysed the potential for anomaly detection with a diverse dataset including power consumption, network traffic and host events data. Datasets have been pre-processed and refined to guarantee high quality input to machine learning models.
- Algorithm Evaluation and Comparison - The performance of Random Forest and K-Nearest Neighbours (KNN) algorithms were implemented and compared on binary and multiclass classification tasks. Highlighted the strengths and weakness of each algorithm with different feature configuration.
- Insights into Feature Importance - Evaluated the effect feature reduction had on model performance and found that by keeping key attributes, major performance improvement in anomaly detection is possible.

2 Related Work

In recent years, there have been many advances in anomaly detection, most notable with respect to cybersecurity and the network system. IoT devices, electric vehicle charging station (EVCS) and rapid increase in smart grids make it more important than ever to ensure that anomaly

detection systems are operating optimally. This section presents a critical review of related works to provide an insight on the strengths, limitations, and relevance of current developments for the development of advanced anomaly detection frameworks.

According to Abdiyeva Aliyeva and Hematyar , the integration of AI in anomaly detection has transformed network security by allowing us to identify threats with a higher degree of precision. In Intrusion Detection Systems (IDS) anomaly detection is a key component. Machine learning techniques including Naive Bayes, Support Vector Machines (SVM), K-Nearest Neighbours (KNN), and Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) have played a big role in improving detection. SVM may be hybridised with Decision Tree, or with Neural Networks and Fuzzy Clustering, thus increasing accuracy and reducing false positives. As an example, the CNN-LSTM models scored 99.98% accuracy on the ISCX2012 dataset. The effectiveness of these methods is validated using evaluation metrics including precision, ROC curves, and F1 scores. But there are still difficulties such as data imbalance and setting correct anomaly thresholds. Future directions are then discussed on real-time adaptive learning and applying AI driven anomaly detection better support critical industry systems. (Abdiyeva-Aliyeva & Hematyar, 2023).

Network anomaly detection using deep learning architectures such as Fully Connected Networks (FCNs), Variational Autoencoders (VAEs), and Sequence-to-Sequence (Seq2Seq) models were studied by Malaiya et al. Furthermore, by evaluating Seq2Seq models on multiple datasets, with over 99% accuracy on all, their findings showed that these models are ideal for handling high dimensional data. Although the models were robust, their dependency on large, labelled datasets and the computationally intensive nature of the problem resulted in a barrier to real time application. In addition, while the study analysed diverse public datasets, the dataset specific optimisations proved important, however the application of optimisations to newer datasets was also uncertain. Given that, additional research is needed for applying the techniques to other datasets(Malaiya et al., 2019).

In 2021, Nassif et al. performed a systematic literature review on machine learning (ML) models for anomaly detection, over 290 papers from 2000 to 2020. Techniques were categorized as supervised, semi-supervised and unsupervised, with unsupervised being the most used because they work on the least input labelled data. Although the scope is comprehensive, the study did not evaluate the performance of these models in settings where the models are subject to real time constraints, important factor to consider when designing applications of cybersecurity. Further, though unsupervised models serve well, they also tend to produce more false positives, which in turn require the need for hybrid approaches which combine supervised and unsupervised techniques in an effort to reduce false positives and increase reliability(Nassif et al., 2021).

An anomaly detection in communication networks using ensemble learning approach based on hybrid algorithms like Adaboosting and Bagging was presented by Oleiwi et al. For datasets such as NSL-KDD and CICIDS2017 their system reached accuracies of 99.6%. Feature selection methods showed improved performance in their integration. Additionally, the approach relies on manually engineered features which makes the approach not suitable for high dimensional features or data streams that are evolving rapidly. To maintain the

effectiveness of ensemble learning against these challenges, we need to incorporate automated feature engineering techniques(Oleiwi et al., 2022).

Anomaly detection for EVSE networks was explored using classical and ensemble learning methods by Hegde et al. In their analysis, they showed that the ensemble methods could do better than classical techniques to classify attack patterns. In spite of the improvement, the practicality of the models' dependency on pre-processed features and the lack of real time evaluation, there is a need for other practical solutions. Furthermore, the study also proved quite informative about the vulnerabilities of EVSEs, but it didn't consider newer threats, such as latest attacks on charging protocols. Advanced training techniques could be incorporated in order to improve the robustness of EVSE network anomaly detection systems(Hegde et al., 2024).

For an AutoML based defender for industrial control systems, Vasan et al focused on lowering the domain expertise needed to deploy ML models. They showed that their framework yielded robust performance on various datasets including EVCS datasets achieving accuracy over 94.38%. However, transparency and interpretability, which are key factors in the security environments, were seriously affected by the reliance on AutoML. Furthermore, AutoML frameworks are computationally intensive, thereby adding to the difficulty of incorporating these into machine learning systems on resource constrained environments. Future work should also consider AutoML solutions that meet both accuracy and efficiency.(Vasan et al., 2024).

Federated learning (FL) and deep swarm particle optimization was employed by the researchers Ullah et al to secure intermittent IoVs. Their method achieved data privacy as well as improved detection performance, particularly for imbalanced datasets. Although it worked, it was computationally resource intensive and suffered from communication securities, so was never a good candidate for large scale deployments. A great direction for future research is enhancing federated learning frameworks to be able to handle diverse data(Ullah et al., 2024).

