Enhanced IoT Security Using Machine Learning Technology

Rawan Yousef Bukhowah, Alanoud Khaled Bu Dookhi, Mounir Frikha
Department of Computer Networks and Communications-College of Computer Sciences and Information Technology,
King Faisal University, Al-Ahsa 31982, Saudi Arabia

***Abstract*—This paper examines the enhancement of security measures for the Internet of Things (IoT) systems through the application of Machine Learning (ML) techniques. As the number of IoT devices continues to rise, ensuring their security has become increasingly critical, given that conventional methods frequently struggle to identify advanced threats. This study explores the implementation of several ML algorithms, including Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN), to identify anomalies and intrusions within IoT networks. By conducting a comprehensive review of existing research and experiments, it highlights the effectiveness of ML in enhancing IoT security, with high detection rates for various threats, including botnet attacks, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) incidents, and intrusion attempts. DoS/DDoS attacks and many types of botnets are the most devastating attacks that have been spreading for a long time, and they are still branching out in new ways against IoT networks. They can damage IoT services and prevent these services from being used by legitimate users. Therefore, securing IoT networks becomes a significant concern. The proposed model is used to increasingly monitor network traffic for any deviations from standard patterns IoT networks. This paper also stresses the necessity of utilising suitable datasets and feature selection techniques to enhance the efficacy of ML models. To train our model, we have utilized a dataset called the IoT23 dataset, which is one of the most recent datasets that has many IoT scenarios and anomalous activities. Furthermore, we utilised two types of feature selection algorithms, the Correlation-based Feature Selection (CFS) algorithm and the Genetic Algorithm (GA), and then we compared the results of these algorithms when training our model. The best performances were obtained with DT and RF classifiers when they were trained with features selected by CFS However, for training and testing times metrics, DT performance was superior across both feature selection methods.**

Keywords*—*Internet of Things; Artificial Intelligence; machine learning; deep learning; security

I. Introduction

The Internet of Things (IoT) is an innovative technology that integrates various solutions across a wide range of applications [1], [2]. IoT refers to a distributed network comprising multiple sensor devices and systems, such as QR code devices, RFID devices, and sensor networks [1]. However, with the rapid growth and easy accessibility of these smart devices and networks, the focus on IoT security has increased due to malicious users, leading to a surge in related challenges [2], [3]. Furthermore, traditional techniques are less effective at detecting various types of attacks [3], [4], which requires the development of new technological advancements [2]. Artificial Intelligence (AI) is a computing technology that improves the ability of the IoT to deliver advanced application services [5]. IoT devices can leverage AI technology, encompassing two types: Machine Learning (ML) and Deep Learning (DL), each with its own set of algorithms and classifiers [6], [5]. IoT employs a multilayer architecture, with one architecture relying on three layers: application, network, and perception layers [1].

A. Problem Statement

In today's world, the lack of IoT security in systems, devices, and networks against attacks and malicious actions by intruders is a significant concern. Despite this, it is challenging to secure IoT devices with traditional security controls, and existing solutions such as the cloud suffer from centralisation and high delay. Another contributing factor is the vulnerability of IoT devices. Furthermore, the lack of security standards has added another dimension to the complexity of securing IoT devices. These challenges call for a monitoring system, such as anomaly detection, at both the device and network levels beyond the organisational boundaries. In recent years, the use of ML techniques has increased, and it is used to develop anomaly-based approaches to protect IoT networks, systems, and applications. It has impressive results while trained on normal and abnormal data in order to detect anomalies. However, building effective and efficient anomaly detection modules is not easy at all. ML models require significant effort, and selecting the best-performing algorithms remains a major challenge. Selecting the best dataset for a system is another challenge because there are many datasets that are available in public, and it isn't easy to choose the best one that fits your needs. Some of the data sets are large and outdated, so in this research, we aimed to enhance IoT security using ML techniques that can detect and mitigate cyber threats over a suitable data set for the IoT environment.

B. Contribution

Our contributions in this paper are summarized as follows:

- Designing a robust anomaly detection framework for the IoT, which facilitates and improves the detection of malicious data.
- Proposing an efficient model to extract relevant IoT features and eliminate unnecessary ones, thereby improving anomaly detection performance.
- Evaluating the effectiveness of the proposed model using recent IoT datasets and comparing it with existing ML models used for anomaly detection.

