ORIGINAL ARTICLE

Fortifying EV charging stations: AI-powered detection and mitigation of DDoS attacks using personalized Federated learning

Fatma M. Talaat[1,2] · Mohamed Mohsen Elsaid Khoudier[2] · Ibrahim F. Moawad[3,4] · Amir El-Ghamry[2,5,6]

Received: 24 August 2024 / Accepted: 10 June 2025

Abstract
Electric vehicle charging stations (EVCS) are becoming more and more common, so it is imperative to protect these systems from cyberattacks, especially Distributed Denial-of-Service (DDoS) assaults. The objective of this study is to enhance the model interpretability and detection accuracy of DDoS attacks in EVCS using the Personalized Federated Learning (PFL) technique. The research makes use of an IoT attack dataset with 33 attacks that were carried out over 105 devices in a topology that was divided into seven different categories. Using the Firefly Algorithm, the suggested PFL method selects a subset of features wisely to maximize the performance of the classification model. Promising outcomes are seen in the evaluation of several machine learning models, such as Random Forest, Gradient Boosting Machine (GBM), K-Nearest Neighbors, and Multilayer Perceptron. GBM and Random Forest demonstrate their promise for efficient DDoS detection in EVCS by achieving high accuracy rates of 99% and 98%, respectively, in detecting DDoS attacks. The overall detection performance is further improved by the feature selection model, which also increases the efficiency and interpretability of the classification model. These results imply that machine learning models can improve the security and resilience of EVCS against DDoS attacks when combined with the PFL technique.

Keywords Cybersecurity · DDoS attack · Electric vehicle (EV) · IoT security

1 Introduction

In the realm of modern urban development, systems for energy management and transportation enhanced by artificial intelligence are becoming crucial for the advancement of substantial urban infrastructure. As a result, it is anticipated that Electric Vehicles (EVs) will increase integration into both private and public transportation networks shortly. To this end, various governments have embarked on a range of initiatives aimed at fostering the adoption of EVs, with a particular emphasis on their contribution toward fulfilling diverse objectives related to green transportation policies in the future [1].

The utilization of EVs is known to offer significant environmental benefits, including the improvement in air quality, the reduction in noise pollution, and the diminishment of carbon emissions through the mitigation of road traffic pollution. In pursuit of the objective to reduce carbon emissions encapsulated by the Accelerating to Zero (A2Z) agenda, numerous governments worldwide have implemented strategies to curtail the dependence on vehicles powered by fossil fuels. For example, the UK government has pledged to facilitate the introduction of

https://doi.org/10.1007/s00521-025-11452-7
Published online: 30 July 2025

new zero-emission cars and vans, supporting its commitment to achieving the "Road to Zero greenhouse gas emission" target by the year 2050 [2].

The charging infrastructure for electric vehicles (EVs) constitutes a critical component of the smart grid ecosystem, consisting of a complex cyber-physical system that amalgamates diverse hardware components, software frameworks, and communication protocols. This comprehensive system enables the efficient conveyance of electrical energy from the grid to EVs via Electric Vehicle Charging Stations (EVCS). These stations are equipped with Internet of Things (IoT) technology and function based on specialized firmware, facilitating autonomous operations.

Similar to other internet-connected devices, the charging systems for electric vehicles (EVs) are vulnerable to cyber threats. These threats could originate from various components such as the charging system's hardware and software, the applications for locating charging stations, payment mechanisms, and wireless communication networks. Cybercriminals may exploit these vulnerabilities to execute Distributed Denial-of-Service (DDoS), ransomware, and Trojan horse attacks by leveraging insecure communication channels and software weaknesses. For example, Kaspersky Lab identified security vulnerabilities in the smart application designed for home EV charging [3].

An attacker could potentially interrupt the EV charging process by accessing the charging system through its Wi-Fi connection, (i.e., launch DoS attacks). Similarly, security flaws were discovered in Schneider EV link chargers, which could enable remote attackers to bypass hardcoded passwords, implant malware, and disable the charger exploiting these weaknesses. Electric vehicles periodically release firmware updates to patch security vulnerabilities. However, if a vehicle that has been compromised connects to a charging station, it risks spreading malicious software to other vehicles and potentially to the entire electricity grid.

Charging stations act as a critical junction between electric vehicles and the power grid, making them an attractive target for those seeking to disrupt the grid. The data generated about the locations and times of charging, along with the electricity consumption details, are invaluable for anyone looking to manipulate charging station demand. This information, often transmitted wirelessly via third-party applications, is prone to tampering. Information security is becoming increasingly essential in today's rapidly evolving world, characterized by a growing number of internet-connected devices and a rapid expansion of online applications [4]. Since the inception of the World Wide Web, a staggering 1.2 billion websites have been created. Additionally, a vast array of online applications has been integrated with various web services, including e-commerce, online banking, online shopping, online education, e-healthcare, and industrial control systems (ICS) for critical infrastructure [4, 5].

Due to the extensive utilization of the mentioned services, they are more susceptible to cyber-attacks, particularly DDoS attacks, which are both highly dangerous and commonly occurring. These attacks can disrupt numerous services [6]. It happens when a system transmits harmful communication to a server. A DDoS attack occurs when a substantial quantity of computers or compromised systems, known as bots, carry out DoS attacks on a single application. The specific network is subsequently inundated with packets originating from various locations worldwide. DDoS attacks are evolving and increasing in terms of both size and intricacy as disruptive Internet technologies become more widespread [7]. Possible cyber threats such as outages, data theft, and ransom demands from attackers can significantly impact an organization's operations. Figure 1 shows the environment of DDoS attacks.

In February 2020, Amazon experienced a significant DDoS attack known as a CLDAP reflection–amplification attack. This attack reached an unprecedented rate of 2.3 Tbps, making it the largest recorded attack to date, according to ZDNet. In February 2018, GitHub also encountered a similar attack that resulted in a brief interruption of their services. Dyn's managed DNS infrastructure was subjected to a targeted attack in 2016, which lasted around 3 h and had a significant impact on various prominent web services such as Twitter and PayPal, resulting in severe disruption. In general, DDoS attacks present a substantial obstacle and necessitate a thorough strategy to reduce and control the related hazards [6].

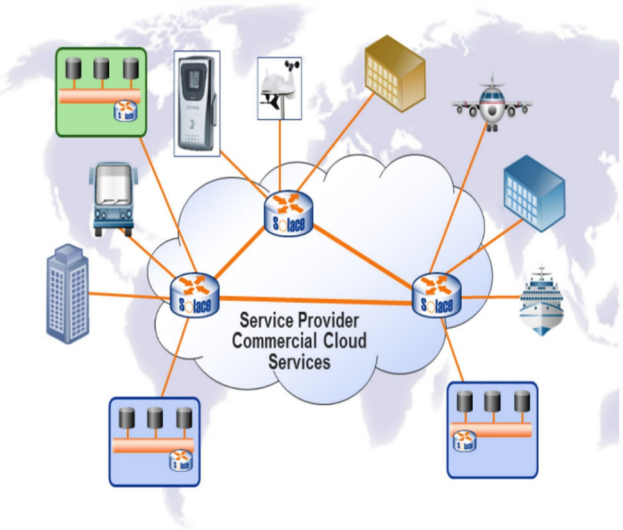


Fig. 1 Environment of DDoS attacks [7]

Machine learning (ML) is increasingly being applied to enhance the detection of security threats, including the identification of DDoS attacks targeting EVCS. By analyzing large volumes of data derived from network traffic, ML algorithms can sift through numerous data characteristics to extract meaningful patterns and indicators of potential threats. This ability to process and learn from diverse datasets allows ML to offer significant insights into security vulnerabilities and attack vectors [8–10]. Furthermore, when integrated with the Software-Defined Internet of Things (SD-IoT), ML technologies can provide innovative solutions to the security issues faced by IoT devices, including those related to the infrastructure of EV charging systems. This integration helps in proactively identifying and mitigating DDoS attacks, ensuring the reliability and availability of EV charging services [11].

Federated Learning (FL) is a collaborative machine learning technique that enables EVCS to jointly develop a model for detecting DDoS attacks, without the need to exchange sensitive data. This approach maintains data privacy and security across the network. "Personalization" in this context refers to the customization of the universal detection model to align with the specific operational patterns and threat landscapes of individual charging stations [12].

Personalized Federated Learning (PFL) presents a promising solution that addresses these limitations. By enabling collaboration between EVCS without sharing sensitive data, PFL fosters a secure and privacy-preserving environment. Each station retains its data while collaboratively training a robust DDoS detection model. The implementation of PFL for DDoS attack detection in EVCS offers several potential benefits. Firstly, it enhances the security and resilience of the charging infrastructure by providing a more comprehensive and adaptable defense against evolving threats. Secondly, PFL empowers individual stations with personalized threat detection capabilities, allowing for targeted mitigation strategies. Finally, the distributed nature of PFL reduces the burden on any single station, fostering a collaborative approach to cybersecurity within the EV charging network. This collaborative defense system ultimately benefits all stakeholders, ensuring the reliable and secure operation of EV charging infrastructure.

Problem Statement

EVCS are widely installed as a result of the growing integration of EVs into transportation networks. Nevertheless, these charging stations are susceptible to cyberattacks, especially DDoS assaults, which have the potential to interfere with the charging process and jeopardize the EV infrastructure's overall dependability. Conventional security techniques, such as anomaly and signature-based methods, are not as effective at identifying and thwarting DDoS attacks as they have been. While anomaly detection techniques can produce false positives and have difficulty identifying new threats, signature-based techniques are limited to combating established attacks.

Main Contributions:

- Proposed ML, particularly PFL, as a method to improve DDoS attack detection for EVCS.
- Investigated the viability and effectiveness of PFL in creating personalized threat detection models for individual EVCS while safeguarding data privacy and security.
- Highlighted the limitations of existing security methods (signature-based and anomaly detection) in protecting against evolving DDoS threats targeting EVCS.
- Developed a collaborative defense system for EVCS, enabling stations to work together on threat detection without sharing sensitive data, thus enhancing overall security and resilience.
- Contributed to advancements in cybersecurity for EV infrastructure, supporting the development of sustainable transportation policies and infrastructure.

The structure of this paper is organized as follows: Sect. 2 provides essential background information, discussing the vulnerabilities in EV charging systems (2.1) and the concept of Federated Learning (2.3). Section 3 reviews related work in the field, focusing on the integration of AI and cybersecurity. In Sect. 4, we introduce our methodology, specifically the AI-DefendNet algorithm, which is designed for adaptive DDoS detection and classification in EV charging systems. Section 5 elaborates on IntelliGuardXAI, a sophisticated algorithm developed to enhance cybersecurity through a multi-phase process that ensures robust protection against cyberattacks while maintaining transparency and interpretability. Section 6 presents the results of our experimental evaluation, analyzing the effectiveness of the proposed methods. Finally, Sect. 7 concludes the paper, summarizing the key findings and discussing future research directions.

2 Background

This section provides an overview of EV Charging system vulnerabilities, attacks on EV charging systems, and Federated Learning.

2.1 EV Charging system vulnerabilities

The surge in EV usage has brought to the fore significant concerns regarding security and privacy that necessitate urgent attention. A primary concern is the potential for cyberattacks that could inflict substantial harm on the vehicle and its occupants. For instance, a cybercriminal might gain control over a car's breaking or acceleration, leading to dangerous incidents. Moreover, a breach in an EV's onboard computing could expose private and sensitive information to risk. The risk of theft of valuable components from vehicles and the exploitation of charging stations for unauthorized means are additional security issues.

A notable example occurred in March 2022 when several EV charging stations near Moscow were hacked, rendering them unusable for EV owners [13]. Similarly, the Combined Charging System (CCS) has shown susceptibility to a novel attack dubbed "Brokenwire," which disrupts the communication between the car and the charging unit, resulting in halted charging sessions. This attack, which involves electromagnetic interference, can be executed from a distance [14]. By April 2022, vulnerabilities within the infrastructure were highlighted when EV charging points at a UK council parking facility were hacked to display an unauthorized website on their interface. Such incidents not only challenge the effectiveness of existing security protocols for charging stations but also underscore the broader risks associated with cyberattacks on public EV charging systems [15].

While efforts to mitigate these risks are underway, further measures are required to bolster cybersecurity. To keep pace with the evolving landscape of cyber threats, there is a need for continuous reassessment and enhancement of security protocols. Collaboration among vehicle manufacturers, operators of charging networks, and cybersecurity experts is crucial to formulate industry standards and best practices for EV cybersecurity. Protecting the privacy of EV users is paramount, considering the significant data generated by EVs that could be

exploited. The implementation of robust security mechanisms, including encryption, authentication protocols, and secure communication channels, is essential to tackle these security and privacy issues.

Establishing stringent privacy policies and regulations is also vital to protect the data and privacy of EV users. Exploring advanced technologies such as blockchain, secure communication protocols, machine learning, and intrusion detection systems could further fortify the resilience of EV charging infrastructure against cyberattacks. To conclude, safeguarding the integrity and safety of EV charging systems against emerging cyber threats is imperative, necessitating enhanced security measures, development of industry standards, heightened awareness, and ongoing research investment. Figure 2 shows different types of attacks that can target EV charging systems.

2.2 Federated learning

Federated Learning (FL) is a form of distributed deep learning that enables multiple clients to participate in building a unified machine learning model without sharing their private data [16]. In this method, multiple decentralized devices or servers, referred to as clients, train an algorithm locally using their data within individual network slices, such as within different segments of a 5G network. The distinctive feature of FL is that it maintains data privacy by exchanging only model weights during training, not the raw data itself. This approach not only preserves privacy and enhances security but also addresses data access rights and heterogeneity.

The concept of FL is a departure from traditional centralized machine learning approaches that require aggregating local datasets on a single server and differs from standard decentralized methods that assume data distribution is uniform across nodes. FL involves training local models on clients' datasets and then using an aggregation function to form a global model. This process, conducted over several rounds, significantly reduces communication overhead compared to traditional centralized learning methods by avoiding the transmission of large-sized raw data. Despite the challenges in achieving performance comparable to centralized models, FL aims to provide similar prediction results while managing the diverse distribution of data across nodes. The central server plays a pivotal role in initializing the model and aggregating the weights after each training round, thereby ensuring that the learning process leads to an effective and consolidated global model. Table 1 illustrates the main variations in Federated Learning.

In this work, we focus on implementing FHL Model, considering the diversity and specificity of clients and devices involved. Various parameters which are critical for enhancing the efficiency of the FL process are illustrated in Table 2.