The authors Purohit & Govindarasu developed an anomaly detection system (FL-EVCS) for EVCS, based on federated learning. On average, the system achieved high accuracy 97% and preserved data privacy by sharing the model parameters not the raw data. But, federated learning brought complexities to model and necessary performance tradeoffs between privacy and detection speed. In addition, the results cannot be generalised across geographically distributed EVCS networks due to limited comprehensive testing. Finally, future studies should explore federated architectures that are capable of fitting in various infrastructure settings(Purohit & Govindarasu, 2024).

In intrusion detection systems, Kostage at el tackled the problem of data imbalance by generating synthetic data using GANs. Based on their results they found that a substantial performance improvement over traditional oversampling techniques. However, the computational cost of GAN training is very high and the possibility of learning to generate unrealistic samples restricts its wider applicability. Meanwhile, GANs can increase dataset balance but are vulnerable to collapse and hard to handle complex attack patterns, which calls for further GAN architecture research. The exploration of the combination of GANs with reinforcement learning can be useful to create more realistic synthetic data(Kostage et al., 2024).

The CICEVSE2024 dataset comprising of power consumption, network traffic, and host events data in EVSE environments was introduced by Buedi et al. The dataset is comprehensive enough for anomaly detection tasks, yet its availability for only simulated scenarios leaves questions open about it fit for use in real world deployments. Additionally, the narrow scope of the attack considered on the dataset limits the usefulness of this dataset in evaluating models against potential unknown adaptive threats. In future, it would be useful to extend the scope of this dataset by recording a wider range of attack patterns in various forms so as to better suit practical use(Buedi et al., 2024).

Paper Name	Objective	Methodology	Results	Challenges	Datasets
(Abdiyeva-Aliyeva & Hematyar, 2023)	Proposes AI based solutions to predict and detect network anomalies.	K nearest neighbours (KNN), support vector machines (SVM), and neural networks.	Talks about enhanced anomaly detection accuracy and real time evaluation.	High computation al complexity.	-NSL KDD - UNSW NB15
(Malaiya et al., 2019)	To evaluate deep learning models for network anomaly detection.	The study used FCNs, VAEs, and Seq2Seq models on public datasets.	Seq2Seq with LSTM achieved over 99% accuracy.	Handling limitations of traditional methods.	- Kyoto Honeypot -NSL KDD - UNSW NB15 - IDS2017 - Mawilab
(Oleiwi et al., 2022)	Proposes a hybrid ensemble learning (EL) system to improve anomaly detection accuracy in communicati on networks.	Random Forest ,SVM,Adaboost.	99.6%	Dependence on labeled data for training, and computation al cost of EL models.	-NSL KDD - UNSW NB15 - CIC IDS2017
(Nassif et al., 2021)	Provides a systematic review of machine learning techniques for anomaly detection.	Analyses 29 distinct ML models, comparing performanc e across supervised, semisupervised, and unsupervise d methods.	Identifies SVM and PCA as the most frequently used models .	Lack of consistent performanc e metrics and reliance on outdated datasets in many studies.	N/A
(Buedi et al., 2024)	Create a comprehensive dataset for EV charging station security	Multiple machine learning algorithms	91.3%	Lack of suitable datasets for EV charging station cybersecurity	CIC EVSE2024
(Vasan et al., 2024)	Develop an AutoML-based ICS security defender	GLM, GBM, RF, DL, XGBoost	94.38%	High computational needs, model optimization.	CIC EVSE2024
(Purohit & Govindarasu, 2024)	Federated Learning-based anomaly detection for EVCS	DNN, Federated Learning	97%	Privacy concerns, network diversity.	CIC EVSE2024

(Hegde et al., 2024)	Improve EVSE anomaly detection using classical and ensemble learning	KNN, SVM, Logistic Regression, XGBoost, AdaBoost, Random Forest	100%	Inconsistent classification in classical models.	CIC EVSE2024
(Kostage et al., 2024)	Use GANs to balance imbalanced datasets for IDS	WGAN-GP, CGAN, CTGAN, CWGAN-GP		GAN complexity and computational demands	CIC EVSE2024, CIC IoT 2023
(Ullah et al., 2024)	Develop FL-based IDS for IoVs with DSPO for feature selection	FL, DSPO, DNN with SGD	99.46%	Scaling FL, optimal feature selection	CIC EVSE2024, CIC IoT 2023

Table1: Summary table of Literature Review

3 Research Methodology

3.1 Data Collection

The data used for this research is the CICEVSE2024 dataset from the repository of University of New Brunswick. The dataset was generated using a real EV charging station and a raspberry pi module which was programmed to emulate the behaviour of an EV charging station. The dataset consists of power consumption, network traffic and host events data at both normal and malicious conditions.

Power Consumption Data - This dataset consists of the time-series data of the consumption of power at the charging station. It is used in establishing abnormal consumption patterns that might reflect the occurrence of energy theft or failure within any device.

Host Events Data - This dataset logs system-level events, such as recon, Cryptojacking, benign activities etc at EV charging stations. Host events will also be monitored for unauthorized access and other suspicious activities.

Network Traffic Data - This consists of network packets exchanged between the EV chargers and central systems. Thereafter, network traffic analysis detects possible cyber-attacks, such as DDoS or man-in-the-middle attacks.