C. Research Questions

- What research has been done on ML for IoT security?
- What methods have been tried and how effective were they?
- What are the limitations or challenges of previous studies?
- How can ML improve threat detection in IoT networks?
- How can feature selection improve ML-based IoT security models?
- What are the most suitable datasets to enhance IoT security using ML algorithms?
- What are the research gaps and future directions in IoT security?
- What is the impact of processing stages-include data preprocessing, feature selection, and sampling-on evaluating the classifier performance in detecting attack types?

This paper is organized as follows. Section II presents the background. Section III reviews previous research studies related to IoT security using ML technology or presents and analyzes the related work. Section IV presents the methodology. Section V discusses the producers of the model setup, including preparing the dataset and training the model with classification algorithms. Section VI presents and analyzes the experimental results. Section VII presents a comparison between our proposed approach and an existing study. Section VIII concludes the research and suggests future work.

II. Background

A. IoT Security Challenges

1) Common security challenges include:

- Weak authentication and access control → Many IoT devices have default passwords or weak credentials.
- Intrusion and malware attacks → IoT networks are vulnerable to cyber threats, such as Distributed Denial-of-Service (DDoS), botnets, and ransomware.
- Data breaches and privacy issues → Sensitive user data (e.g. health or financial data) can be intercepted.
- Lack of real-time threat detection → Traditional security methods (e.g. firewalls, rule-based detection) struggle to detect new and evolving threats in real time.
- Resource constraints in IoT devices → Many IoT devices have limited processing power, making complex security algorithms difficult to implement.

B. AI Technology for IoT Security

Artificial Intelligence (AI), including ML and DL, is an effective solution to increase system security with regard to IoT. Traditional solutions require more processing ability for large datasets and are less efficient than AI technology. The remarkable ability of learning models to solve long-standing problems has made them increasingly important in recent times. Another significant advantage of learning algorithms is their ability to improve accuracy through real-time learning from test data. This section reviews various ML algorithms, how they function, and their contributions to IoT network security.

C. Machine Learning in IoT Security

Recent research shows that ML models significantly improve intrusion detection, anomaly detection, and malware classification. Studies have explored Convolutional Neural Networks (CNNs) for traffic classification, Long Short-Term Memories (LSTMs) for sequential anomaly detection, and Federated Learning for decentralised security. However, a trade-off between accuracy and computational efficiency remains a challenge.

1) Role of machine learning in IoT security: ML has emerged as a promising solution for securing IoT environments. ML-based security models can detect, predict and respond to cyber threats by learning patterns in network traffic. Unlike traditional security approaches, ML can handle large datasets, adapt to evolving threats, and minimize false positives. Various ML techniques have been applied in IoT security, including supervised learning, unsupervised learning, and deep learning models.

2) Key advantages of ML in IoT security:

- Anomaly Detection: ML can detect suspicious patterns in IoT traffic and identify potential cyberattacks.
- Adaptive Security: Unlike rule-based systems, ML continuously learns and updates itself to detect new threats.
- Low False Positive Rate: ML reduces false alarms compared to traditional signature-based detection.
- Scalability: ML can efficiently analyze large networks.
- Automated Threat Detection: ML models can work in real time, preventing attacks before they escalate.

Table I below briefly describes the K-means clustering and Q-Learning methods that belong to the RL model.

III. Related Work and Literature Analysis

This section reviews previous research on IoT security using ML. The studies analysed focus on various ML models, datasets, and techniques to enhance IoT security.

Nazir et al. [6] proposed a novel Collaborative Threat Intelligence Framework that integrates blockchain and ML for enhanced IoT security. The study used Random Forest (RF), Decision Trees (DT), Ensemble Learning, LSTM, and CNN models, using the IoT23 dataset. The primary issue addressed is the dynamic and evolving nature of IoT threats, which traditional security methods often fail to mitigate. The key contribution is a real-time threat detection system that improves accuracy by reducing false negatives through collaborative intelligence. Although the results showed improved