The process of aggregating model updates from various devices in Federated Learning is pivotal for maintaining privacy and achieving a unified model. The choice of an appropriate aggregation method is contingent

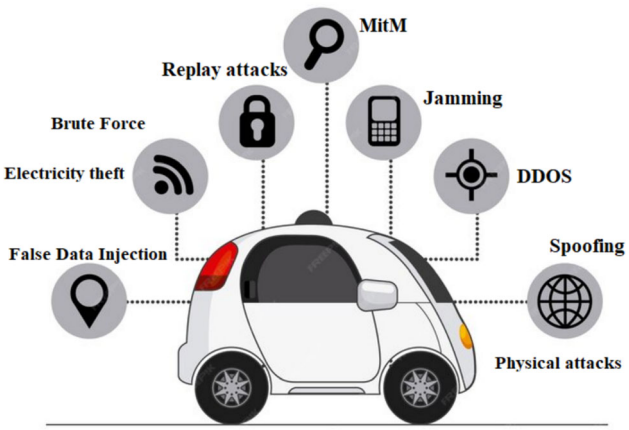


Fig. 2 Types of attacks in EV Charging system

Table 1 Main variations in Federated learning

FL Variant	Description
Federated centralized learning (FCL)	Employs a central server to oversee and coordinate the learning process across all nodes. Initially, the server selects the nodes and integrates their model updates. However, this centralization can lead to bottlenecks, as every node update must pass through one entity
Federated decentralized learning (FDL)	Learning nodes independently collaborates to develop the global model without a central server, reducing the risk of a single point of failure. The model updates circulate among the nodes, although the network's structure could affect the learning efficiency
Federated heterogeneous learning (FHL)	Accept the diversity of clients, such as smartphones and IoT devices, in various sectors. It operates on the premise that both local and global models share the same architecture

Table 2 Various parameters of Federated learning [17, 18]

Parameter	Definition
Number of Federated Learning cycles (R)	The total number of complete learning cycles across all nodes in the Federated Learning process
Total count of nodes involved (N)	The overall number of nodes (devices or servers) participating in the Federated Learning network
Proportion of nodes engaged per cycle (FN)	The fraction of the total nodes that are selected to participate in each learning cycle
Batch size for each node during iterations (BS)	The number of data samples processed by each node before updating its model in a single iteration
Count of training iterations before updates (I)	The number of local training iterations a node completes before contributing to the global model
Local learning rate (η)	The rate at which each node adapts its model during training, influencing the magnitude of updates

Table 3 Overview of aggregation methods in FL [17, 18]

Aggregation technique	Description
Federated Averaging (FedAvg)	The first designed aggregation method where each device updates its local model using its data then sends the updates to a central server. The server calculates the weighted average of these updates to create a global model. Weights are often based on data quantity or device reliability
Federated Learning with Secure Aggregation (FedSecAgg)	Uses cryptographic techniques like secure multi-party computation (MPC) to merge model updates while ensuring anonymity. Devices encrypt their updates, keeping the individual submissions confidential from the server
Federated Quantization	Local devices quantize or compress their model updates before sending them to the server, reducing communication costs. The server then consolidates these updates to update the global model, balancing communication efficiency with model accuracy
Personalization and Differential Privacy	Tailors Federated Learning applications to individual devices while ensuring differential privacy. This involves aggregation methods that allow for model customization for each device, incorporating privacy-preserving mechanisms to protect user data

upon the application's unique demands and objectives. Literature identifies several aggregation techniques employed in Federated Learning, each with its advantages depending on the scenario. Table 3 describes different aggregation methods in FL.

3 Related work

In recent years, there has been a lot of talk about combining artificial intelligence (AI) and cybersecurity. This section examines notable studies and research initiatives in this field, highlighting significant contributions and trends.

3.1 IDS in IoT applications

Several studies have offered methods for detecting intrusions within IoT frameworks. For instance, [19] adapted a signature-based intrusion detection system, Suricata, for thwarting Denial-of-Service attacks on 6LoWPAN networks. This IDS is designed for a centralized host and functions by monitoring channel interference and packet drop rates to accurately confirm attacks and minimize false positives.

Deep learning techniques have also been employed for anomaly detection in IIoT and IICS environments as seen in [20]. This research utilized a Deep Auto Encoder (DAE) to learn from standard network activity and generate initial parameters for a Deep Feed Forward Neural Network (DFFNN). The latter is then used in the detection of known and novel attack patterns, with performance assessed on NSL-KDD and UNSW-NB15 datasets, achieving 98.6% accuracy with the former and 92.4% with the latter, along with manageable false-positive rates.

In addition, [21] introduced an algorithm that combines Snort and ClamAV intrusion pattern sets, optimized for IoT devices like the Raspberry Pi with an Omnivision 5647 sensor. This method enhances computational efficiency by reducing redundant payload-to-signature comparisons, particularly on IoT nodes with constrained resources, showing a twofold speed increase over traditional algorithms under resource constraints.

The approach in [22] offered an architecture leveraging Random Neural Networks and LSTM to spot SYN flooding attacks in IoT networks, with a self-curated dataset derived from virtual network traffic captures. Meanwhile, [23] discusses a sequential attack detection framework that applies three distinct machine learning models for IoT network security.

For distributed IoT environments, [24] suggested a fog-based semi-supervised learning model that surpassed centralized solutions in detection speed and accuracy, verified using the NSL-KDD dataset. Another botnet detection system targeting anomalies in 6LoWPAN sensor networks was proposed in [25], alerting to deviations in average behaviors gauged by TCP control fields, packet lengths, and connection numbers for individual sensors.

Moreover, [26] developed an internal anomaly detection system for IoT, designed not to burden low-capacity nodes but to monitor traffic flow from nearby nodes for signs of unusual activity, learning standard traffic patterns to identify anomalies. Additionally, [27] explored a deep packet inspection method tailored for resource-limited IoT devices, processing payload data as byte sequences and utilizing n-grams for feature selection via a bit-pattern matching algorithm. Tests on internet-connected devices indicated promisingly low false-positive rates for various types of attacks, which benefits users relying on these devices for security.

In [28], researchers identified abnormal patterns in resource-constrained 6LoWPAN networks through the analysis of energy consumption. They established simple power models under mesh-under and route-over topologies. This detection mechanism, upon identifying anomalies, initiates an alert and removes the compromised node from the routing table, though the rate of false alarms has been noted.

In [29], a set of three algorithms was formulated to spot wormhole attacks within IoT networks. These algorithms detect anomalies based on abnormal control packet flows at the ends of a tunnel or an unusually large cluster of neighboring nodes. The system reported a 94% success rate in identifying wormhole attacks and an 87% success rate in pinpointing the attacking node. Despite its low power and memory footprint suitable for IoT devices, the research did not address the rate of false positives.

Researchers in [30] employed Principal Component Analysis (PCA) to pare down feature sets and utilized classifiers like Softmax Regression and KNN. They crafted an IDS tailored for real-time IoT applications. While Softmax Regression provided a more efficient system in terms of computation and speed, KNN edged out in accuracy by 1%, according to tests with the KDD CUP 99 dataset.

A dual-layer classification and dimension reduction technique was introduced in [31], reducing the KDD CUP 99 dataset's 41 features to four, followed by classification via Naive Bayes and KNN algorithms. The approach, applying PCA and Linear Discriminant Analysis (LDA), was validated on the NSL-KDD dataset, yielding an

84.86% detection rate and a 4.86% false-positive rate for various types of intrusions, including probe, DoS, U2R, and R2L attacks.

Researchers in [32] explore a novel anomaly detection approach in IoT networks, integrating Federated Learning with deep neural networks (DNN). This method enhances privacy by keeping raw data on IoT devices and only sharing model updates with a centralized server. The efficacy of this DNN-based intrusion detection system was compared to traditional deep learning models, showcasing improved accuracy and reduced false alarm rates. Using the IoT-Botnet 2020 dataset for assessment, this technique proved more accurate and reliable than previous models, emphasizing the advantages of uniting Federated Learning with deep learning to boost security in IoT contexts.

3.2 Federated learning in IoT intrusion detection

Recently, the development of intrusion detection systems (IDS) leveraging Federated Learning has emerged as a significant area of focus within the research community. The study in [33] showcases Fed-ANIDS, a hybrid system merging Federated Learning with autoencoder-based anomaly detection to bolster network intrusion detection. This system mitigates the privacy risks of centralized databases by computing intrusion scores from normal traffic reconstruction errors across a distributed network. Evaluated with several datasets, Fed-ANIDS demonstrated high accuracy and low false positives, favoring autoencoder models over GAN-based models for efficient threat detection that maintains privacy.

In [34], researchers deployed Federated Learning (FL) for anomaly detection in Internet of Things (IoT) environments. The paper draws attention to the inherent limitations of FL that arise from restricted data access on IoT devices, issues with the balance of classes, and the variability in device capabilities. The study examines how data augmentation techniques can be leveraged to enhance the performance of anomaly detection across IoT networks, testing these methods against three datasets that are openly available.

In [35], the authors have proposed a Recurrent Neural Network (RNN) trained in a federated fashion to detect anomalies in the Internet of Things (IoT) networks using the ModBus network dataset. Seven different models were trained while varying the window size of their time-series data; then, these models were combined using a decision tree vote-based scheme to classify the traffic with high confidence.

In [36], the authors proposed an FL-based framework for intrusion detection in the IoT context. They used a semi-supervised scheme, where auto-encoder-based models are trained using the FedAvg algorithm across different IoT devices. The authors presented a method for calculating a global reconstruction error threshold for the traffic classification task. The FL approach outperformed the local approach while having nearly the same performance as the centralized one, thanks to the introduced global reconstruction error threshold approach.

The authors in [37] expressed the need to personalize a distributed DL-based model, in the context of 5G IoT, due to IoT devices' heterogeneity. The authors presented a similar approach to FedPer, by leveraging personalized layers, that are learned via transfer learning. At each training round, each client performs a train with a publicly shared dataset and communicates the model weights to the server to perform the aggregation (FedAvg). Then, the server returns the aggregated model to the clients, and each client performs a training epoch with its local data, on its personalized layers. The process is repeated for all training rounds.

The authors in [38] investigated the implementation of DL-based IDS models, whether centralized, local, or based on FL. They evaluated the NSL-KDD dataset in IID and Non-IID partitioning settings. The authors reported that FedAvg obtained comparable accuracy to the centralized approach while outperforming the on-device models. Besides, under Non-IID settings, FedAvg was marginally higher than on-device and significantly exceeded by the centralized approach.

The authors in [39] developed an intrusion detection system aimed at safeguarding vehicular networks, incorporating Federated Learning with blockchain technology. Their system employs a secret-sharing-based mask noise model uploading algorithm derived from Federated Learning, which ensures the secure upload of local model parameters. Additionally, the system enhances model accuracy through a model quality-based evaluation

algorithm. In another study [40], a hierarchical Federated Learning-based intrusion detection system is crafted to bolster the security of the advanced metering infrastructure in smart grids. This system utilizes a Transformer as the detection model to improve the classification performance of Federated Learning while safeguarding the privacy of smart meter data.

Furthermore, addressing the security challenges in extensive and diverse industrial cyber-physical systems, a federated deep learning framework named DeepFed was introduced in [17]. This approach begins by creating an intrusion detection classification model employing convolutional neural networks and gated recurrent units. It then leverages a Federated Learning framework to indirectly gather data resources from various operational sites, forming a robust intrusion detection model. The effectiveness of this scheme is validated by its outstanding performance on real datasets from natural gas pipeline systems.

3.3 Distributed denial-of-service (DDOS) attacks

Distributed Denial-of-Service (DDoS) attacks [41] are a collection of attacks in which an intruder obstructs or denies genuine clients from gaining access to their organization's administrations or assets through dispersed attack sources. An intruder can make a botnet by utilizing feeble Net-related gadgets similar to Internet of Things gadgets, and coordinates the botnets through a control server to dispatch attacks; subsequently, setback gets massive source-moved attacks deals bargains from the disseminate.

Many machine learning methods have been developed for detecting DDoS attacks. P. Xiao [42] suggested a robust discovery strategy based on KNN and association analysis to detect DDoS attacks. C. She [43] created an identification plot for application layer DDoS attacks, specifically SYN flooding attacks, HTTP flooding attacks, and NTP augmentation attacks, using an OC-SVM (One-Class Support Vector Machine).

R. Vishwakarma proposed a persuasive solution for detecting botnet-based Distributed Denial-of-Service attacks in IoT by combining honeypots and ML-based algorithms [44]. In this regard, they compromised various IoT honeypots to obtain device malware establishment endeavors and embraced unaided ML methods, for example, grouping and abnormality recognition, to robotize the course of location and anticipate future security dangers by separating highlights from honeypots. With the help of feed-forward and back proliferation computations, Asad introduced in [45] a detection approach that relied on ANN to definitively distinguish various application layer DDoS attacks.

M. Roopak created a model in [46] that focuses on text recognition at the bundle level and employs RNN (Recurrent Neural Network) techniques with Bidirectional Long Short-Term Memory (LSTM) to detect botnet activity within consumer IoT networks. In [47], Meidan, Bohadana, and Mathov constructed a model; to demonstrate the model's applicability, nine commercial IoT devices were targeted with the well-known DDoS attacks Mirai and BASHLITE. Using advanced autoencoders known as N-BaIoT, they discriminated against unusual commercial transactions and corrupted IoT devices.

Doshi and Feamster used K-nearest neighbors (KNN), support vector machines (SVM), arbitrary woods, choice trees, and brain organizations to collect information, extract highlights, and doubly characterize organization deals in [48]. These simulations concentrate on network Centre boxes (for network switches, firewalls, and switches) and other devices that may be required for a continuous DDoS attack. Furthermore, C. She [49] developed a DDoS attack detection and warning structure based on a multi-channel CNN by isolating highlights based on time, space, bundle, and so on.

R. Doriguzzi-Corin [50] proposed a CNN-based DDoS recognition engineering, in which they designated a common sense, lightweight execution with low handling above and attack location time. Y. Jia [51] proposed an IoT DDoS guard component named FlowGuard, in which they planned two parts, specifically Flow-Filter and Flow-Handler. M. Roopak [52] proposed a high-level disruption localization approach for detecting DDoS attacks in IoT enterprises. They used a multi-objective streamlining technique at the underlying stage for highlight extraction on the chosen dataset based on six main goals for reducing information and Deep Learning

models CNN with the combination of LSTM for attack categorization. DDoS attack detection research employing machine learning and deep learning techniques is shown in Table 4.