3.2 Tools and Techniques

Weka

Weka (Waikato Environment for Knowledge Analysis) is popular and powerful open source software for machine learning and data mining(*Weka 3 - Data Mining with Open Source Machine Learning Software in Java*, n.d.). Weka is a comprehensive collection of tools for data preprocessing, classification, regression and association rule mining and visualization that has been developed by the University of Waikato. The tool is implemented in Java and has a user-friendly GUI, as well as command line and Java API functionalities. The tool allows both novice and experienced users to easily take advantage of it.

Key Features of Weka

Preprocessing Tools - Many filters for data cleaning, normalization, and transformation are contained in Weka, making it possible to pre-process the data before applying machine learning algorithms.

Machine Learning Algorithms - Weka offers a wide range of algorithms like Naive Bayes, SVM, KNN, Decision Trees etc. Without programming expertise these algorithms can be directly applied on to the datasets.

Evaluation Metrics - Robust tools for model evaluation are also included, such as cross validation, confusion matrices, precision, recall, F1 score and ROC analysis.

Visualization - It provides visualization tools to explore datasets, understand distributions and model outputs to interpret results.

Support for ARFF - Datasets are in the ARFF format (Attribute-Relation File Format) which are easy to create and edit. In addition, it supports import of data from CSV and other standard format.

Weka and Python are the two tools used to implement machine learning models using this dataset. Python offers more tools and options when it comes to machine learning. Weka on the other hand is easy to use with its simple graphical interface. Weka was utilised to build and evaluate all three parts of the dataset. Python on the other hand was primarily used to merge individual csv files of Network Traffic dataset and to clean Host Events dataset.

3.3 Data Preprocessing

Power Consumption:

Data Cleaning: Initially, the dataset was cleaned by removing duplicate entries and checking for missing values including misaligned data.

Feature Engineering: Numerical features like shunt_voltage, bus_voltage_V, current_mA, power_mW were normalised to have them on a similar scale.

Network Traffic:

Merging of CSV Files – The network traffic files were separated as charging, idle and malicious. These set of files were merged to form a single CSV file. The original files each had 86 columns. The merged dataset has 88 columns the two additional attributes being label and class group which was added using python

Removal of Columns – InfoGainAttributeEval function of Weka was used to calculate the information gain of each column in the dataset. All columns with a gain below 0.1 was removed. Information Gain was calculated for remaining columns and evaluation were done on features with gains greater than 0.2, 0.5 and 1.

Feature Scaling – The dataset features were standardised to get more accurate results from the machine learning model.

Host Events

Dataset Cleaning – The dataset was cleaned by removing duplicates and adding missing values. Some rows were corrupted where values were misplaced into different columns. These irregularities were fixed using python.

3.4 Data Transformation

Transformation techniques tailored for specific dataset characteristics were made to prepare datasets for the analysis. In the case of Power Consumption, we measured how feature removal affected the classification performance. We then used the transformed datasets for anomaly detection in multiple classification tasks. For the Network Traffic, the features having values greater than 0.2, 0.5 and 1.0 were chosen based on the information gain threshold. It enabled us to train the model by only relying on the most informative attributes, to optimize the performance and reduce complexity. Host Events feature selection was conducted by removing columns that were highly correlated to the classes. This helped decreasing risk of overfitting. We split the datasets into training (70%) and testing (30%) subsamples in order to test our model.

3.5 Data Mining and Model Development

Two machine learning algorithms Random Forest and K nearest neighbours (KNN) were employed for mining patterns and classifying resulted anomalies. Because of its robustness and its ability to deal with high dimensional data, Random Forest was chosen. KNN was chosen for its simplicity and effectiveness in local pattern recognition, especially for binary tasks. All datasets were explored by applying these algorithms across binary and multiclass classification tasks with configurations adjusted with respect to the specific dataset(*What Is the Difference between the Three Machine Learning Models?*, n.d.).

Class Configurations for Power Consumption Dataset
Label – Evaluated the efficiency to distinguish between benign/attack.
Attack – To classify different types of attacks.
Attack Groups – Broader classification of attacks into groups.

Class Configurations for Network Traffic Dataset
Two classifications were used: Label and Class Group.
Models were evaluated for varying feature selection thresholds.

Class Configurations for Network Traffic Dataset
Label – To find the binary classification between attack/benign
Attack – To classify different types of attack.
Scenario – To group attacks into different classes.

3.6 Evaluation

Performance Metrics:
Accuracy: It refers to the percentage of correctly classified instance.
Precision: True positives to predicted positive.
Recall: Number of true positives out of actual positives.
F1 Score: Precision and recall harmonic mean.
ROC Area: It evaluates the trade-offs between sensitivity and specificities.

Train-Test Split:70% of each dataset was used for training and 30% for testing. Cross validation was not performed to focus on static evaluation.

Comparative Analysis:
The power consumption dataset was grouped by target class (Label, Attack, and Attack Group), where the classification performance was evaluated over configurations with different levels of feature removal. Similarly host events data was also grouped based on target class. The performance over Information Gain thresholds was compared for the network traffic dataset.

4 Design Specification

The methodology integrates preprocessing, model training and model evaluation to the system. Host Events data was cleaned using python. Python was also used to combine multiple CSV files of the network traffic dataset into one analysable file. Each dataset was further subjected to Weka to perform feature scaling, normalisation and attributes selection. Evaluation measure such as Precision, Accuracy, Recall and F1Score were used to measure the performance consistently across the datasets.