TABLE I. LEARNING MODELS

Learning model	Classifiers	Principle	Advantages	Disadvantages	Applications applicable to IoT
Supervised learning	SVM	Generates classified data by establishing a hyperplane between two or more classes. Expands the margin around the hyperplane.	• Stable in handling datasets with numerous features	• Inefficient with very large datasets • Struggles with multi-classification scenarios	• Intrusion detection • DoS/DDoS detection • Malware detection
Supervised learning	DT	Splits datasets into branches based on feature values, forming a tree structure. Branches represent decision rules ending in outcomes.	• Quick computations • Allows intuitive decision making	• Becomes overly complex with multi-dimensional data	• Intrusion detection • DoS/DDoS detection • Device authentication
Supervised learning	KNN	Assigns classifications based on the nearest sample votes.	• Straightforward and adaptable • Effective for both non-linear and multi-class scenarios	• Time-consuming to optimize K value	• Intrusion detection • DoS/DDoS detection
Supervised learning	ANN	Mimics a biological neural network using sample weights and biases.	• Handles incomplete data effectively • Capable of managing complex models due to hidden layers	• High computational demand • Challenges in tracing predictions	• Intrusion detection • DoS/DDoS detection • Malware analysis • Device authentication
Supervised learning	NB	Assumes independence among input variables	• Usually applicable for both small-scale data and large-scale data • Simple training process	• Independence assumption often not met	• Intrusion detection • Device authentication
Supervised learning	RF	Utilizes multiple decision trees, each trained on random data subsets.	• Better than single-model • Difficult to overfit • Excellent noise resilience • Works with high-dimensional data • Handles both continuous and discrete data	• Potential for overfitting • Degrades with noisy data	• Intrusion detection • DoS/DDoS detection
Unsupervised learning	PCA	Transforms correlated features into fewer uncorrelated ones.	• Simplifies models • No parameter constraints	• Less interpretable components • Sensitive to scaling and rotation	• Feature selection • Feature detection
Unsupervised learning	K-means Clustering	Groups samples into K clusters based on feature similarity.	• Efficient anomaly detection • Useful for data anonymisation	• May lack accuracy • Not ideal for complex data structures	• Network normalization • Sybil detection in WSNs • Private data anonymisation
Reinforcement learning	Q-Learning	Optimizes decisions in IoT through trial and error learning.	• Requires few parameters • Can be implemented offline	• High memory and time cost • Not ideal for high-dimensional data	• Intrusion detection • DoS/DDoS detection • Malware detection • Device authentication

accuracy and adaptability in IoT security applications, reliance on blockchain increased computational overhead.

Kirana et al. [7] developed an Intrusion Detection System (IDS) using Naïve Bayes (NB), SVM, DT, and AdaBoost classifiers. The testbed simulated IoT environments with ESP8266 and DHT11 sensors, collecting network traffic for ML model training. The study focused on detecting Man-in-the-Middle (MITM) attacks and demonstrated high accuracy in anomaly classification. However, adversarial attacks on ML models remain an unresolved challenge.

Alwahedi et al. [8] provided a comprehensive survey on ML-based Intrusion Detection Systems (IDS), including DL models such as Artificial Neural Networks (ANN), SVM, RF, and DT. The key issue tackled is the heterogeneity of IoT security threats. A significant gap identified is the lack of real-time response mechanisms in existing IDS solutions. The proposed solution integrates Generative AI and large language models (LLMs) for adaptive cyber threat mitigation.

Zawaideh and Abu Baker [9] introduced a cascaded ML approach for detecting botnet attacks in IoT networks. The study utilised a Recurrent Neural Network (RNN) combined with LightGBM, trained on the BOT-IoT dataset. The proposed model outperformed traditional classifiers with an accuracy of 99.9% and a recall of 100%, effectively handling imbalanced IoT traffic. The main limitation was the higher computational complexity of the cascaded model.

Bagaa et al. [10] developed an ML-driven security framework leveraging Software-Defined Networking (SDN) and Network Function Virtualisation (NFV). The framework employs One-Class SVM, Distributed Data Mining, and Neural Networks for real-time anomaly detection in innovative building IoT environments. Experimental results showed a 99.71% detection accuracy is achieved, although scalability issues in edge computing environments remain a challenge.

Al-Garadi et al. [3] presented a comprehensive review of ML and DL techniques for IoT security. The research

emphasised the effectiveness of DL models like CNNs and RNNs for detecting cyber threats. A key limitation highlighted is the lack of labelled datasets for IoT-specific threats, making supervised learning less effective in real-world scenarios.