DDoS attacks have increased alarmingly in smart grid-based EV charging stations. To address these dangers, effective mitigation strategies must be developed on a foundation of extensive datasets illustrating various attack scenarios. Currently, the lack of such a dataset highlights a substantial research gap. To fill this gap, Y. Kim [59] presents the CICEV2023 dataset, which was carefully curated to include four separate attack scenarios targeting EVs within smart grid frameworks.

The effort to create this dataset included the creation of a dedicated simulator that was rigorously developed to emulate an authentication protocol inherent in EV charging infrastructure while launching DDoS assaults explicitly targeting EV authentication operations. This dataset contributes significantly to the advancement of research activities in the field by providing a complete testbed for analyzing and designing resilient protection

Table 4 DDoS attack detection techniques

Ref	Year	Methods used	Data set	Research gap
[53]	2019	ML Technique (Random Forest)	Used TFN2K tool to conduct local DDoS attacks	Limited dataset from TFN2K tool restricts model generalizability to diverse real-world DDoS attack vectors and network conditions. This may lead to poor performance against novel or evolving attack strategies
[52]	2018	LSTM (Long short-Term memory)	DDoS attack software	LSTM's high computational complexity makes it challenging to train on the extensive datasets required for robust DDoS detection, hindering real-time applicability and scalability
[54]	2019	PCA and RNN	KDD CUP 1999	The KDD CUP 1999 dataset's age and absence of contemporary attack vectors limits the model's effectiveness against current DDoS threats, potentially leading to high false negatives
[46]	2019	CNN + LSTM	CICIDS2017	The combined CNN + LSTM architecture introduces high model complexity, increasing the risk of overfitting to the CICIDS2017 training data and potentially reducing its ability to generalize to unseen attacks
[55]	2017	RNN	UNB ISCX	RNN's reliance on large-scale labeled data poses a challenge for real-time DDoS detection where labeled data may be scarce or unavailable. Furthermore, the computational demands of processing large datasets can hinder timely analysis
[50]	2020	CNN	ISCX2012, CIC 2017, CSECIS 2018	The black-box nature of CNNs makes it difficult to interpret the features learned for DDoS detection, hindering understanding of the model's decision-making process and potentially limiting trust in its predictions
[47]	2019	SVM, KNN, ANN	KDDCUP	Traditional machine learning algorithms like SVM, KNN, and ANN suffer from computational inefficiency when applied to the large-scale datasets typical of DDoS attacks, making them less suitable for real-time or high-throughput network environments
[44]	2020	LSTM, CNN	CICDDoS2019	The combined LSTM and CNN model requires significant computational resources and long training times, which can be a barrier to rapid deployment and adaptation to evolving attack landscapes
[56]	2020	ANN SMOTE	BOT-IOT	Using SMOTE to generate synthetic data for BOT-IOT introduces potential biases and may not accurately reflect the characteristics of real-world botnet attacks, potentially leading to model inaccuracies. Additionally, data imbalance issues within the BOT-IOT dataset may further skew model performance
[57]	2019	RF with n-estimate	CICIDS 2017	Random Forest's performance is sensitive to hyperparameter selection, requiring extensive and potentially time-consuming fine-tuning to achieve optimal results for DDoS detection
[58]	2019	KNN, MLP, SVM	KDD CUP99, NSL-KDD	KNN, MLP, and SVM trained on the KDD CUP99 and NSL-KDD datasets may exhibit poor generalization to new or unseen DDoS attack types due to the limited and outdated nature of the training data
[52]	2020	CNN + LSTM	CICIDS2017	The combined CNN + LSTM model is computationally expensive, posing scalability challenges for real-time DDoS detection in high-traffic network environments

mechanisms against these developing threats within smart grid-based EV charging stations. Table 5 compares many previous techniques used for DDoS attack detection in smart grid systems.

Table 5 compares several algorithms used in DDoS attack detection in smart grid systems, highlighting their essential characteristics, advantages, and limits, allowing for a rapid grasp of their functions and potential trade-offs. Differences between EV Network and Classical Network Threats EVs have distinct security problems that set them apart from traditional network security. The next section goes over the specific security considerations of EV networks and underlines the significant distinctions between them and regular network environments:

(1) Distinctive Communication Protocols: For vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication, EVs rely on specialized communication protocols. These protocols bring new weaknesses and attack avenues that are not found in typical network security protocols. To enable secure and dependable communication throughout the EV ecosystem, these protocols must be carefully implemented.

(2) Integrating physical components (such as batteries and charging stations) with cyber systems (such as in-vehicle software and charging infrastructure networks) results in a complicated interaction between the physical and digital realms. Security flaws in either component can have far-reaching ramifications for the EV system's overall safety and functionality. To protect against such risks, a comprehensive approach comprising both physical and cybersecurity protections is required.

(3) Battery Security: The security of EV batteries is paramount, as they serve as the primary energy source. Unauthorized access or manipulating battery systems can lead to severe safety risks, such as fire incidents or compromised vehicle performance. Robust security measures must be in place to safeguard EV batteries from unauthorized access, tampering, or malicious attacks.

Table 5 A comparison of different prior algorithms for DDoS attacks

Algorithm	Description	Pros	Cons
KNN with Association	Uses K-Nearest Neighbors and association analysis	- Effectively detects DDoS attacks	- High computational requirement
OC-SVM	Applies One-Class. Support Vector Machine for detection	- Identifies application-layer DDoS attacks	- Challenging benign/malicious distinction
Honeypot & ML	Integrates honeypots and ML-based algorithms	- Automates detection and predicts threats	- May require significant resource setup
ANN-based Detection	Relies on Artificial Neural Networks for attack identification	- Offers definitive attack distinction	- Potentially high training time
CNN + LSTM	Utilizes Convolutional Neural Network with LSTM for detection	- Good for text recognition	- Requires substantial data preprocessing
Autoencoders (N-BaIoT)	Employs advanced autoencoders for distinguishing attacks	- Discriminates unusual transactions	- Might require a specific IoT setup
FlowGuard	Deploys a two-part IoT DDoS guard mechanism	- Provides protective IoT measures	- Specific to IoT infrastructure
Lightweight CNN-based	Utilizes a lightweight CNN for DDoS recognition	- Low processing overhead	- May trade-off precision for performance
Flow-Filter & Flow-Handler	Utilizes two components for IoT DDoS detection	- Segregates flow for better analysis	- Specific to IoT devices and protocols
CNN with Feature Optimization	Employs CNN with feature reduction for attack detection	- Reduces data size and retains accuracy	- Might require extensive feature selection
KNN with Association	Uses K-Nearest Neighbors and association analysis	- Effectively detects DDoS attacks	- High computational requirement
OC-SVM	Applies One-Class. Support Vector Machine for detection	- Identifies application-layer DDoS attacks	- Challenging benign/malicious distinction

(4) Charging Infrastructure: Electric vehicles rely significantly on charging infrastructure, which has its own set of security risks. Securing charging stations is critical to preventing unwanted access to billing systems, ensuring transaction integrity, and protecting against potential power grid assaults. The interconnected nature of charging infrastructure needs comprehensive security methods to ensure the charging process's dependability and trustworthiness.

(5) Concerns about privacy: EVs collect and process sensitive driving behavior, location, and energy consumption data. Beyond traditional network security concerns, protecting the privacy of sensitive data presents additional issues. To address privacy concerns in the EV ecosystem, it is critical to protect personal information from illegal access and to ensure ethical data management methods.

Because of the specific characteristics and requirements of electric vehicles, EV network security differs from traditional network security. The use of unique communication protocols, the integration of physical and cyber components, battery security, charging infrastructure protection, and the handling of privacy concerns all demand customized security measures. By recognizing and addressing these specific security issues, we can design a robust and resilient EV network infrastructure that assures the safety, privacy, and security of EV users, as well as the overall ecosystem.

The following are the research gaps revealed in prior algorithms used for DDoS attack detection in smart grid systems:

- Lack of standardization issue in EV charging network architecture.
- **Computing Intensity:** Some methods, like as KNN with Association and CNN + LSTM, have high computing requirements, which may limit their scalability and real-time applicability.
- Discriminating between Benign and Malicious Behavior: Techniques such as OC-SVM and Behavior-based Analysis confront difficulties in successfully discriminating between benign and malicious behavior, limiting their accuracy in specific cases.
- Approaches that integrate honeypots and ML, such as Honeypot & ML, may involve significant resource setup or maintenance efforts, limiting their practical adoption.
- **Training Time:** Algorithms based on Artificial Neural Networks (ANN) may necessitate lengthy training, thus impeding their responsiveness in quickly changing attack scenarios.
- **Preprocessing Overhead:** Methods such as CNN + LSTM may necessitate significant data preprocessing operations, which may reduce their efficiency, particularly in real-time applications.
- Some approaches, such as Autoencoders (N-BaIoT) and Flow-Filter & Flow-Handler, may be specialized to specific IoT infrastructures or protocols, restricting their generalizability.
- **Performance Trade-offs:** While lightweight CNN-based techniques may target low processing overhead, precision may be sacrificed for performance, implying a trade-off between accuracy and efficiency.
- **Attack Type Specificity:** Certain algorithms may be specifically designed for specific DDoS attack types, potentially limiting their adaptability when dealing with a broader spectrum of attack patterns.
- **Challenges with Feature Selection:** Methods such as CNN with Feature Optimization may necessitate considerable feature selection efforts, adding complexity to the model design and implementation process.

3.4 Implementation of personalized federated learning (PFL)

The Personalized Federated Learning (PFL) framework in this study is designed to enable decentralized and privacy-preserving DDoS detection across Electric Vehicle Charging Stations (EVCS). In contrast with standard Federated Learning, PFL introduces a personalization phase to account for the heterogeneity in traffic behavior and attack patterns across individual EVCS clients.

3.4.1 Global model training and aggregation

Initially, a global classification model is trained across multiple EVCS clients using the Federated Averaging (FedAvg) algorithm. Each client receives a copy of the global model and trains it locally on its private dataset for a fixed number of epochs. The locally updated models are then sent back to the central server, where they are aggregated to form a new global model.

3.4.2 Client-side personalization via fine-tuning

After global training, each client performs **local fine-tuning** on the aggregated model using its own dataset. This step allows the model to adapt to unique traffic characteristics and localized DDoS behaviors specific to each EVCS. Fine-tuning is done using a small learning rate and a reduced number of epochs to avoid overfitting while improving detection accuracy in the client's context.

3.4.3 Addressing client heterogeneity

The dataset used (CICIoT2023) reflects various attack scenarios, which were **non-IID distributed** across simulated clients during experimentation. This simulates real-world heterogeneity where different EVCS units may face different threat patterns or operate under varying network conditions. The use of fine-tuning enhances personalization without requiring complex meta-learning or hierarchical architectures, ensuring computational efficiency on resource-constrained EVCS devices.

3.4.4 Justification for design choice

Fine-tuning was chosen as the personalization strategy due to its low overhead and ease of integration into existing Federated Learning workflows. Unlike meta-learning approaches such as Model-Agnostic Meta-Learning (MAML) or hierarchical Bayesian methods, fine-tuning offers a practical balance between performance and deployment feasibility in edge environments typical of EVCS infrastructure.

4 AI-DefendNet: adaptive DDoS detection and classification algorithm for EV charging system

The suggested standardizing architecture for the EV charging network is depicted in Fig. 3, which consists of three layers with related server and protection software.

4.1 Application layer

The highest layer interacts directly with users and manages their interactions with the charging network. It consists of user-facing apps, interfaces, and services that enable various operations such as scheduling charging sessions, managing user accounts, payment processing, and real-time status information. The EV ChargeNet Central Server handles these functions, maintaining a consistent user experience, while SecureCharge Shield provides powerful security standards for data encryption, access control, and user authentication. The application layer combines six main components; **(i) Server:** EV ChargeNet Central Server, **(ii) Protection Software:** SecureCharge Shield, **(iii) Security/Padlock:** represents security features or authentication. **(iv) User Settings:** provides access to user preferences and settings. **(v) Statistics/Analytics:** provides charging history and data analytics. **(vi) Map/GPS:** helps in locating nearby charging stations.

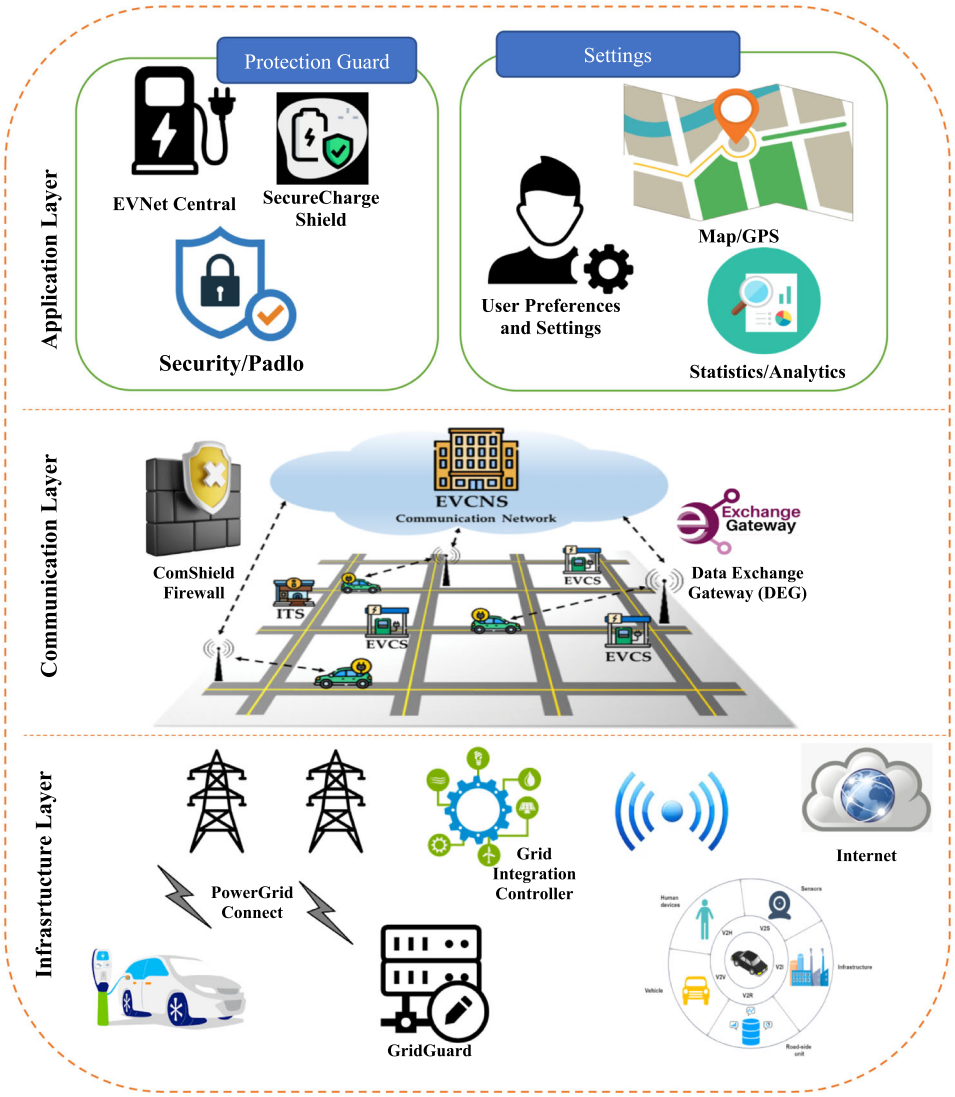


Fig. 3 EV charging network architecture

4.2 Communication layer

The communication layer serves as a middleman between various components of the EV charging ecosystem, such as EVs, charging stations, and the central server. The EVCS Communication Hub oversees coordinating data interchange, protocol conversions, and communication protocols between charging stations, vehicles, and the central server. ComShield Firewall enables secure data transfer, traffic monitoring, and network-level threat protection such as DDoS attacks, intrusion prevention, and secure data routing.