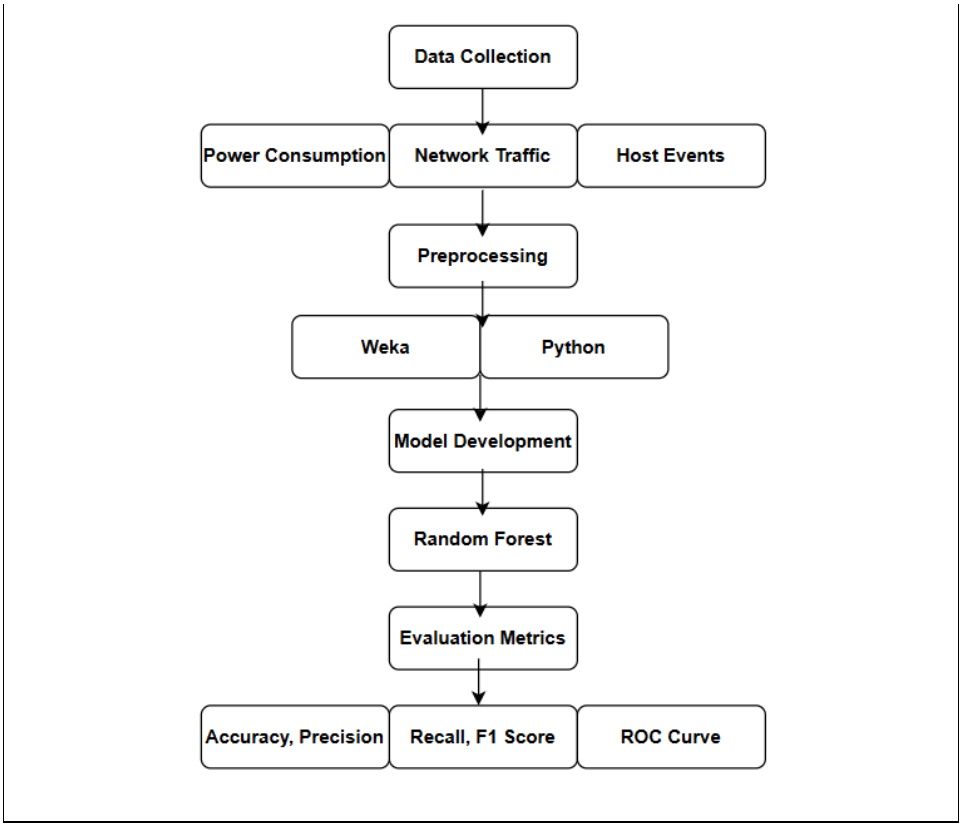

Figure1: Workflow Diagram

5 Implementation

5.1 Network Traffic

The network traffic dataset used for anomaly detection had 88 features. The result of merging and preprocessing using python introduced two additional columns which are label and class group. Then Weka workbench was used to build machine learning model and analyse anomalies. Due to the large size of the dataset Weka's memory had to be increased 8GB to avoid runtime problems. This upgrade allowed to execute feature selection and model training smoothly.

Feature Selection

Feature Selection was performed based on the Information Gain (IG) which is a widely used method for calculating the relevance of each feature from the target dataset(*What Is Information Gain*, n.d.). Thresholds of IG greater than 0.2, 0.5 and 1.0 were used to filter features. This feature selection resulted in the building of six distinct models.

- Label with IG > 0.2.
- Label with IG > 0.5
- Label with IG > 1.0
- Class Group with IG > 0.2
- Class Group with IG > 0.5
- Class Group with IG > 1.0

Label represents the different type of network traffic to the EV charging stations. The Class group categorises these labels into charging, idle and malicious.

These thresholds were chosen so that we could systematically understand the effect of the model's classification performance in the presence of different degrees of feature importance. The lower thresholds included a greater number of features, offering better input for the models, but high thresholds sacrificed the least significant features for purposes of computational efficiency and the elimination of noise in the dataset.

All models were trained with the Random Forest algorithm in Weka as it is robust and efficient when dealing with large datasets with high dimensional features. Because of its capacity to manage complex interactions among features, as well as its robustness to overfitting, Random Forest was chosen for this task. The following settings were employed for Random Forest:

Number of Trees (I): 100

Minimum Instances per Leaf (M): 1

Seed (S): 1 for reproducibility

Test Mode: Sampled into 70% training and 30% testing data.

The dataset was split into 70% train and 30% test. It was split so that models would be trained on a wide range of data and would have enough remaining to test performance. Accuracy, precision, recall, F1 score, and confusion matrices were defined as models' evaluation metrics to evaluate models effectiveness in classifying anomaly.

5.2 Power Consumption

Features like shunt voltage, bus voltage, current, and power measurements were features in the power consumption dataset used for anomaly detection in electric vehicle charging stations.

The time column was removed in preprocessing as it did not help in building the model. The dataset was categorized into three target classes: Label, Attack and Attack-Group. These classes were evaluated by systematically removing certain features from the dataset. Four models were implemented for each target class, resulting in a total of 12 models.