Boukerche and Coutinho [11] outlined best practices for implementing ML-based security mechanisms in IoT networks. The study focused on optimising computational efficiency while maintaining high detection accuracy. The trade-off between detection accuracy and computational cost remains a primary challenge in real-world deployments.

Makkar et al. [12] introduced an ML-based spam detection framework for IoT devices. The study evaluated five ML models, using the REFIT Smart Home dataset. The best-performing models demonstrated high accuracy in detecting anomalous behavior, though the framework struggles with adversarial ML attacks.

Dhanke et al. [13] developed an ML-based IDS using RF, DT and SVM models. The study focused on Cyber-Physical Systems (CPS) and Smart Home environments, using ML to detect abnormal network traffic and malicious activities. The key findings revealed that ML improved security by detecting intrusions efficiently, achieving 85% accuracy. However, the study identified a gap in real-time adaptability, as some ML models struggled with evolving threats.

Gad et al. [14] introduced a distributed IDS leveraging RF, XGBoost, and LightGBM for anomaly detection in IoT networks. The study utilised the ToN-IoT dataset, which includes diverse cyberattacks. The key finding was that distributed ML models enhanced intrusion detection. However, high resource consumption on edge devices remained a limitation. The proposed lightweight IDS aimed to balance security and computational efficiency.

Ahmad and Almada [4] conducted a systematic literature review on the role of ML and DL techniques in IoT security. The review identified that supervised learning methods were more effective for known attack detection, while unsupervised learning performed better in zero-day attack scenarios. A key challenge highlighted was the scarcity of high-quality, labelled datasets, which limits the effectiveness of ML-based security models.

Amanullah et al. [5] investigated the use of DL for IoT security by analysing cyber threats using CNN and RNN. The research proposed a big data-driven approach for handling large-scale security threats. The study concluded that DL models significantly improved detection accuracy; however, the high computational overhead limited their deployment in low-power IoT devices.

Alrowais et al. [15] introduced an Automated Machine Learning (AutoML) approach using Mayfly Optimisation (MFO) with Regularised Extreme Learning Machine (RELM) for IoT security threat detection. The research used real-world IoT security datasets, demonstrating superior performance in detecting various attack types. The main limitation was the difficulty in real-time adaptation of AutoML models.

Awajan. [16] Introduced a DL-based IDS using a fully connected neural network. The study focused on detecting Blackhole, DDoS, and Sinkhole attacks with an average accuracy of 93.74%. A major challenge identified was the lack of communication protocol-independent security models, which limited interoperability across different IoT systems.

Tahir et al. [17] introduced an ML-based anomaly detection system to strengthen IoT security. The study employed RF, DT, SVM, and Gradient Boosting models to analyse IoT network traffic data. The key problem addressed was adaptive security mechanisms for anomaly detection in IoT environments. The research found that Gradient Boosting achieved the highest precision (89.34%), demonstrating the effectiveness of ML in securing IoT systems. However, a significant limitation identified was computational inefficiency, making real-time threat detection difficult in resource-constrained IoT environments.

Musleh et al. [18] proposed an IDS using feature extraction with ML algorithms. The study tested RF, KNN, SVM, and stacking models combined with feature extraction techniques such as VGG-16 and DenseNet. The dataset used was the IEEE Data Port, and the key finding was that VGG-16 with stacking achieved 98.3% accuracy, demonstrating the potential of DL for intrusion detection in the IoT. However, the reliance on large labelled datasets and computational overhead remains a significant challenge.

Khatun et al. [19] reviewed the application of ML in securing Healthcare Internet of Things (H-IoT). The study focused on intrusion detection, authentication, and anomaly detection using RF, DT, and Gradient Boosting models. The key finding was that ML algorithms significantly improve healthcare security, but privacy concerns and adversarial attacks on IoT devices remain a challenge. The study proposed a robust authentication mechanism using ML but highlighted that real-time data analysis remains computationally expensive.

Ahmad et al. [20] conducted a comprehensive review on ML and DL techniques for IoT security. The study highlighted the effectiveness of supervised models for known attack detection, while unsupervised learning was better suited for zero-day threats. A key challenge was the lack of labelled datasets, limiting the effectiveness of ML-based security models.