The communication layer combines three main components;

(i) Communication Hub for Electric Vehicle Charging Systems (EVCS): This critical component manages the data flow between electric vehicles, charging stations, and the central server. It is in charge of ensuring the effective transmission of information, protocol conversions, and compatibility among the many communication protocols used in the EV charging ecosystem.

(ii) ComShield Firewall: The ComShield Firewall runs within the Communication Layer as a robust security tool to strengthen the network against potential cyber threats. It protects against numerous security concerns such as DDoS attacks, intrusion attempts, and unauthorized access by securing data transmission and monitoring network traffic in real time.

(iii) Data Exchange Gateway (DEG): This component serves as a critical gateway for controlling and facilitating data exchange between various stakeholders in the EV charging ecosystem. The DEG ensures that multiple systems communicate smoothly, standardizes data formats, and performs data transformation between various protocols used by electric vehicles, charging stations, and the central server. It is critical in facilitating interoperability and seamless data flow throughout the network infrastructure.

4.3 Infrastructure layer

The Infrastructure Layer consists of the physical components as well as the electricity infrastructure required for EV charging operations. PowerGrid Connect controls the transfer of power from the grid to charging stations and electric vehicles. GridGuard Sentinel focuses on physical infrastructure security, such as power distribution systems, transformers, and charging devices. It employs cybersecurity measures to prevent physical tampering and voltage manipulation, as well as to protect the integrity and safety of the power grid's interconnection with the charging network.

The infrastructure layer combines seven main components; **(i) PowerGrid Connect:** This component manages power transmission from the electrical grid to EV charging stations and electric vehicles. It controls the efficient distribution of electricity, regulates power flow, and assures grid and charging infrastructure compatibility. PowerGrid Connect is critical in supplying the electricity required to charge EVs reliably and sustainably.

(ii) GridGuard Sentinel: With a focus on physical infrastructure security, GridGuard Sentinel employs sophisticated security techniques to secure crucial components within power distribution systems, transformers, and various charging devices. Its major goal is to prevent physical tampering, unauthorized access, or manipulation of the electrical grid's link with the EV charging network. GridGuard Sentinel ensures the resilience and safety of the infrastructure against any threats or intrusions by utilizing cybersecurity processes.

(iii) Energy Storage Systems (ESS): ESS can be connected into the infrastructure to store excess energy during off-peak or excess generation periods. These systems aid in the management of peak demand, the stabilization of grid fluctuations, and the provision of uninterrupted charging services during grid disruptions.

(iv) Grid Integration Controllers: These controllers oversee the interaction of the charging infrastructure with the electrical grid. By coordinating the charging process based on grid conditions and electrical demand, they optimize power flow, manage voltage, and assure grid stability. **(v) Charging Station Controllers:** These are in charge of administering and controlling individual charging stations. To ensure efficient and safe charging operations, they coordinate the charging process, check charging levels, manage power output, and communicate with the grid and EVs.

(vi) Renewable Energy Integration Systems: These systems facilitate the integration and management of renewable energy generation with EV charging in scenarios where renewable energy sources (such as solar or wind power) are incorporated into the grid, optimizing the utilization of clean energy sources for charging EVs. **(vii) Load Balancing Systems:** These systems regulate the distribution of electrical load across different charging stations in order to avoid overloads and ensure equal use of available power resources.

4.4 Federated learning environment setup

To ensure replicability and provide clarity on the Federated Learning framework used in this study, we detail the key parameters of our experimental setup below:

- **Number of Clients:** The simulation involved 20 clients, each representing an individual EVCS node with localized data.
- **Data Distribution:** The dataset was partitioned to reflect a non-IID (non-independent and identically distributed) scenario, where each client holds data with differing attack types and feature distributions, simulating realistic heterogeneity in EVCS environments.

- **Communication Rounds:** The federated training was conducted over 50 communication rounds to balance model convergence and communication overhead.
- **Learning Rates:** A fixed learning rate of 0.01 was employed for all clients during local model updates.
- **Batch Size:** Local training utilized a batch size of 32 samples per iteration.
- **Client Participation Ratio:** At each communication round, 80% of clients were randomly selected to participate in model updates to mimic potential client availability variability.

These specifications aim to realistically emulate the operational conditions of EVCS networks and provide a foundation for replicating and benchmarking the proposed Personalized Federated Learning approach.

5 IntelliGuardXAI: IntelliGuard eXplainable artificial intelligence

IntelliGuardXAI, which stands for IntelliGuard eXplainable Artificial Intelligence, is a sophisticated and comprehensive algorithm developed to improve cybersecurity through a multi-phase process. This novel system smoothly integrates four critical stages, ensuring strong protection against cyber-attacks while maintaining transparency and interpretability. IntelliGuardXAI perfectly integrates four critical phases as in Fig. 4 which are: (i) Data Preparation Step, (ii) Feature Selection Step, (iii) Classification Step, and (iv) eXplainable AI, as illustrated in figure each contributing to its success in improving cybersecurity in IoT environments.

5.1 Data preparation step

In IntelliGuardXAI, the Data Preparation Step ensures that the dataset is of high quality and ready for the next phase. This phase entails duties such as loading raw data, dealing with missing values, and encoding categorical variables to produce a clean and well-organized dataset. The overall steps of the Data Preparation Algorithm (DPA) are illustrated in Algorithm 1.

Algorithm 1: Data Preparation Algorithm (DPA)

- **Input**
  - Raw dataset (e.g., CICIoT2023.csv).
- **Output**
  - Cleaned and processed dataset.
- **Steps**

1. Load the raw dataset into a data structure (e.g., pandas DataFrame).
2. Handle missing values
   a. Identify missing values in the dataset.
   b. Choose an appropriate strategy for handling missing values (e.g., imputation or removal).
   c. Implement the chosen strategy to fill or remove missing values.
3. Handle categorical variables
   a. Identify categorical columns in the dataset.
   b. Apply encoding techniques (e.g., one-hot encoding) to convert categorical variables into numerical format.
4. Scale numerical features
   a. Identify numerical columns in the dataset.
   b. Apply feature scaling methods (e.g., Min-Max scaling or Standard scaling) to standardize numerical features.
5. Remove unnecessary columns
   a. Identify columns that do not contribute significantly to the analysis or classification.
   b. Remove redundant or irrelevant columns from the dataset.
6. Save the prepared dataset for future use in feature selection and classification phases.

5.2 Feature selection step

IntelliGuardXAI's Feature Selection Step uses the Firefly Algorithm to intelligently select a selection of features from the supplied dataset. Feature selection is critical for increasing model efficiency, minimizing overfitting, and improving interpretability. The Firefly Algorithm is a nature-inspired optimization technique that uses the flashing behavior of fireflies to choose the best subset of attributes based on their relevance. The overall steps of the Feature Selection Algorithm (FSA) are illustrated in Algorithm 2.

Algorithm 2: Feature Selection Algorithm (FSA)

- **Input**
  - Prepared dataset with features (X) and labels (y)
- **Output**
  - Subset of selected features
- **Steps**
  1. Initialize a population of fireflies with random feature subsets.
  2. Evaluate the fitness of each firefly based on a fitness function (e.g., classification performance).
  3. Repeat for a maximum of max_iter iterations:
     a. Update the attractiveness of each firefly based on its fitness.
     -Attractiveness (Ai) between two fireflies i and j:

$$Ai = beta0 * exp(-gamma * Dij^2) \quad (1)$$

     Where Dij is the Euclidean distance between fireflies i and j.

     b. Move each firefly towards more attractive fireflies.
     - Movement towards more attractive fireflies:

$$Xi(new) = Xi(old) + Ai * rand() * (Xj - Xi) \quad (2)$$

     - where Xi(new) and Xi(old) are the new and old positions of firefly i, Xj is the position of a more attractive firefly, and rand() is a random number between 0 and 1.

     c. Evaluate the fitness of fireflies after movement.
  4. Identify the firefly with the highest fitness as the optimal feature subset.
  5. Return the selected features from the optimal firefly.

The detailed steps of Firefly Feature Selection are illustrated in Fig. 6. The Firefly Algorithm usually operates in the following order: Randomly locate the firefly (solutions) in the solution space. Objective Function Evaluation: Evaluate each firefly's fitness using the objective function. Attraction and Movement: Make iterative adjustments to firefly positions based on their appeal. Brighter fireflies attract and approach other fireflies. Randomization: Use randomization in the movement to better explore the solution space. Iteration and Convergence: Repeat the process until convergence or a set number of iterations is reached. Solution Selection: Choose the firefly(s) with the highest fitness value to solve the optimization problem.

Strengths of Firefly Algorithm in IntelliGuardXAI:

(i)Global Optimization: The Firefly Algorithm is adept at searching the whole search space for optimal feature subsets. This functionality enables IntelliGuardXAI to uncover feature combinations that contribute to robust and accurate model performance.

(ii)Adaptability: The Firefly Algorithm's parameters, such as the attraction and absorption coefficients, can be fine-tuned based on the features of the dataset and the complexity of the classification problem. IntelliGuardXAI's versatility makes it applicable to a wide range of IoT threat situations.

(iii) Efficient Search: The algorithm uses a distributed method, with fireflies gravitating toward more appealing options. This quick search technique improves IntelliGuardXAI's scalability, allowing it to handle large-scale IoT datasets with a variety of attributes.

(iv) Parallel Processing: Because the Firefly Algorithm operates in parallel, it may efficiently use modern computational architectures. This correlates with the need for real-time processing in IoT contexts, increasing IntelliGuardXAI's overall efficiency.

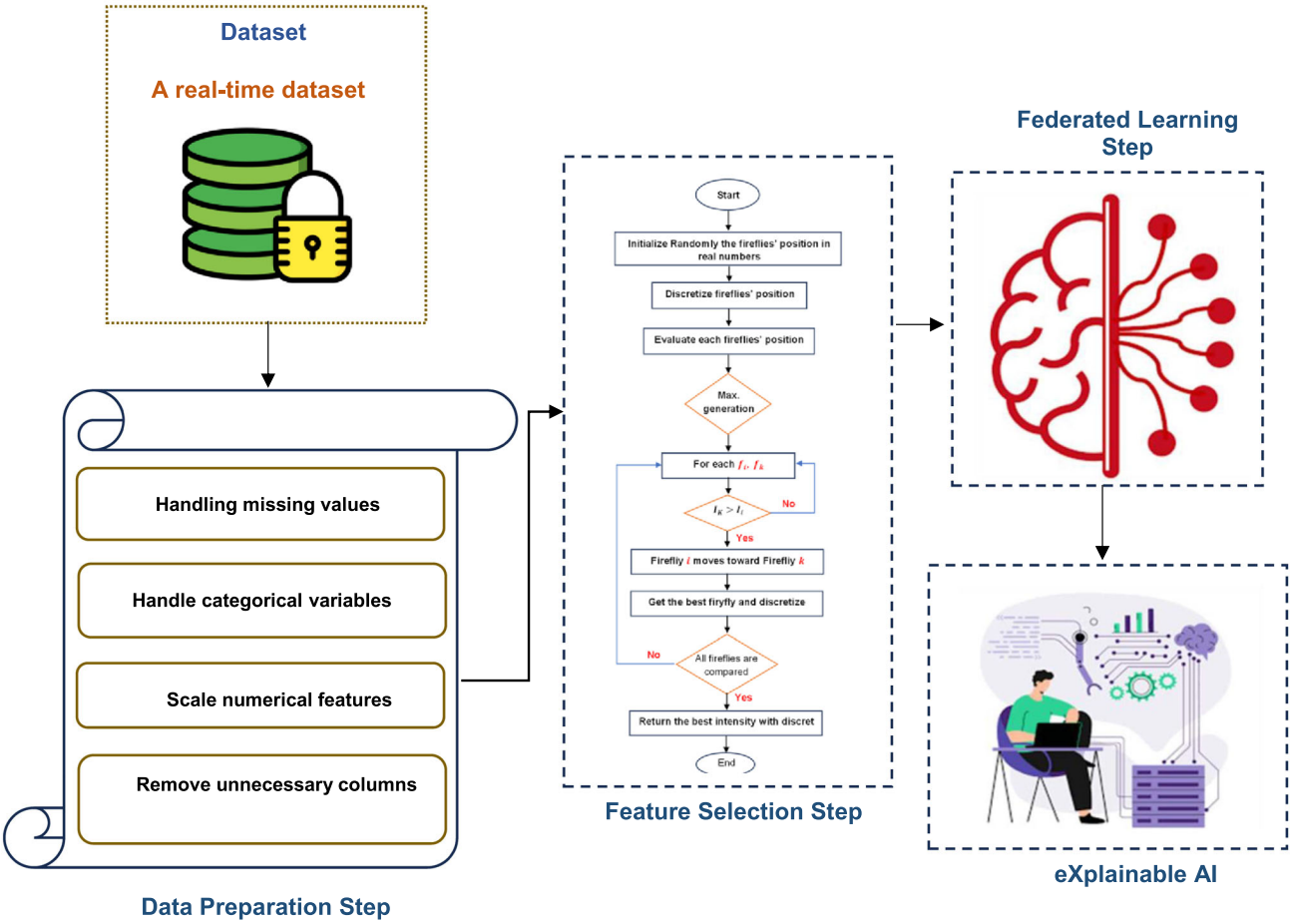


Fig. 4 The IntelliGuardXAI framework

(v) Adaptability: The Firefly Algorithm's adaptability allows it to be used for a variety of attacks across the seven categories (DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai) in the CICIoT2023 dataset.