Feature Selection and Preprocessing

Initial Features: The original dataset was a time series dataset with voltage, current and power information. Removing time column, as it did not bring significant improvement to the classification tasks.

Classes: The dataset was divided into the three classes.

- Label: Indicates whether status is benign or attack.
- Attack: Specifies the type of attack.
- Attack Group: Broadly categorizes attacks into groups.

Evaluations were performed for each class by iteratively removing different combinations of columns like State, Interface, and other class specific features. The intention of this approach was to find the effect of feature deletion on model performance. Removing these features systematically allowed us to gain some understanding of which features are most important for anomaly detection.

Two machine learning algorithms KNN and Random Forest were used to analyse power consumption dataset. These algorithms were applied to three classification tasks, Label, Attack and Attack Group, to detect and classify anomalies in electric vehicle charging stations. We also removed time column and columns which could act as predictors for different classes to avoid over fitting. For example, in Label classification task, we didn't use attack type, attack group as they may act as predictors.

Random Forest was selected for its robustness and due to its ability to work with high dimensional data. Simplicity and effectiveness of KNN to capture local patterns were exploited for the performance comparison with Random Forest. These models were trained on 70% of the dataset, then evaluated on the remaining 30% by metrics like accuracy, precision, recall, F1 score and ROC area. KNN brought an extra dimension and can be especially helpful in cases where Random Forest performance drops significantly.

5.3 Host Events

Random Forest models were used to find anomalies in host events. The dataset contained features representing kernel events, with classifications performed across three distinct target classes: Attack, Attack Scenario Types, and Binary classification. The columns for each classification task had to be systematically removed to prevent the models from overfitting as the data corrupted the models. The columns Attack and Scenario were excluded for the Label class, as they could serve as predictors. In a similar fashion Label and Scenario columns were dropped for classifying Attack Types. For the Attack Scenario Types, the Attack and label columns were removed. The idea behind this preprocessing strategy was to ensure that models are not overfit on remaining features to ensure increased accuracy.

We chose Random Forest as the main algorithm because of its good performance and robustness with high dimensional data. We split the dataset into 70% training and 30% for testing to assess model performance. For the Random Forest classifier 100 trees were used, minimum one instance per leaf and seed value 1 for reproducibility was used. The models showed consistent performance even when possibly predictive features were removed.

6 Evaluation

6.1 Network Traffic

Setting higher IG thresholds lowered the number of features significantly which made the model training and testing faster without loss of accuracy. The importance of good feature selection is thus emphasized for improving computational efficiency at a minimal loss of predictive performance. The highest classification performance was achieved by all the models with IG > 1.0, which means that a relatively small number of highly relevant features are often sufficient for accurate anomaly detection.

Configuration	Features Selected	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F1 Score
Class Group with IG > 0.2	31	99.9977%	0.0023%	0.9999	0.9999	0.9999
Class Group with IG > 0.5	20	99.9977%	0.0023%	0.9999	0.9999	0.9999
Class Group with IG > 1.0	9	99.9982%	0.0018%	1.0000	1.0000	1.0000
Label with IG > 0.2	31	99.9943%	0.0057%	0.9998	0.9998	0.9998
Label with IG > 0.5	20	99.9956%	0.0044%	0.9999	0.9999	0.9999
Label with IG > 1.0	9	99.9999%	0.0001%	1.0000	1.0000	1.0000

Table 2: Results of Network Traffic Data

The network traffic dataset was evaluated using feature selection thresholds based on information gain (IG) for two classification tasks: Class Group and Label. Results show excellent performance across all configurations, with all models attaining near perfect classification accuracy and classification metrics. In case of Class Group classification, the IG > 0.2 and IG > 0.5 configurations kept 31 and 20 features, respectively, and obtained the same accuracy of 99.9977% correctly classified instances and a precision, recall and F1 Score of 0.9999. IG > 1.0 configuration that retained only 9 features performed slightly better, obtaining 99.9982% correctly classified instances and perfect precision, recall, and F1 score of 1.0000. This shows that the reduced feature set retained the most important attributes with as much as less noise as possible while still achieving high classification accuracy.

For the Label classification, in the case of the IG > 0.2 configuration, we obtained 99.9943 % correctly classified instances and a precision, recall and F1 scores of 0.9998. IG > 0.5 with 20 features yielded 99.9956% accuracy and IG> 1.0 with 9 features resulted in 99.9999% correctly classified instances with perfect precision, recall and F1 score. Dimensionality reduction with information gain is shown effective to optimize model performance and reduce computational complexity.

6.2 Power Consumption

Label Classification Results

Algorithm	Configuration	Correctly Classified (%)	Incorrectly Classified (%)	Precision	Recall	F1 Score	ROC Area
Random Forest	Label (All Columns)	100	0	1.000	1.000	1.000	1.000
Random Forest	Label (Attack, Attack Group Removed)	100	0	1.000	1.000	1.000	1.000
Random Forest	Label (Interface, Attack, Attack Group Removed)	94.9464	5.0536	0.948	0.949	0.948	0.969
Random Forest	Label (State, Interface, Attack, Attack Group Removed)	94.0443	5.9557	0.938	0.940	0.939	0.960
KNN	Label (All Columns)	100	0	1.000	1.000	1.000	1.000
KNN	Label (Attack, Attack Group Removed)	100	0	1.000	1.000	1.000	1.000
KNN	Label (Interface, Attack, Attack Group Removed)	93.7813	6.2187	0.936	0.938	0.936	0.933
KNN	Label (State, Interface, Attack, Attack Group Removed)	92.8937	7.1063	0.926	0.929	0.927	0.920