Mukherjee. I et al. [21] proposed a model of two different approaches to predict anomalies in IoT networks using a 350k-record dataset and deploying ML models. The study used Logistic Regression (LR), DT, RF, NB, and ANN algorithms first with all features, then with selected features after preprocessing. The models achieved 99.4% accuracy with all features and 99.99% after feature selection. It categorises attacks into DoS, malicious control, malicious operation, incorrect setup, and spying. The model was trained and tested on all attack types, rather than only common ones.

Bedi. P et al. [22] presented four models based on ML technology; these methods are ANN, LR, RF, SVM and DT. The paper aims to assess the results of all these models via evaluation metrics to find the best performance of the ML algorithm in detecting attacks in IoT sensor networks. The proposed IoT framework, structured by dataset perception, assortment and information processing, was executed on the dataset as an essential stage. Furthermore, the dataset used is known as the DS2OS dataset, which contains eight classes and seven attack types, including DoS attacks, data probing, malicious control, malicious operation and spying. The results showed that ANN and DT classifiers have better performance than LR, EF and SVM.

Altulaihan.E et al. [23] proposed a new mechanism using IDS to enhance the security of IoT networks. The paper focused on mitigating a type of attack called a DoS (Denial of Service) attack through four supervised classifier algorithms: KNN, SVM, DT, and RF, within an existing dataset known as the IoTD20 dataset, achieving an accuracy of 99.9%. Other datasets mentioned were CIDDS-001, UNSW-NB15, NSL-KDD, Bot-IoT, and KDD99. The results showed that DT and RF were the most efficient methods compared to others.

Alotabib et al. [24] Proposed a stacked DL approach for IoT security. This method aims to detect malicious traffic data, including phishing, DDoS, spamming campaigns, and data leakage. The proposed approach utilises pre-trained residual networks (ResNets), which help discover the characteristics of activities, whether normal or abnormal. The experiment has been done by using two cases using two datasets, including the N-BaIoT dataset and the power system dataset. Furthermore, the two scenarios target two heterogeneous IoT environments: smart homes and smart grids. NB, SVM and DT are ML algorithms that have been used for the proposed model.

Trcˇek.D et al. [25] presented a new anomaly detection system that is based on appropriate lightweight ML algorithms known as Profile and Hierarchical Incremental Clustering-based Anomaly Detection (PHICAD). The PHICAD enhances security, which works on the assumption of a large number of frequent flows that include normal activities and abnormal activities, and a small number of infrequent ones. Furthermore, it enables detection of unknown and new anomalies. Furthermore, the flows are arranged into profiles and each represents a model of network activity it assigned into a single network entity and it has it won IP address. This paper has been evaluate PHICAD by using CIC-IDS-2017 which contains most of network cybersecurity attacks (web attacks, brute force, port scan, heartbleed, DDoS, DoS, botnet and infiltration). Other ML algorithms performance have been compared such as KNN, RF, ID3 and NB using CIC-IDS-2017 dataset and result showed that the proposed approach achieved the highest precision rate.

A. Key Findings and Insights

Tables II and III presents a compilation of research studies that explore the application of ML techniques to enhance security in IoT environments.

In Tables II and III, we have highlighted several significant findings that enhance IoT security through the application of ML models. A key observation is the variety of ML models employed in various studies to tackle a broad spectrum of IoT security threats. These models range from traditional approaches such as DT, RF, SVM, KNN, and NB, to more sophisticated DL techniques including ANN, CNN, and RNN.