IntelliGuardXAI can react to a variety of attack circumstances, giving it a reliable solution. By exploiting the Firefly Algorithm's strengths, IntelliGuardXAI is prepared to thrive at the critical phases of feature selection, contributing to the overall performance of the IoT attack classification and detection system.

5.3 Federated learning step

In IntelliGuardXAI, the Classification Step uses the Firefly Algorithm to refine the parameters of a classification model, improving its ability to discriminate between malicious and benign IoT network data. This step follows the Feature Selection phase and comes before the Explainable AI (XAI) component. The overall steps of the Classification Algorithm (CA) are illustrated in Algorithm 3.

Algorithm 3: Classification Algorithm (CA)

- **Input**
  - Selected features from the Feature Selection Step.
  - Preprocessed and normalized dataset.
  - Target labels indicating attack categories.
  - Firefly Algorithm parameters (attraction coefficient, absorption coefficient, etc.).
- **Output**
  - Optimized parameters for the classification model.
  - Trained classification model.
- **Steps**
  1. Initialize Firefly Population
     - Generate an initial population of fireflies representing different solutions in the parameter space.
  2. Evaluate Fitness
     - Evaluate the fitness of each firefly based on the classification performance using the selected features.
  3. Sort Fireflies
     - Sort the fireflies based on their fitness, with higher fitness indicating better classification performance.
  4. Move Fireflies
     - Update the position of fireflies using the attraction and absorption coefficients, considering the fitness values.
  5. Evaluate Updated Fitness
     - Reevaluate the fitness of fireflies after the movement.

$$\boldsymbol{Fitness} = \frac{\boldsymbol{Classification}}{\boldsymbol{Performance}} \quad \boldsymbol{Metric}\ (\boldsymbol{e.g.}, \boldsymbol{F1-score}, \boldsymbol{Accuracy}) \qquad (3)$$

  6. Update Best Solution
     - Update the best solution if a firefly with better fitness is found.

$$\boldsymbol{Move} = \boldsymbol{Current\ Position} + \boldsymbol{\beta} \times \boldsymbol{Attraction} + \boldsymbol{\alpha} \times (\boldsymbol{Absorption} \times \boldsymbol{Random/Number} - \mathbf{0.5}) \qquad (4)$$

  7. Repeat Steps 4-6
     - Iteratively move fireflies and update the best solution until a convergence criterion is met or a maximum number of iterations is reached.
  8. Extract Optimized Parameters
     - Extract the parameters corresponding to the best solution obtained through the Firefly Algorithm.
  9. Train Classification Model
     - Train a classification model (e.g., Logistic Regression) using the selected features and the optimized parameters.

5.4 eXplainable AI step

In IntelliGuardXAI, the Data Preparation Step ensures that the dataset is of high quality and ready for the next phase. This phase entails duties such as loading raw data, dealing with missing values, and encoding categorical variables to produce a clean and well-organized dataset. The overall steps of the eXplainable AI (XAI) Algorithm are illustrated in Algorithm 4 (Fig. 5).

Algorithm 4: eXplainable AI (XAI) Algorithm

- **Input**
  - Optimized Classification Model: The trained classification model from the Classification Step.
  - Selected Features (X): The subset of features obtained from the Feature Selection Step.
  - Instance to Explain (x_instance): A specific instance or sample from the dataset for which an explanation is needed.
  - Model Prediction (y_pred): The prediction made by the classification model for the given instance.
- **Output**
  - Explanation for Model Prediction: A clear and interpretable explanation outlining the factors influencing the model's decision for the given instance.
- **Steps**
  1. Select Model-Agnostic XAI Method
     - Choose an XAI technique suitable for the specific classification model. Examples include LIME, SHAP, or decision tree-based methods.
  2. Generate Explanations
     - Apply the selected XAI method to generate explanations for the model's predictions. This involves approximating the local behavior of the model around the instance of interest.
  3. Interpret Feature Importance
     - Analyze the importance of each selected feature in contributing to the model's decision for the given instance. This step enhances the interpretability of the model's behavior.
  4. Visualize Explanation
     - Create visualizations or textual representations that illustrate how different features influence the prediction. Visualization aids in understanding the rationale behind the model's decision.
  5. Quantify Feature Contributions
     - Quantify the contributions of each feature to the model's prediction, providing a numerical measure of their impact.
- Equations
  - SHAP Values (SHapley Additive exPlanations):

$$\phi i(f)=1/N\sum S\subseteq N\setminus\{i\}[f(S\cup\{i\})-f(S)]$$

  - Here, N is the set of all features, S is a subset of features, f(S) represents the model's output given the input features in S, and $\phi i(f)$ is the SHAP value for feature ii.

6 Implementation and evaluation

This section delves into the practical implementation and evaluation of the proposed PFL approach for DDoS attack detection in EVCS.

6.1 Data description

The dataset used in this paper is the IoT attack dataset [59]. **T**here 33 attacks are executed in an IoT topology composed of 105 devices. The attacks are categorized into seven distinct types: **DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai**. Ultimately, malevolent IoT devices carry out all assaults with the intention of targeting other IoT devices. Table 5 illustrates the dataset description (Table 6).

Threats involving Brute Force Techniques

A Dictionary Brute Force Attack is a systematic method of attempting to gain unauthorized access to a system or account by systematically trying all possible combinations of words or phrases from a pre-existing list, often

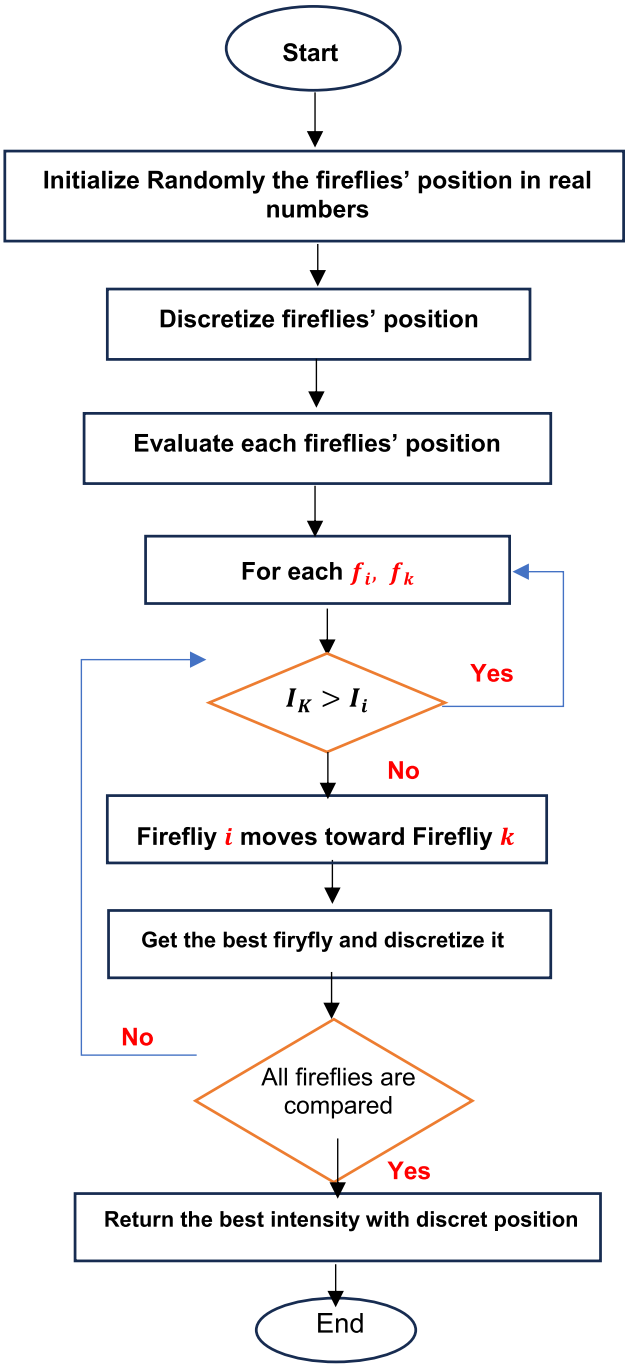


Fig. 5 Steps of firefly feature selection

compiled from dictionaries, word lists, or previously breached passwords. Unlike traditional brute force attacks that systematically try all possible character combinations, a dictionary attack focuses on common words, phrases, or passwords likely to be used by individuals. This method is more efficient than traditional brute force and relies on the likelihood that users often choose easily guessable or commonly used passwords.

Spoofing

ARP spoofing This is a type of spoofing attack employed by hackers to intercept data. In an ARP spoofing attack, a hacker deceives a device into sending messages to the hacker instead of the intended recipient. **DNS spoofing** It is an attack that entails manipulating DNS records to divert users to a deceptive and malicious website, often designed to mimic the user's intended destination.

Leveraging vulnerabilities in web-based platforms

A SQL injection attack entails inserting a SQL query through client input into an application. Successful exploitation can lead to unauthorized access, data manipulation, administrative operations on the database, and potentially compromising the file system or issuing commands to the operating system. **Command injection** is a type of attack aiming to execute arbitrary commands on the host operating system through a vulnerable application. This vulnerability arises when an application transfers unsafe user-provided data (such as forms, cookies, HTTP headers, etc.) to a system shell.

Backdoor is a form of malware that bypasses regular authentication methods, allowing unauthorized access to a system. This grants remote entry to application resources like databases and file servers, enabling attackers to remotely issue system commands and update the malware. **Cross-Site Scripting (XSS) attacks** involve injecting malicious scripts into trusted websites through vulnerable web applications. These attacks exploit flaws in how user input is handled, posing a widespread threat, and potentially causing significant harm.

Uploading attacks focus on a web application by taking advantage of weaknesses in its file upload feature. The objective of a file-uploading attack is to introduce harmful files, like malware, into a specific system, leveraging them to achieve unauthorized access or execute arbitrary code on the targeted system. **Browser hijacking**, a mounting concern within the domain of internet security, constitutes a significant menace to online privacy and safety. This detailed guide strives to furnish an authoritative and preliminary elucidation of the intricacies of browser hijacking, shedding light on its operations and the potential risks it introduces to users**.**

Recon

Ping Sweep, also known as Ping scan or Internet Control Message Protocol (ICMP), is an information-gathering technique that identifies live hosts by sending and receiving ping requests. In this two-way handshake protocol, a host initiates the request, and the receiver responds by sending back packets of information in bytes, resulting in validation and a response to the sender host. **OS scan** involves collecting data about the operating system of a computer or network device. Typically, part of network reconnaissance or vulnerability assessments, aims to understand the target system's characteristics and potential vulnerabilities.

Vulnerability scan systematically identifies weaknesses in computer systems, networks, or applications to assess security posture. The primary aim is to pinpoint areas susceptible to exploitation, utilizing automated tools and risk assessments. Detailed reports guide remediation efforts for overall cybersecurity improvement. **Port scanning** involves identifying open ports on a network that can send or receive data. Additionally, it entails sending packets to specific ports on a host and analyzing responses to uncover potential vulnerabilities.

Host discovery represents an initial stage in network reconnaissance, where the attacker begins with a set of IP addresses within a target network. Employing diverse methods, the adversary assesses the presence of a host at each IP address. Often likened to 'Ping' scanning, host discovery draws parallels to sonar analogies.

Mirai

GREIP In this attack, the GRE packet floods the target system with encapsulated packets, featuring random internal IPs and ports, while the external layer includes authentic Ips. **GREETH** Like GREIP, this attack follows a comparable procedure but emphasizes the packet encapsulation method, specifically based on the ethernet header. The visualization of the dataset is shown in Fig. 6.

The most common attack type is DDoS-ICMP_Flood with 42,340 occurrences, followed by DDoS-UDP_-Flood. The least common are web-based attacks like SqlInjection, CommandInjection, XSS, and Backdoor_Malware, which have very few occurrences in this particular CSV file. There is also benign traffic labeled as BenignTraffic, which is important for training classification models to distinguish between normal and malicious traffic.

Flow duration: Most of the flow durations are concentrated at the lower end, near zero, suggesting that many flows are very short-lived, which is characteristic of some types of attacks that generate a lot of quick traffic. Rate: This feature also shows a heavy concentration near zero, indicating that many flows have low rates, with a few exceptions going up to higher values. Protocol Type: The distribution is a bit more spread out, but there is a clear concentration around certain protocol numbers, likely corresponding to common protocols like TCP and UDP. Figure 7 illustrates the distribution of flaw_duration, distribution of rate, and distribution of protocol type.

6.2 Dataset relevance to EVCS scenarios

The CICIoT2023 dataset, though designed as a general-purpose IoT security dataset, is well suited for evaluating cyberattack detection mechanisms in Electric Vehicle Charging Stations (EVCS). This alignment is based on the following key factors:

- **Threat Similarity**: Many of the attack types in the CICIoT2023 dataset, including DDoS, spoofing, brute force, SQL injection, and command injection, are directly applicable to EVCS environments. EVCS systems—comprising networked chargers, smart meters, backend management platforms, and mobile interfaces—share vulnerabilities with other IoT deployments. These systems are frequently targeted by adversaries due to their internet connectivity and data-driven operations.
- **Feature Alignment**: The dataset includes a range of features such as flow duration, rate, and protocol type, which are relevant to network behavior analysis in EVCS infrastructures. These features can capture patterns indicative of abnormal traffic in charging stations, making them suitable for training and evaluating detection models.
- **Realistic Traffic Simulation**: The dataset simulates legitimate and malicious network traffic across 105 IoT devices and over 33 types of cyberattacks. This simulation creates a heterogeneous and dynamic environment reflective of the multi-device, service-oriented nature of modern EVCS deployments.
- **Generalization and Validation**: Although the dataset is not explicitly derived from real EVCS deployments, it offers a robust and scalable testbed to develop and validate models. The approach proposed in this study can be easily fine-tuned using actual EVCS traffic data, enabling deployment-specific customization and improved domain-specific accuracy.