Table 3: Power Consumption binary classification results

Attack Classification Results

Algorithm	Configuration	Correctly Classified (%)	Incorrectly Classified (%)	Precision	Recall	F1 Score	ROC Area
Random Forest	Attack (All Columns)	91.8009	8.1991	0.897	0.918	0.904	0.983
Random Forest	Attack (Label, Attack Group Removed)	72.8151	27.1849	0.714	0.728	0.719	0.917
Random Forest	Attack (Interface, Label, Attack Group Removed)	63.4132	36.5868	0.619	0.634	0.624	0.893
Random Forest	Attack (State, Interface, Label, Attack Group Removed)	60.181	39.819	0.585	0.602	0.591	0.885
KNN	Attack (All Columns)	91.1648	8.8352	0.894	0.912	0.900	0.976
KNN	Attack (Label, Attack Group Removed)	72.2571	27.7429	0.709	0.723	0.713	0.897

KNN	Attack (Interface, Label, Attack Group Removed)	59.9988	40.0012	0.590	0.600	0.592	0.857
KNN	Attack (State, Interface, Label, Attack Group Removed)	53.8235	46.1765	0.528	0.538	0.531	0.839

Table 4: Power Consumption attack classification results

Attack Group Classification Results

Algorithm	Configuration	Correctly Classified (%)	Incorrectly Classified (%)	Precision	Recall	F1 Score	ROC Area
Random Forest	Attack Group (All Columns)	100	0	1.000	1.000	1.000	1.000
Random Forest	Attack Group (Attack, Label Removed)	79.0165	20.9835	0.795	0.790	0.788	0.925
Random Forest	Attack Group (Interface, Attack, Label Removed)	69.2272	30.7728	0.696	0.692	0.691	0.894
Random Forest	Attack Group (State, Interface, Attack, Label Removed)	65.8475	34.1525	0.656	0.658	0.653	0.872
KNN	Attack Group (All Columns)	100	0	1.000	1.000	1.000	1.000
KNN	Attack Group (Attack, Label Removed)	78.4238	21.5762	0.787	0.784	0.783	0.914
KNN	Attack Group (Interface, Attack, Label Removed)	65.7492	34.2508	0.659	0.657	0.657	0.858
KNN	Attack Group (State, Interface, Attack, Label Removed)	59.2905	40.7095	0.589	0.593	0.590	0.813

Table 5: Power Consumption attack group classification results

Comparison between Random Forest and KNN provides some important insights into the performance of these algorithms when set up under different configurations. For tasks that have a binary classification (Label class), both algorithms performed well obtaining perfect accuracy and perfect F1 scores when all features are retained. By this result, we can see how the binary classification is simple as opposed to a more difficult multiple classification problem where the model has to discriminate only between benign and attack events.

Random Forest was able to maintain a higher performance level as features were removed when compared to KNN. For example, with the Label class with several columns eliminated, Random Forest attained an accuracy rate of 94.04% and KNN got 92.89%. The Random Forest is better than KNN and has the capability to capture the complex feature interaction from combining multiple decision trees. On the other hand, KNN is a distance dependent algorithm

which, as the feature space is reduced and datapoints become less distinguishable, decreases in effectiveness.

There were extreme differences in performance between Random Forest and KNN for multiclass tasks like Attack and Attack Group. For all features, Random Forest performed best in the Attack classification getting 91.80% accuracy, and KNN got 91.16%. But as features were removed the performance of KNN decreased drastically compared to Random Forest. For example, Random Forest managed to maintain accuracy of 60.18% in the configuration when state, interface, attack group and label features were removed, while KNN dropped to 53.82%. The decrease demonstrates that KNN relies on a larger feature set. These differences are further shown by the Attack Group classification. With all features retained, Random Forest and KNN got a perfect accuracy of 100%. However, as we removed features Random Forest performed better. When state, interface, attack and label columns are removed from the configuration, Random Forest achieves accuracy of 65.84% and KNN reduces to 59.29%.

Random Forest is shown to be more stable algorithm for this dataset in the multiclass tasks and smaller feature configuration. The KNN is good for data having lot of features but turns out to be bad on most complex tasks. These results underscore the importance of choosing the right algorithm depending on the nature of the problem as well as the nature of the dataset.

6.3 Host Events

The results of three classification, namely, Label, Attack Types, and Attack Scenario Types are presented below. Near perfect performance was achieved by the Label (Binary) classification with an accuracy of 99.67% and Kappa statistic of 0.9931. Similarly, precision and F1 scores were high, confirming the good performance of differentiating between benign and attack events. Attack Types were 87.51%accurate with precision and recall fluctuating for different attack types. Cryptojacking turned out to be a perfect class, and while ICMP fragmentation's performance was lower, it could still be improved. In Attack Scenario Types classification, the algorithm achieved an accuracy of 98.37% and a Kappa statistic of 0.9774, with only minimal misclassification across the scenarios.