Additionally, some research incorporated hybrid or automated solutions like LightGBM, AutoML, and Generative AI. The effectiveness of each model varied depending on the specific security challenge—such as intrusion detection, DoS/DDoS attacks, botnet identification, and anomaly detection—indicating that a model's success is closely tied to the type of attack and the characteristics of the dataset. Furthermore, a notable insight is the consistent high detection accuracy reported across numerous studies. For example, Zawaideh and Abu Baker achieved 99.9% accuracy and 100% recall in identifying botnet attacks using a hybrid RNN and LightGBM method, while Bagaa et al. reported 99.71% accuracy in a smart building scenario using a One-Class SVM model. These results underscore the capability of well-trained ML models to provide reliable and accurate detection of cyber threats within IoT settings. However, despite these encouraging outcomes, challenges persist, particularly in realizing real-time detection and adaptability in dynamic, resource-limited IoT networks. As highlighted by Gad et al., the high computational requirements, especially at the edge level, can impede the deployment of distributed ML-based anomaly detection systems. This underscores the necessity for lightweight, adaptive models that can sustain performance in the face of evolving threats. The quality and selection of datasets are vital for the performance of models. Popular datasets like IoT23, BOT-IoT, and CIC-IDS-2017 offer extensive attack scenarios, but relying on outdated, unbalanced, or simulated data can limit the generalizability of studies. This highlights the need for choosing or creating datasets that accurately represent real-world IoT security situations. Feature selection also plays a crucial role in the success of models.

Many studies point out the advantages of employing feature selection techniques, such as PCA or correlation-based methods, to minimize dimensionality and enhance both accuracy and efficiency. Effective feature engineering is essential for optimizing model architecture and preventing overfitting. Furthermore, a significant issue that has arisen is the susceptibility of ML models to adversarial attacks. Malicious actors can alter inputs to trick models into making incorrect classifications. This necessitates the creation of robust models that are aware of adversarial threats and the incorporation of privacy-preserving strategies, particularly due to the sensitive information managed by IoT devices.

Finally, the balance between model complexity and computational efficiency continues to pose a challenge. Although DL models typically provide greater accuracy, their substantial resource requirements can hinder their use in real-world, low-power IoT environments. Hybrid strategies that merge lightweight traditional ML techniques with DL elements present a promising way to achieve a balance between detection quality and practical application.

Fig. 1 illustrates the accuracy percentages of different ML models across various studies as presented in Table II. Each segment of the pie represents the accuracy achieved in different studies, highlighting the performance of the models used in IoT networks.

After analyzing Table II, we suggest using ML models such as RF, DT, and CNN for IoT security. These models have shown strong performance in detecting a wide range of IoT security threats, including intrusion detection, DoS/DDoS attacks, botnet detection, and anomaly detection. RF and DT are particularly useful for their robustness, ease of implementation, and good performance on structured data. CNNs, on the other hand, are beneficial for more complex attack detection scenarios, particularly those involving network traffic and sequential data patterns.

In terms of datasets, we recommend using IoT23. This dataset provides a wide range of IoT-specific attack scenarios