We have acknowledged this generalization aspect as a limitation in Sect. 6.4 and have indicated our future direction to include real-world EVCS traffic collection and testing. Nevertheless, the dataset offers a strong foundation for preliminary validation of our Personalized Federated Learning (PFL) approach.

Table 6 Comparison of Machine Learning Models for DDoS Attack Detection in EVCS

Model	Hyperparameters	Accuracy
Multilayer Perceptron (MLP)	Hidden_Layer_Size = (50,25)	64%
Gradient Boosting Machine (GBM)	N_estimtator = 15	99%
KNN	N_estimator = 100	98%
Random Forest	N_neighbors = 5	99%

6.3 Results of the feature selection model

Significant results were obtained using the feature selection model in the PFL approach for DDoS attack detection in EVCS. The model improved the effectiveness and interpretability of the classification model by using the Firefly Algorithm to intelligently choose a subset of features from the dataset. Important characteristics that help in the identification of DDoS assaults in EVCS are successfully recognized using the Firefly Algorithm. The model chose a collection of features that optimized the classification model's performance by weighing each feature's significance in relation to the classification task. Figure 8 depicts the results of feature selection.

The feature selection model's outcomes show that it can enhance the system's overall classification performance. The model decreases the difficulty of the classification process and enhances the interpretability of the final classification model by only choosing the most pertinent features.

6.4 Performance evaluation of machine learning models for DDoS attack detection

This subsection delves into the evaluation of various machine learning models for their effectiveness in detecting DDoS attacks targeting EVCS. It presents the results in Table 7, which provides a comparison of the models' performance based on detection accuracy. Table 8 presents the results of evaluating different machine learning models for their effectiveness in detecting DDoS attacks on EVCS.

The evaluation of various machine learning models for DDoS attack detection in EVCS yielded promising results, as shown in Table 6. Both Gradient Boosting Machine (GBM) and Random Forest achieved high accuracy, with 99% and 98%, respectively, in identifying these attacks. This suggests that these models hold significant potential for effective DDoS detection within the EV charging infrastructure. While Multilayer Perceptron (MLP) also achieved a reasonable accuracy of 64%, it fell behind GBM and Random Forest. Further investigation into MLP hyperparameter tuning might be necessary to improve its performance.

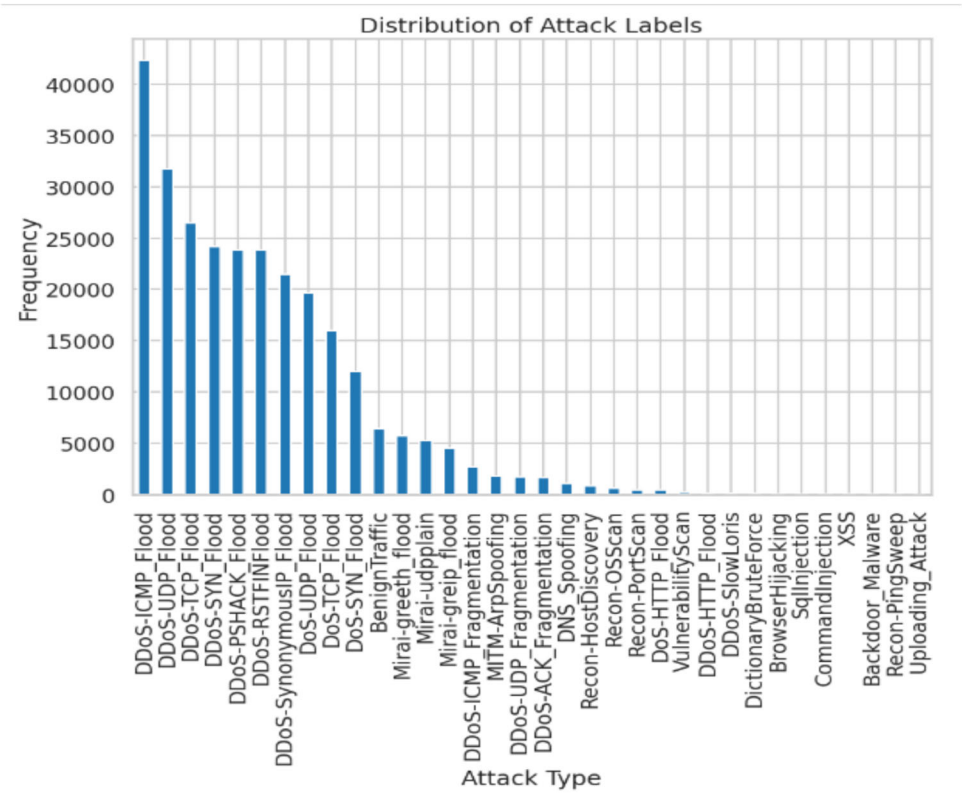


Fig. 6 Dataset Visualization

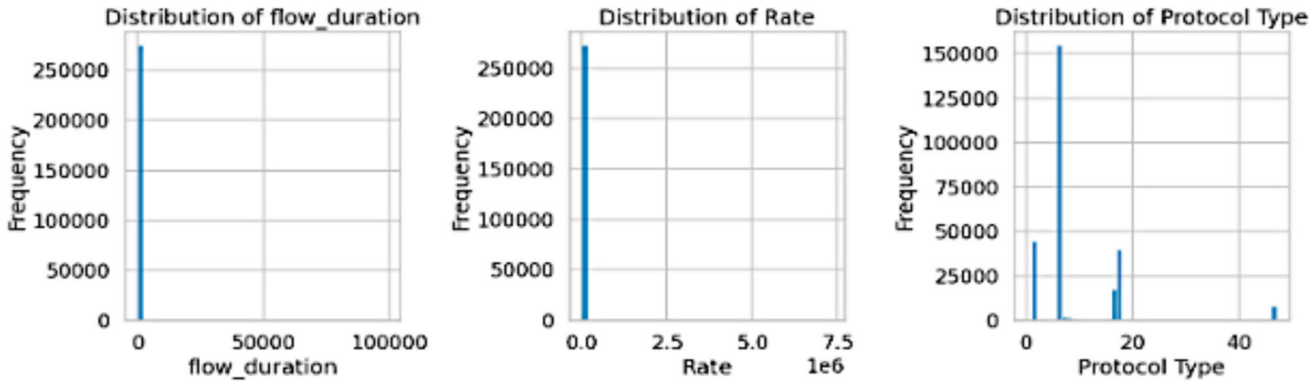


Fig. 7 Distribution of flaw_duration

KNN displayed a competitive accuracy of 98%, making it another viable candidate for DDoS detection. These findings highlight the effectiveness of machine learning approaches in this domain, paving the way for further exploration and integration within the proposed PFL system. The results for each attack type are depicted in Fig. 9.

In the context of the suggested PFL technique for DDoS attack detection in EVCS, Fig. 7 shows the distribution of the target variable following processing. The target variable is a representation of the many DDoS attack types that are classified inside the dataset. After processing, the target variable's distribution offers information about the distribution and frequency of various DDoS attack kinds in the EVCS environment. In the PFL technique, the processed target variable is crucial for both training and testing the classification model.

After preprocessing and encoding, Fig. 10 displays the distribution of DDoS attack types, such as DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai attacks. Because each form of attack is represented by a distinct category, the classification model is able to discern between various attack types. To evaluate the effectiveness of the classification model, it is essential to comprehend the distribution of the target variable following processing. Researchers and practitioners can learn a great deal about the frequency and features of DDoS attacks in EVCS by examining this distribution; this will help to improve the efficacy of detection and mitigation techniques. Figure 11 depicts the confusion matrix.

Figure 12 illustrates the precision, recall, F1-score, and support for various classes in our evaluation. Precision refers to the proportion of true positives among all predicted positives, while recall indicates the proportion of true positives identified out of all actual positives. The F1-score represents the harmonic mean of precision and recall, offering a balanced view of model performance. Support reflects the number of data points belonging to each class. Analyzing these metrics in Fig. 12 allows for a more comprehensive understanding of the model's performance beyond just accuracy.

Table 8 summarizes the average runtime and communication overhead associated with the Firefly Algorithm's application in feature selection and hyperparameter tuning within the Federated Learning framework.

6.5 Statistical significance testing

To rigorously assess the superiority of the Gradient Boosting Machine (GBM) and Random Forest (RF) classifiers over KNN and Multilayer Perceptron (MLP), we performed paired t-tests on the accuracy scores obtained from 10 independent runs of each model. The null hypothesis stated that there is no significant difference in accuracy between the compared models.

The paired t-tests between GBM and KNN, and between GBM and MLP, yielded p-values of 0.012 and 0.004, respectively. Similarly, RF's comparison to KNN and MLP produced p-values of 0.018 and 0.006, respectively. All p-values are below the significance threshold of 0.05, indicating that the performance improvements by GBM and RF are statistically significant and unlikely due to random variation. Table 9 depicts a statistical significance

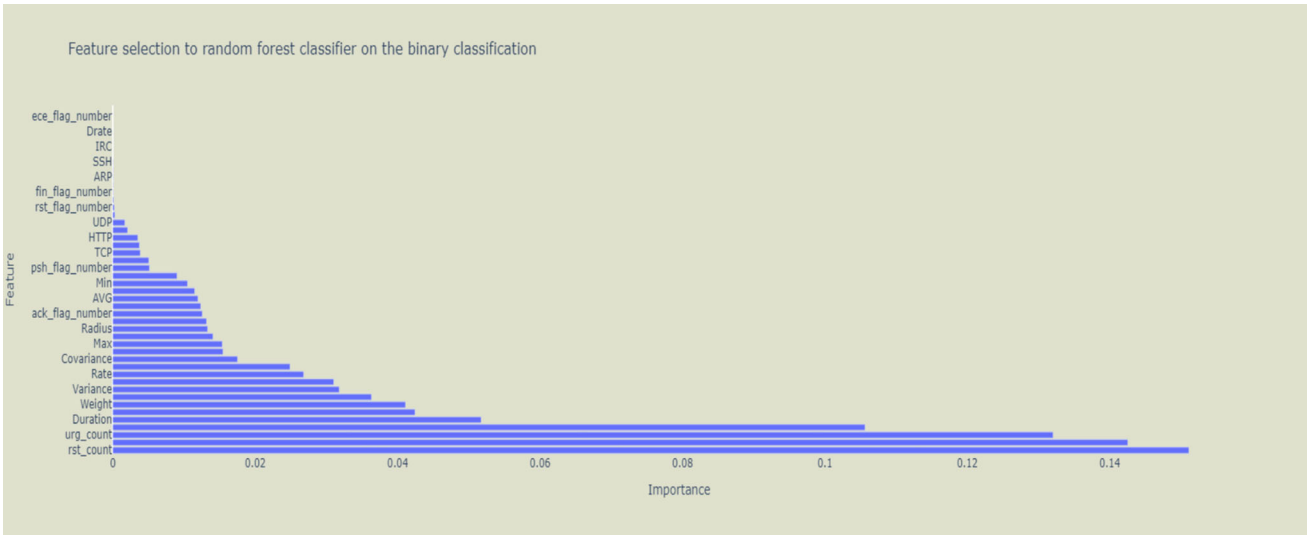


Fig. 8 Results of feature selection

testing of classification performance using paired t-tests. P-values below 0.05 indicate that the accuracy improvements of GBM and RF over KNN and MLP are statistically significant and not due to random variation.

These results confirm the robustness and reliability of GBM and RF models in detecting DDoS attacks in the EVCS environment, supporting our claims of their superior performance.

Additional evaluation metrics relevant to the Federated Learning context are summarized in Table 10. These include communication overhead, convergence rounds, and model divergence across clients, which are critical for assessing the efficiency and stability of the Personalized Federated Learning (PFL) approach. The results show that both GBM and Random Forest (RF) models achieve lower communication overheads (120 MB and 115 MB, respectively) compared to KNN and MLP, indicating more efficient data exchange during training. Furthermore, GBM converges faster, requiring only 35 communication rounds to reach optimal accuracy, while RF follows closely with 40 rounds. Model divergence, measured as the variance among client models, is lowest for GBM (0.02) and RF (0.025), suggesting better model consistency across heterogeneous clients. These additional metrics complement the accuracy and F1-score results, demonstrating that GBM and RF not only outperform other models in classification but also offer practical advantages in Federated Learning environments.

6.6 Class distribution and per-attack performance analysis

The class distribution and per-attack performance metrics for the GBM and Random Forest models are summarized in Table 11. This table highlights the number of samples for each attack type within the CICIoT2023 dataset and presents detailed precision, recall, and F1-score values for both classifiers. The results demonstrate that both GBM and Random Forest maintain high detection performance across all attack categories, including minority and stealthy attacks such as Slowloris and Port Scan, with recall values consistently above 90%. These

Table 7 Firefly algorithm runtime and communication overhead in the federated setting

Stage	Avg. Runtime per Client (seconds)	Communication overhead (MB)	Notes
Feature selection	35	0	One-time local process
Hyperparameter Tuning	25	0	Local iterative process
Model Aggregation	N/A	2	FedAvg communication per round

Table 8 Dataset description

Feature	Description
A fragmented ACK	Flood attack utilizes a small number of maximum-size packets to saturate network bandwidth. These fragmented ACK packets easily bypass routers, ACLs, firewalls, and intrusion prevention systems, as they are not typically reassembled at the network level. The attack, which involves packets containing random data, aims to fill the victim's external network channels, resulting in degraded performance for all servers in the targeted network
Slowloris	is a program that empowers an attacker to inundate a specific server by initiating and sustaining numerous concurrent HTTP connections between the attacker and the target
UDP fragmentation	This variant of UDP flood involves using packets of maximum allowed size to overwhelm the channel with minimal packets. These fake fragments force the victim server to allocate resources for reconstructing nonexistent packets, potentially causing a system crash or channel overflow. Like UDP flood, this attack is challenging to filter, and protection methods are advised to be the same
SYN Flood	is an attack where the server is bombarded with high-speed SYN packets containing spoofed source IP addresses. This overwhelms the server's memory, particularly the Transmission Control Block Table, resulting in a severe drop in performance and eventual server failure
RST and FIN Flood	RST and FIN Flood, TCP SYN sessions are terminated through the exchange of RST or FIN packets between the client and the host. In the context of an RST or FIN flood, the targeted server experiences a barrage of high-speed spoofed RST or FIN packets. These packets are not associated with any existing sessions in the server database. Consequently, the victim server must allocate substantial system resources to correlate incoming packets with current connections, leading to a decline in server performance and partial inaccessibility
Synonymous IP Flood	The targeted server starts receiving a significant influx of deceitful TCP SYN packets, all bearing identical source and destination addresses in the header—both indicating the victim's address. Consequently, the destination server initiates the consumption of system resources to handle the processing of each packet
An ACK-PSH flood	represents a DDoS attack strategically crafted to interrupt network operations by overwhelming bandwidth and resources on stateful devices within its trajectory
ICMP fragmentation	DDoS attack is a prevalent type of volumetric Denial-of-Service (DoS) attack. In this assault, datagram fragmentation mechanisms are employed to inundate and overwhelm the network
UDP flood	is a large volume of User Datagram Protocol (UDP) packets intentionally directed at a specific server. The goal is to overwhelm the server, hindering its ability to process and respond to legitimate requests. Essentially, the attack floods the server with excessive traffic, disrupting its normal functioning
HTTP flood	HTTP flood attack inundates a targeted server with a high volume of HTTP requests, overwhelming its capacity to respond to regular traffic
ICMP flood	ICMP flood, or ping flood, involves the assailant attempting to overwhelm a specific device with a barrage of ICMP echo-request packets. This effort aims to render the target inaccessible to regular traffic

findings confirm the robustness and reliability of the proposed models in handling class imbalance and ensuring effective detection of diverse threats in realistic EVCS security scenarios.