Classification	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F1 Score	ROC Curve
Label (Binary)	99.6757	0.3243	0.997	0.997	0.997	1.000
Attack	87.5135	12.4865	0.883	0.875	0.876	0.994
Scenario	98.3784	1.6216	0.984	0.984	0.984	1.000

Table 6: Host Events classification results

This table provides the performance metrics for each of the classifications. For Label, a binary classification task, the accuracy and consistency among metrics were highest. An accuracy of 99.67% and a Kappa statistic of 0.9931 shows the reliability of the binary classification model for distinguishing benign and attack events. In other words, the features that were not related with the class labels were also sufficient to extract the key patterns and

make the accurate classification. The precision and recall further serve to show that benign and attack classes were identified without high error and hence a very high F1 score of 0.997. On the other hand, the Attack Types and Attack Scenario Types multiclass classification tasks showed slightly worse results, especially for Attack Types classification. Results indicate accuracy of 87.51%, and a Kappa statistic of 0.8374, demonstrating the increased challenge in differentiating among several attack types. The precision values were also varied across classes and we have obtained perfect precision and recall for cryptojacking attack whereas other attack types like ICMP fragmentation had a lower precision and recall. However, this indicates that it is harder to detect some attack types, maybe because of data imbalance or lack of distinct features. We achieve an accuracy of 98.37% and Kappa statistic of 0.9774 for the Attack Scenario Types classification. The precision and F1 scores were also consistently high for all the scenarios. From the results, it appears the model can accurately portray the overall picture of attack scenarios.