TABLE II. SUMMARY OF EXISTING STUDIES ON ML TECHNIQUES IN IOT SECURITY

Reference	ML Models Used	Problem Addressed	Datasets Used	Results
Nazir et al. (2024) [6]	RF, DT, Ensemble Learning, LSTM, CNN	Evolving IoT threats and ineffective traditional security	IoT23 dataset	Enhanced real-time threat detection, reduced false negatives
Kirana et al. (2020) [7]	Naïve Bayes, SVM, DT, AdaBoost	MITM attack detection in IoT environments	Simulated IoT environment (ESP8266, DHT11 sensors)	High accuracy in detecting MITM attacks
Alwahedi et al. (2024) [8]	ANN, SVM, RF, DT	Heterogeneous IoT security threats and lack of real-time response	Not specified	Proposed Generative AI and LLMs for adaptive threat mitigation
Zawaideh and Abu Baker (2023) [9]	RNN, LightGBM	Botnet attack detection in IoT networks	BOT-IOT dataset	99.9% accuracy, 100% recall
Bagaa et al. (2020) [10]	One-Class SVM, Distributed Data Mining, Neural Networks	Real-time anomaly detection in smart building IoT	SDN/NFV-based IoT environments	99.71% detection accuracy
Al-Garadi et al. (2020) [3]	CNN, RNN	Cyber threat detection using ML/DL	Various IoT security datasets (review study)	DL models effective for IoT security
Boukerche & Coutinho (2020) [11]	ML-based security mechanisms	Optimizing computational efficiency while maintaining detection accuracy	Not specified	Trade-off between detection accuracy and computational cost remains a challenge
Makkar et al. (2020) [12]	Evaluated five ML models	Spam detection for IoT devices	REFIT Smart Home dataset	High accuracy in detecting anomalies, but vulnerable to adversarial ML attacks
Dhanke et al. (2021) [13]	RF, DT, SVM	Intrusion detection in CPS and Smart Homes	Not specified	85% accuracy, but struggled with evolving threats in real-time adaptability
Gad et al. (2022) [14]	RF, XGBoost, LightGBM	Distributed IDS for IoT anomaly detection	ToN-IoT dataset	Enhanced intrusion detection, but high resource consumption on edge devices
Ahmad & Alsmadi (2021) [4]	Supervised and Unsupervised Learning	ML and DL techniques in IoT security	Various datasets	Supervised learning best for known attacks, unsupervised better for zero-day attacks; lack of high-quality labeled datasets remains a challenge
Amanullah et al. (2020) [5]	CNN, RNN	Analyzing cyber threats in IoT using DL	Not specified	Improved detection accuracy, but high computational overhead limits deployment in low-power IoT devices
Alrowais et al. (2022) [15]	AutoML (MFO-RELM)	Automated ML for IoT security threat detection	Real-world IoT security datasets	High accuracy in detecting various attack types; difficulty in real-time adaptation
Awajan (2023) [16]	Fully connected neural network (DL-IDS)	Intrusion detection (Blackhole, DDoS, Sinkhole attacks)	Not specified	93.74% accuracy; challenge in developing protocol-independent security models
Tahir et al. (2024) [17]	RF, DT, SVM, Gradient Boosting	Adaptive anomaly detection in IoT security	IoT network traffic data	Gradient Boosting achieved 89.34% precision; computational inefficiency limits real-time detection
Musleh et al. (2023) [18]	RF, KNN, SVM, Stacking (VGG-16, DenseNet)	Feature extraction for IDS	IEEE Dataport	VGG-16 with stacking achieved 98.3% accuracy; challenges in dataset labeling and computational overhead
Khatun et al. (2023) [19]	RF, DT, Gradient Boosting	ML for Healthcare-IoT (H-IoT) security	Not specified	ML improves healthcare security; challenges in adversarial attacks and real-time computational efficiency
Ahmad et al. (2021) [20]	Supervised and Unsupervised Learning	ML and DL techniques for IoT security	Various datasets	Supervised models effective for known attacks, unsupervised better for zero-day threats; challenges in dataset availability
Mukherjee.I et al. (2020) [21]	LR, DT, RF, ANN, NB	Detect IoT threats	DS2OS traffic traces	Most classifiers achieved 99.9% accuracy except RF with 94%, indicating lower performance

and the latest real-world data, making it ideal for training and testing ML models in the context of IoT security. IoT23 is well-suited to detect a variety of network attacks and provides comprehensive coverage for anomaly detection and botnet identification. On the other hand, many mentioned papers ether used outdated datasets, lacked most recent attack types, complex dataset, lacked of IoT traces, and lacked contained features related to IoT.

These ML models and datasets, when combined, will enable more effective and robust detection of IoT security threats, enhancing the overall security posture of IoT networks.

IV. METHODOLOGY

ML classification algorithms were used to detect anomalies in IoT networks. This approach monitors network traffic and data flows for deviation from normal network profiles based on anomaly detection. This methodology consists of several key stages, as shown in Fig. 2. First is the processing phase, where the preferred dataset is selected to train ML classification algorithms. Preprocessing is then performed to clean and transform the data. Null and irrelevant values are removed, as they are difficult to handle and may lead to incorrect results. Second is the feature selection phase, which includes preprocessing the features. Third, the dataset is divided into two subsets: the training subset and the test subset. By selecting the right testing and training data, classification accuracy can be improved. The training data are used to train the model, while the test data evaluate the model's performance. Finally, classification identifies whether the input belongs to a normal class or a specific type of attack using selected ML algorithms. We will