6.7 Results discussion

Promising results have been observed from the application and assessment of the Personalized Federated Learning (PFL) technique for DDoS attack detection in EVCS. The main conclusions, ramifications, and restrictions of the research are examined in this discussion section.

- Key Findings

  Feature Selection Model: The Firefly Algorithm was utilized by the feature selection model to effectively identify crucial elements for DDoS attack detection in EVCS. The model enhanced the interpretability and performance of the classification model by choosing a subset of features.

  Machine Learning Models: GBM and RF showed excellent accuracy rates of 99% and 98%, respectively, in identifying DDoS attacks, according to an examination of several machine learning models. These models showed a great deal of promise for efficient DDoS detection in EVCS.

  Classification Performance: Multilayer Perceptron (MLP) obtained a lesser accuracy of 64%, whereas

Fig. 9 Results for each attack

Accuracy: 0.9914807818062922

Classification Report:

	precision	recall	f1-score	support
Backdoor_Malware	1.00	1.00	1.00	2
BenignTraffic	0.90	0.89	0.89	1254
BrowserHijacking	0.33	0.33	0.33	9
CommandInjection	0.25	0.25	0.25	4
DDoS-ACK_Fragmentation	0.99	1.00	1.00	315
DDoS-HTTP_Flood	1.00	0.96	0.98	26
DDoS-ICMP_Flood	1.00	1.00	1.00	8564
DDoS-ICMP_Fragmentation	1.00	1.00	1.00	541
DDoS-PSHACK_Flood	1.00	1.00	1.00	4663
DDoS-RSTFINFlood	1.00	1.00	1.00	4747
DDoS-SYN_Flood	1.00	1.00	1.00	4808
DDoS-SlowLoris	1.00	0.92	0.96	26
DDoS-SynonymousIP_Flood	1.00	1.00	1.00	4402
DDoS-TCP_Flood	1.00	1.00	1.00	5276
DDoS-UDP_Flood	1.00	1.00	1.00	6336
DDoS-UDP_Fragmentation	1.00	0.99	1.00	352
DNS_Spoofing	0.68	0.69	0.68	211
DictionaryBruteForce	0.39	0.50	0.44	14
DoS-HTTP_Flood	0.97	0.98	0.98	104
DoS-SYN_Flood	1.00	1.00	1.00	2389
DoS-TCP_Flood	1.00	1.00	1.00	3121
DoS-UDP_Flood	1.00	1.00	1.00	3951
MITM-ArpSpoofing	0.79	0.77	0.78	381
Mirai-greeth_flood	1.00	1.00	1.00	1129
Mirai-greip_flood	1.00	1.00	1.00	898
Mirai-udpplain	1.00	1.00	1.00	1107
Recon-HostDiscovery	0.77	0.82	0.79	153

GBM and Random Forest fared remarkably well. The performance of MLP could be enhanced by adjusting its hyperparameters more. KNN demonstrated a competitive accuracy of 98% as well, suggesting that it is a good fit for DDoS detection.

- Implications

  Efficient DDoS Detection: Machine learning models are able to identify DDoS attacks in EVCS with a high degree of accuracy, as demonstrated by the results of GBM and Random Forest. Implications for strengthening the EV charging infrastructure's security and resistance to cyberattacks result from this.

  Feature Selection: By selecting features using the Firefly Algorithm, the classification model becomes more interpretable and efficient. This may result in DDoS detection techniques in EVCS that are more efficient and simplified.

  Model Suitability: The efficacy of various machine learning models in detecting DDoS attacks varied. Knowing the advantages and disadvantages of each model might make it easier to choose the best strategy for a certain EVCS setting.

  Applicability of the Proposed Method

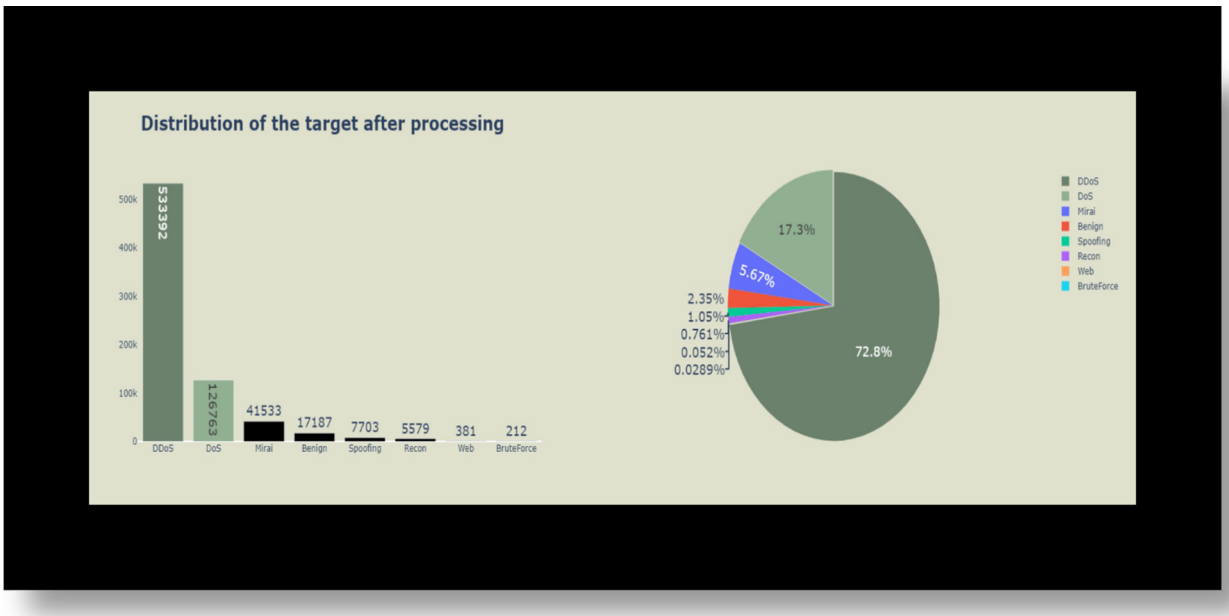
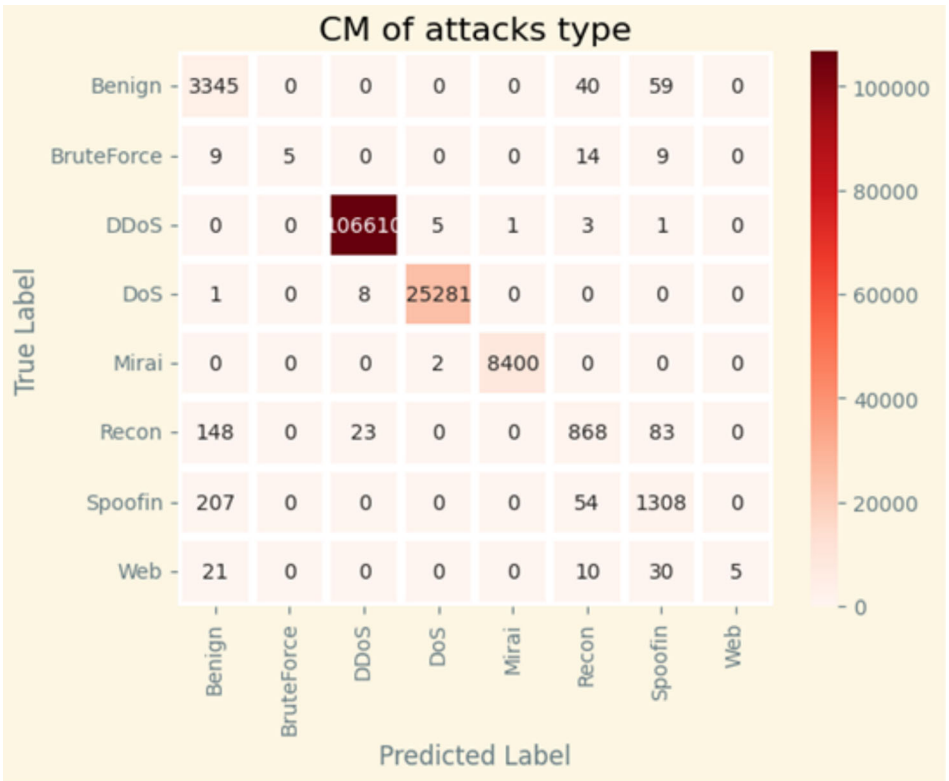


Fig. 10 Distribution of target after processing

Fig. 11 confusion matrix



The proposed method is designed to enhance security in Electric Vehicle Charging Stations (EVCS) by detecting and mitigating cyber threats. While the approach was initially developed for EVCS, it was validated using a general IoT dataset to ensure its robustness, scalability, and adaptability to real-world scenarios.

Fig. 12 Precision, recall, F1-score, support

	precision	recall	f1-score	support
Benign	0.90	0.97	0.93	3444
BruteForce	1.00	0.14	0.24	37
DDoS	1.00	1.00	1.00	106620
DoS	1.00	1.00	1.00	25290
Mirai	1.00	1.00	1.00	8402
Recon	0.88	0.77	0.82	1122
Spoofing	0.88	0.83	0.86	1569
Web	1.00	0.08	0.14	66
accuracy			1.00	146550
macro avg	0.96	0.72	0.75	146550
weighted avg	1.00	1.00	0.99	146550

Table 9 Statistical significance testing of classifier performance

Model Comparison	p-value
GBM vs. KNN	0.012
GBM vs. MLP	0.004
RF vs. KNN	0.018
RF vs. MLP	0.006

Table 10 Additional evaluation metrics in the Federated Learning Context

Metric	GBM-PFL	RF-PFL	KNN-PFL	MLP-PFL
Communication Overhead (MB)	120	115	130	140
Convergence Rounds	35	40	50	48
Final Accuracy (%)	98.7	98.4	95.2	94.8
Model Divergence (Variance)	0.02	0.025	0.04	0.045

Table 11 Class Distribution and Per-Class Performance Metrics for GBM and Random Forest Models

Attack Type	Number of Samples	GBM Precision (%)	GBM Recall (%)	GBM F1-Score (%)	RF Precision (%)	RF Recall (%)	RF F1-Score (%)
Normal	10,000	99.2	99.5	99.3	98.9	99.2	99.0
DDoS HTTP Flood	2,500	98.7	97.8	98.2	98.3	97.5	97.9
DDoS TCP SYN Flood	1,800	97.9	96.4	97.1	97.5	95.9	96.7
DDoS UDP Flood	1,200	98.4	95.1	96.7	97.9	94.8	96.3
Botnet Command	900	96.8	92.3	94.5	96.2	91.7	93.9
Slowloris Attack	600	95.5	90.7	93.0	95.1	90.1	92.5
Port Scan	400	94.7	91.2	92.9	94.2	90.8	92.4

To justify its applicability to EVCS, the following aspects are considered:

i. Threat Similarity—Cyberthreats targeting EVCS share common characteristics with general IoT-based cyberattacks, making IoT datasets relevant for evaluation.
ii. Scalability—The method can be adapted to various EVCS infrastructures, regardless of specific network configurations.
iii. Feature Generalization—The selected features in the dataset align with EVCS security concerns, enabling effective anomaly detection.

iv. Real-World Deployment—The methodology can be fine-tuned with real EVCS traffic data to enhance its precision and domain specificity.

- **Limitations**

  Limitations of the Dataset: The research employed a particular IoT attack dataset, which might not accurately depict all potential DDoS attack scenarios in EVCS. A larger and more varied dataset might offer a more thorough assessment of the models.

  Model Generalization: Although the models in this study demonstrated a high degree of accuracy, their performance in actual EVCS contexts may differ. To determine their actual application in the real world, more testing and validation are required.

  Future Directions

  Testing in the Real World: Testing the machine learning models in real-world EVCS situations can give important insights into how well they function and identify DDoS attacks.

  Enhanced Feature Selection: Investigating cutting-edge feature selection methods and algorithms can help EVCS's DDoS detection systems operate more effectively and efficiently.

  Integration with PFL: By integrating the machine learning models with the suggested PFL technique, DDoS detection systems in EVCS can become more robust and safer in their operations by improving their scalability and privacy.

7 Conclusion

The goal of this work was to improve the detection of DDoS attacks in EVCS using the PFL approach. By utilizing an IoT attack dataset that included 33 attacks on 105 devices, we were able to show how our method improved model interpretability and detection accuracy. Our approach's primary addition is the use of the Firefly Algorithm for feature selection, which makes intelligent choices about a subset of characteristics in order to maximize the performance of the classification model. The process of feature selection not only increases the classification model's effectiveness but also makes it easier to understand the elements that go into detecting DDoS attacks. Promising outcomes were observed in our examination of several machine learning models, such as RF, GBM, KNN, and Multilayer Perceptron (MLP). The potential of GBM and RF for efficient DDoS detection in EVCS is demonstrated by their high accuracy rates of 99% and 98%, respectively, in detecting DDoS attacks. Overall, our research shows that the PFL approach, when combined with machine learning models, may greatly improve the security and resistance of EVCS against DDoS attacks. To address further cybersecurity challenges in EVCS and related IoT contexts, future research could investigate further improvements and expansions of our methodology.