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   <-- classified as
 516   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   a = cryptojacking
   0 701   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   b = none
   0   0  42   0   0   2   1   0   0   0   3   0   0   3   0   4   0  11 |   c = aggressive-scan
   0   0   0  31  13   0   0   4   0   1   1   0   0   1   0   0   1   0 |   d = icmp-flood
   0   0   0   3  10   0   0   1   0   1   0   0   0   0   0   0   0   0 |   e = icmp-fragmentation_old
   0   0   5   0   0  16   1   0   0   0   4   0   0   3   0   1   0   0 |   f = os-fingerprinting
   0   1   1   0   0  11  29   0   0   0   5   0   0   1   0   6   0   0 |   g = port-scan
   0   0   0   2   2   0   0  40   0   3   0   1   0   0   2   0   1   0 |   h = push-ack-flood
   0   0   0   0   0   2   0   0  18   0   0   0   0   1   0   0   0   0 |   i = serice-detection
   0   0   0   1   2   0   0   0   0  29   0   0   0   0   0   0   0   0 |   j = syn-flood
   0   4   1   0   0   1   9   0   1   0  27   0   0   2   0   5   0   2 |   k = syn-stealth
   0   1   1   1   0   0   0   0   0   1   1  28   0   0   1   0   9   0 |   l = tcp-flood
   0   3   1   0   0   0   0   0   0   0   0   0  24   0   2   0   2   0 |   m = udp-flood
   0   2   3   0   0   5   2   0   1   0   5   0   0  24   0  14   0   5 |   n = vuln-scan
   0   1   0   0   0   0   0   0   0   0   0   0   0   0  30   0   0   0 |   o = icmp-fragmentation
   0   1   3   0   0   3   1   0   0   0   4   0   0   7   0  17   0   5 |   p = service-detection
   0   0   0   0   2   0   0   0   0   0   0   7   1   0   3   0  25   0 |   q = synonymous-ip-flood
   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0  12 |   r = os-scan
```

Figure 2: Confusion metrics for Attack Types

The table shows that overall, the binary classification tasks are easier and attain better performance metrics than the multiclass tasks. We could observe that the accuracy, precision, and recall of those multiclass classifications decrease especially with regard to accuracy. However, the performance across all models has been consistently good, verifying the preprocessing and strategies used in this study.

6.4 Discussions

An important strength of this framework is that it achieves consistent classification performance even after feature reduction. For example, in the network traffic dataset, by eliminating features with a pre specified information gain thresholds (IG > 1.0) there was minimal loss of accuracy and a reduction in computational complexity. Random Forest exhibits a consistent performance across binary as well as multiclass tasks further enhancing its robustness. The same was true for the host events dataset. The removal of predictive classes resulted in better model performance. Another strength of the methodology is that it can be scaled. Feature selection techniques, including information gain for network traffic, and feature

removal for host events, show the flexibility of the framework to different datasets. As a result, the framework can be extended to other critical infrastructure like smart grid or IoT systems.

Despite the strength the process of evaluation also highlighted some limitations. K-Nearest Neighbours (KNN) performed well in cases of binary tasks, but performance decreased for multi class especially when the number of features was decreased. Further, Random Forest showed superior performance, however its computational requirements especially in large datasets may be problematic for real time deployment. Another problem is to address the issues of class imbalance in the datasets. In some configurations, minority classes displayed a worse recall as well as precision. Therefore, we need to apply advanced techniques like oversampling or ensemble methods in order to solve the class imbalance problem as it may seriously hurt the performance of the model.

6.5 Comparison With Past Papers

Study	Dataset Used	Algorithms	Classification Type	Performance Metrics
This Study	Power Consumption	Random Forest KNN	Binary Multiclass	Binary; Accuracy – 100% ROC Curve – 1.000 Multiclass; Accuracy – 91.800% ROC Curve – 0.983
(Buedi et al., 2024)	Power Consumption (HPC)	KNN Random Forest	Binary Multiclass	Binary; Accuracy – 91.3% Multiclass; Accuracy – 78.8%
(Buedi et al., 2024)	Host Events	Adaboost XGBoost SVM	Binary Multiclass	Binary; Accuracy – 91.88% Multiclass; Accuracy – 71.9%

Table 7: Comparison with past papers

7 Conclusion and Future Work

This work developed a comprehensive anomaly detection framework using multiple datasets and machine learning algorithms to enhance the EVCS security. Analysis was done on power consumption, network traffic and host events data to discover its anomaly detection capability using random forest and K-Nearest Neighbours (KNN) algorithms on binary and multiclass classification tasks. This work demonstrated that Random Forest performed better than KNN and is also robust in an environment with much smaller feature set. KNN worked well for binary classification, but for multiclass the performance decreased especially when number of features were reduced. Furthermore, an analysis of host events data reinforces the importance of efficient automated feature removal to avoid overfitting and guarantee consistent model performance. Overall, the study demonstrates that issues in an EVCS are most effectively detected by preprocessing, feature engineering, and the choice of algorithm. The analysis over

datasets and algorithms offers a solid basis to increase the resilience and security of charging infrastructures.

Future works based on the result of this research should build upon the development of more advanced methodology for the improvement of anomaly detection in EVCS. Improving feature engineering through techniques like principal component analysis (PCA), or autoencoders may increase efficiency of the system. Other possibilities for improving this model include deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We can further improve by adding new attributes such as user logs and other such details. Adding Random Forest to deep learning might be a hybrid approach to improving the model to handle more complex data. In addition, the proposed framework can be extended to other relevant critical infrastructure such as a smart grid or to an IoT ecosystem.

References

Abdiyeva-Aliyeva, G., & Hematyar, M. (2023). AI-Based Network Security Anomaly Prediction and Detection in Future Network. In D. J. Hemanth, T. Yigit, U. Kose, & U. Guvenc (Eds.), *4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering* (pp. 149–159). Springer International Publishing. https://doi.org/10.1007/978-3-031-31956-3_13

Buedi, E. D., Ghorbani, A. A., Dadkhah, S., & Ferreira, R. L. (2024). *Enhancing EV Charging Station Security Using A Multi-dimensional Dataset: CICEVSE2024*. Research Square. https://doi.org/10.21203/rs.3.rs-4046330/v1

EVSE Dataset 2024 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved December 11, 2024, from https://www.unb.ca/cic/datasets/evse-dataset-2024.html

Hegde, S. R., V., V., Chandrakala, K. R. M. V., T., G. K., & Shankar, V. K. A. (2024). Enhancing Anomaly Detection in Electric Vehicle Supply Equipment (EVSE) Networks Using Classical and Ensemble Learning Approaches. *2024 Control Instrumentation System Conference (CISCON)*, 1–5. https://doi.org/10.1109/CISCON62171.2024.10696830

Kostage, K., West, D., Meinert, T., Qu, C., Calyam, P., & Mazzola, L. (2024). Enhancing Autonomous Intrusion Detection System with Generative Adversarial Networks. *2024 IEEE 20th International Conference on E-Science (e-Science)*, 1–10. https://doi.org/10.1109/e-Science62913.2024.10678662

Malaiya, R. K., Kwon, D., Suh, S. C., Kim, H., Kim, I., & Kim, J. (2019). An Empirical Evaluation of Deep Learning for Network Anomaly Detection. *IEEE Access*, *7*, 140806–140817. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2943249

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access*, *9*, 78658–78700. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3083060

Oleiwi, H. W., Mhawi, D. N., & Al-Raweshidy, H. (2022). MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks. *IEEE Access*, *10*, 91006–91017. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3201869

Purohit, S., & Govindarasu, M. (2024). FL-EVCS: Federated Learning based Anomaly Detection for EV Charging Ecosystem. *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*, 1–9. https://doi.org/10.1109/ICCCN61486.2024.10637543

Root, T. (n.d.). EV Charger Hacking Poses a 'Catastrophic' Risk. *Wired*. Retrieved December 11, 2024, from https://www.wired.com/story/electric-vehicle-charging-station-hacks/

Ullah, F., Srivastava, G., Mostarda, L., & Cacciagrano, D. (2024). Collaborative Intrusion Detection System for Intermittent 10 Vs Using Federated Learning and Deep Swarm Particle Optimization. *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–8. https://doi.org/10.1109/DSAA61799.2024.10722781

Vasan, D., Alqahtani, E. J. S., Hammoudeh, M., & Ahmed, A. F. (2024). An AutoML-based security defender for industrial control systems. *International Journal of Critical Infrastructure Protection*, *47*, 100718. https://doi.org/10.1016/j.ijcip.2024.100718

Weka 3—Data Mining with Open Source Machine Learning Software in Java. (n.d.). Retrieved December 11, 2024, from https://ml.cms.waikato.ac.nz/weka/

What is Information Gain. (n.d.). Retrieved December 11, 2024, from https://www.activeloop.ai/resources/glossary/information-gain/

What is the difference between the three Machine Learning models? (n.d.). ResearchGate. Retrieved December 11, 2024, from https://www.researchgate.net/post/What_is_the_difference_between_the_three_Machine_Learning_models