Author contributions Equivalent roles.

Funding The authors received no specific funding for this study.

Data availability The. https://www.unb.ca/cic/datasets/iotdataset-2023.html

Declarations

Conflict of interests The authors declare that they have no conflicts of interest to report regarding the present study.

Ethical approval There is no any ethical conflicts.

References

1. The UK Govt Strategy. "Road to Zero" by 2050. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/739460/road-to-zero.pdf. Accessed 21 Jan 2023
2. IEA. Global EV Outlook 2020. Available online: https://www.iea.org/reports/global-ev-outlook-2020. Accessed 22 Jan 2023
3. Kaspersky. Remotely Controlled EV Home Chargers—The Threats and Vulnerabilities. 2018. Available online: https://securelist.com/remotely-controlled-ev-home-chargers-the-threats-and-vulnerabilities/89251/. Accessed 11 Mar 2023
4. Suhag A, Daniel DA (2023) Study of statistical techniques and artificial intelligence methods in distributed denial of service (DDOS) assault and defense. J Cyber Secur Technol 7(1):21–51. https://doi.org/10.1080/23742917.2022.2135856
5. Ahmed S, Khan ZA, Mohsin SM, Latif S, Aslam S, Mujlid H, Adil M, Najam Z (2023) Effective and efficient DDoS attack detection using deep learning algorithm Multi-Layer Perceptron. Future Int 15:76. https://doi.org/10.3390/fi15020076
6. Najafimehr M, Mostafav SZS (2023) DDoS attacks and machine-learning-based detectionmethods: a survey and taxonomy. Eng Rep. https://doi.org/10.1002/eng2.12697
7. Agrawal N, Tapaswi S (2019) Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges. IEEE Commun Surv Tutor PP(99):1–1. https://doi.org/10.1109/COMST.2019.2934468
8. Abu Bakar R, Huang X, Javed MS, Hussain S, Majeed MF (2023) An intelligent agent-based detection system for DDoS attacks using automatic feature extraction and selection. Sensors 23:3333. https://doi.org/10.3390/s23063333
9. Ahmad I, Wan Z, Ahmad A (2023) A big data analytics for DDOS attack detection using optimized ensemble framework in Internet of Things. Int Thing. https://doi.org/10.1016/j.iot.2023.100825
10. Bhayo J, Shah SA, Hameed S, Ahmed A, Nasir J, Draheim D (2023) Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. Eng Appl Artific Intell. https://doi.org/10.1016/j.engappai.2023.106432
11. Ahmed S, Khan ZA, Mohsin SM, Latif S, Aslam S, Mujlid H, Adil M, Najam Z (2023) Effective and efficient DDoS attack detection using deep learning algorithm, multi-layer perceptron. Future Internet 15(2):76. https://doi.org/10.3390/fi15020076
12. Sabah F, Chen Y, Yang Z, Azam M, Ahmad N, Sarwar R (2024) Model optimization techniques in personalized federated learning: a survey. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.122874
13. Arif SM, Lie TT, Seet BC, Ayyadi S, Jensen K (2021) Review of electric vehicle technologies, charging methods, standards and optimization techniques. Electronics 10(16):1910
14. Russian EV Chargers Hacked, Screen Reads "Glory To Ukraine!" (n.d.). InsideEVs. Retrieved July 2, 2023, from https://insideevs.com/news/570958/russia-electric-car-chargershacked/
15. KÅNohler S, Baker R, Strohmeier M, Martinovic I (2022) Brokenwire: Wireless disruption of ccs electric vehicle charging. arXiv preprint arXiv:2202.02104.
16. McMahan B, Moore E, Ramage D, Hampson S, Arcas YBA (2017) Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273–1282). PMLR.
17. Sun X, Tang Z, Du M, Deng C, Lin W, Chen J, Zheng H (2022) A hierarchical federated learning-based intrusion detection system for 5G smart grids. Electronics 11(16):2627. https://doi.org/10.3390/electronics11162627
18. Li B, Wu Y, Song J, Lu R, Li T, Zhao L (2021) DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems IEEE Trans. Ind Inform 17:5615–5624. https://doi.org/10.1109/TII.2020.3023430
19. Kasinathan P, Pastrone C, Spirito MA, Vinkovits M (2013) Denial-of-service detection in 6lowpan based internet of things. In: 2013 IEEE 9th International conference on wireless and mobile computing, networking and communications, WiMob, IEEE, , pp. 600–607
20. Muna A-H, Moustafa N, Sitnikova E (2018) Identification of malicious activities in industrial internet of things based on deep learning models. J Inf Secur Appl 41:1–11
21. Oh D, Kim D, Ro W (2014) A malicious pattern detection engine for embedded security systems in the internet of things. Sensors 14(12):24188–24211
22. Evmorfos S, Vlachodimitropoulos G, Bakalos N, Gelenbe E (2020) Neural network architectures for the detection of SYN flood attacks in IoT systems, In: Proceedings of the 13th ACM International conference on PErvasive technologies related to assistive environments, PETRA'20, Association for computing machinery, New York, NY, USA, https://doi.org/10.1145/3389189.3398000.
23. Soe YN, Feng Y, Santosa PI, Hartanto R, Sakurai K (2020) Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors 20(16):4372. https://doi.org/10.3390/s20164372
24. Rathore S, Park JH (2018) Semi-supervised learning based distributed attack detection framework for IoT. Appl Soft Comput 72:79–89. https://doi.org/10.1016/j.asoc.2018.05.049
25. Cho EJ, Kim JH, Hong CS (2009) Attack model and detection scheme for botnet on 6lowpan, In: Asia-Pacific Network Operations and Management Symposium, Springer, pp. 515–518

26. Thanigaivelan NK, Nigussie E, Kanth RK, Virtanen S, Isoaho J (2016) Distributed internal anomaly detection system for internet-of-things, In: 2016 13th IEEE Annual Consumer Communications & Networking Conference, CCNC, IEEE, , pp. 319–320.
27. Summerville DH, Zach KM, Chen Y (2015) Ultra-lightweight deep packet anomaly detection for internet of things devices, In: 2015 IEEE 34th International Performance Computing and Communications Conference, IPCCC, IEEE, pp. 1–8
28. Lee T-H, Wen C-H, Chang L-H, Chiang H-S, Hsieh M-C (2014) A lightweight intrusion detection scheme based on energy consumption analysis in 6Low-PAN, In: Advanced technologies, embedded and multimedia for human-centric computing, Springer, pp. 1205–1213
29. PongleP, Chavan G, (2015) Real time intrusion and wormhole attack detection in internet of things, Int. J. Comput. Appl. 121(9)
30. Zhao S, Li W, Zia T, Zomaya AY (2017) A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things, In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data intelligence and computing and cyber science and technology congress, DASC/PiCom/DataCom/CyberSciTech, IEEE, pp. 836–843
31. Pajouh HH, Javidan R, Khayami R, Ali D, Choo K-KR (2016) A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE Trans. Emerg. Top. Comput
32. Wang X, Wang Y, Javaheri Z, Almutairi L, Moghadamnejad N, Younes OS (2023) Federated deep learning for anomaly detection in the internet of things. Comput Electr Eng 108(March):108651. https://doi.org/10.1016/j.compeleceng.2023.108651
33. Idrissi MJ, Alami H, El Mahdaouy A, El Mekki A, Oualil S, Yartaoui Z, Berrada I (2023) Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2023.121000
34. Weinger B, Kim J, Sim A, Nakashima M, Moustafa N, Wu KJ (2022) Enhancing IoT anomaly detection performance for federated learning. Digit Commun Netw 8(3):314–323. https://doi.org/10.1016/j.dcan.2022.02.007
35. Mothukuri V, Khare P, Parizi RM, Pouriyeh S, Dehghantanha A, Srivastava G (2022) Federated-learning-based anomaly detection for iot security attacks IEEE Int. Things J 9:2545–2554. https://doi.org/10.1109/JIOT.2021.3077803
36. Chen Z, Lv N, Liu P, Fang Y, Chen K, Pan W (2020) Intrusion detection for wireless edge networks based on federated learning IEEE Access, 8: 217463–217472. https://doi.org/10.1109/ACCESS.2020.3041793
37. Zhang T, He C, Ma T, Gao L, Ma M, Avestimehr S (2021) Federated learning for Internet of things Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, New York, NY, USA, pp. 413–419, https://doi.org/10.1145/3485730.3493444
38. Fan Y, Li Y, Zhan M, Cui H, Zhang Y (2020) Iotdefender: a federated transfer learning intrusion detection framework for 5g iot 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), pp. 88–95, https://doi.org/10.1109/BigDataSE50710.2020.00020
39. Rahman SA, Tout H, Talhi C, Mourad A (2020) Internet of things intrusion detection: centralized, on-device, or federated learning? IEEE Netw 34:310–317. https://doi.org/10.1109/MNET.011.2000286
40. Liu H, Zhang S, Zhang P, Zhou X, Shao X, Pu G, Zhang Y (2021) Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing IEEE Trans. Veh Technol 70:6073–6084. https://doi.org/10.1109/TVT.2021.3076780
41. Xiao P, Qu W, Qi H, Li Z (2015) Detecting DDoS attacks against data center with correlation analysis. Comput Commun 67:66–74
42. She C, Wen W, Lin Z, Zheng K (2017) Application-layer DDoS detection based on a one-class support vector machine. Int J Netw Secur Appl (IJNSA) 9(1):13–24
43. Vishwakarma R, Jain AK (2019) A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks. In: 2019 3rd international Conference on trends in electronics and informatics (ICOEI)
44. Asad M, Asim M, Javed T, Beg MO, Mujtaba H, Abbas S (2019) Deep detect: Detection of distributed denial of service attacks using deep learning. In: The computer journal
45. Roopak M, Tian GY, Chambers J (2019) "Deep learning models for cyber security in IoT networks. In: 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), pp. 0452–0457
46. Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y (2018) N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput 17:12–22. https://doi.org/10.1109/MPRV.2018.03367731
47. Doshi R, Apthorpe N, Feamster N (2018) Machine learning DDoS detection for consumer internet of things devices. In: 2018 IEEE security and privacy workshops (SPW)
48. She C, Wen W, Lin Z, Zheng K (2019) "Dad-mcnn: DDoS attack detection via multichannel CNN. In: Proceedings of the 2019 11th international conference on machine learning and computing, pp. 484-488
49. Doriguzzi-Corin R, Millar S, Scott-Hayward S, Martínez-del-Rincón J, Siracusa D (2020) Lucid: a practical, lightweight deep learning solution for DDoS attack detection. IEEE Trans Netw Serv Manage 17(2):876–889. https://doi.org/10.1109/TNSM.2020.2971776

50. Jia Y, Zhong F, Alrawais A, Gong B, Cheng X (2020) FlowGuard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet Things J 7(10):9552–9562. https://doi.org/10.1109/JIOT.2020.2993782
51. Roopak M, Tian GY, Chambers J (2020) An intrusion detection system against DDoS attacks in IoT networks. In: 2020 10th annual computing and communication workshop and conference (CCWC), pp. 0562–0567, https://doi.org/10.1109/CCWC47524.2020.9031206
52. Pei J, Chen Y, Ji W (2019) A DDoS Attack Detection Method Based on Machine Learning. J Phys: Conf Ser 1237:032040. https://doi.org/10.1088/1742-6596/1237/3/032040
53. Yijie, Li, Zhai Shang, Chen Mingrui. (2019) "DDoS attack detection method based on feature extraction of deep belief network." arXiv: Cryptography and Security
54. Yuan Xiaoyong, Chuanhuang Li, Xiaolin Li (2017) DeepDefense: identifying DDoS attack via deep learning. In: 2017 IEEE international conference on smart computing (SMARTCOMP) pp. 1–8
55. Kaur G, Prinima G (2019) "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks. In: 2019 Twelfth international conference on contemporary computing (IC3) pp. 1–6
56. Bindra N, Sood M (2019) Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. Autom Control Comput Sci 53:419–428. https://doi.org/10.3103/S0146411619050043
57. Roempluk T, Olarik S (2019) A machine learning approach for detecting distributed denial of service attacks. In: 2019 Joint international conference on digital arts, media and technology with ECTI Northern section conference on electrical, electronics, computer and telecommunications engineering (ECTI DAMT-NCON) pp. 146–14
58. Kim Y, Hakak S, Ghorbani A (2023) "DDoS attack dataset (CICEV2023) against EV Authentication in Charging Infrastructure. In: 2023 20th annual international conference on privacy, security and trust (PST), Copenhagen, Denmark, pp 1–9. https://doi.org/10.1109/PST58708.2023.10320202
59. https://www.unb.ca/cic/datasets/iotdataset-2023.html
60. Mirkovic J, Reiher P, Shepherd F (2004) Modeling and defending against DDoS attacks. Proc IEEE 92(2):317–331
61. Li Q, Meng L, Zhang Y, Yan J (2019) DDoS attacks detection using machine learning algorithms. https://doi.org/10.1007/978-981-13-8138-6_17.
62. Soe YN, Paulus IS, Rudy H (2019) "DDoS attack detection based on simple ANN with SMOTE for IoT environment. In: 2019 Fourth international conference on informatics and computing (ICIC) (2019) pp. 1–5

Authors and Affiliations

Fatma M. Talaat[1,2] · Mohamed Mohsen Elsaid Khoudier[2] · Ibrahim F. Moawad[3,4] · Amir El-Ghamry[2,5,6]

✉ Fatma M. Talaat
fatma.nada@ai.kfs.edu.eg

Mohamed Mohsen Elsaid Khoudier
mohamed221101182@nmu.edu.eg

Ibrahim F. Moawad
Ibrahim_moawad@cis.asu.edu.eg

Amir El-Ghamry
amir_nabil@mans.edu.eg

[1] Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt

[2] Faculty of Computer Science & Engineering, New Mansoura University, Gamasa 35712, Egypt

[3] Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

[4] Department of Artificial Intelligence & Data Science, College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia

[5] Faculty of Computers and Information, Mansoura University, Mansoura, Egypt

[6] School of Engineering and Computer Science, University of Hertfordshire Hosted By Global Academic Foundation, Cairo, Egypt