TABLE III. SUMMARY OF EXISTING STUDIES ON ML TECHNIQUES IN IOT SECURITY

Reference	ML Models Used	Problem Addressed	Datasets Used	Results
Khatun et al. (2023) [19]	RF, DT, Gradient Boosting	ML for Healthcare-IoT (H-IoT) security	Not specified	ML improves healthcare security; challenges in adversarial attacks and real-time computational efficiency
Ahmad et al. (2021) [20]	Supervised and Unsupervised Learning	ML and DL techniques for IoT security	Various datasets	Supervised models effective for known attacks, unsupervised better for zero-day threats; challenges in dataset availability
Mukherjee I. et al. (2020) [21]	LR, DT, RF, ANN, NB	Detect IoT threats	Ds2OS traffic traces	All classifiers except RF achieved 99.9% accuracy; RF achieved 94%, indicating lower performance
Bedi P. et al. (2021) [22]	ANN, RF, LR, DT, SVM	Assessment of ML algorithms to select best for attack detection in IoT sensor networks	DS2OS dataset	DT and ANN classifiers achieved high precision rates around 99%
Altulaihan E. et al. (2024) [23]	DT, RF, KNN, SVM	Detection of DoS attacks in IoT networks	IoTID20 dataset	DT and RF achieved 100% detection performance
Alotabib B. et al. (2024) [24]	NB, DT, SVM	Cyberattack detection against IoT devices	Power system dataset and N-BaIoT dataset	RF achieved highest accuracy: 100% and 99.9%, respectively
Trček D. et al. (2024) [25]	PHICAD	New lightweight ML approach to detect various anomalies in IoT network	CIC-IDS-2017 dataset	Achieved 99% precision, 84% recall, and 91% F1-score

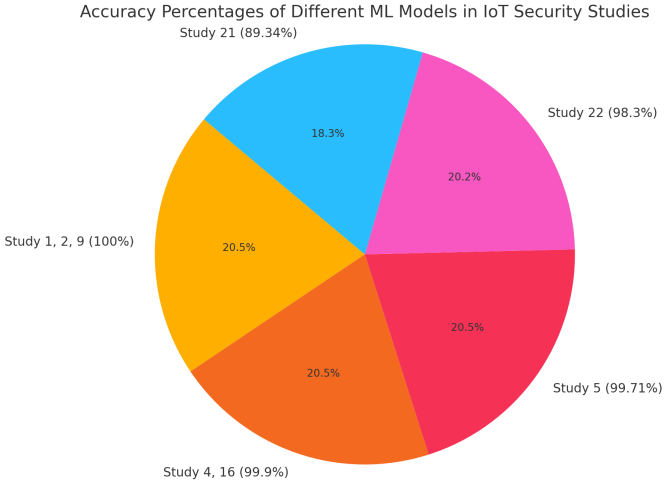

Fig. 1. The accuracy percentages of different ML models.

validate our proposed scheme using a confusion matrix.

A. Evaluation and Analysis

The evaluation and analysis process includes the following steps:

- Evaluate the performance of the IDS in simulated or real-world environments.
- Analyze the system's performance using metrics such as detection rate, false positive rate, and response time.

B. Optimization and Improvement

The optimization and improvement phase includes the following steps:

- Identify potential areas for improvement in the attack detection system based on the evaluation results.
- Conduct further research and development to optimize the system's performance and enhance its robustness against evolving attack techniques.

C. Identification and Analysis of Alternative Solutions

Multiple alternative solutions exist for implementing the system, and various methods can be used. In this section, we explain each solution.

1) Feature selection algorithms: Two different feature selection algorithms were used to improve detection accuracy and the speed of identifying malicious activities. The process involves both selecting the most relevant features and removing redundant and irrelevant features. Additionally, we compared the performance of Genetic Algorithm (GA) and Correlation-Based Feature Selection (CFS) methods.

- GA is a natural selection-based optimization technique that involves several steps. It begins with creating an initial population of possible features, followed by evaluating these subsets through a predictive model. Next, subsets are selected to continue based on tournament selection, where each member competes for survival. In the next generation, winners undergo crossover and mutation to form new subsets of features.
- CFS is a filter-based approach that evaluates the correlation between input and output features. It eliminates redundant or irrelevant features that could increase computation time or reduce detection accuracy.

Table IV provides a comparison of the advantages and disadvantages of both GA and CFS algorithms.

TABLE IV. ADVANTAGES AND DISADVANTAGES OF GA AND CFS ALGORITHMS

Algorithm	Advantages	Disadvantages
GA	• Finds good solutions quickly and accurately • Easy to implement	• Difficult parameter selection • May not yield optimal solutions
CFS	• Lower computational complexity than GA • Strong predictive power for gene selection • Less prone to overfitting	• May not fit data well due to model dependency

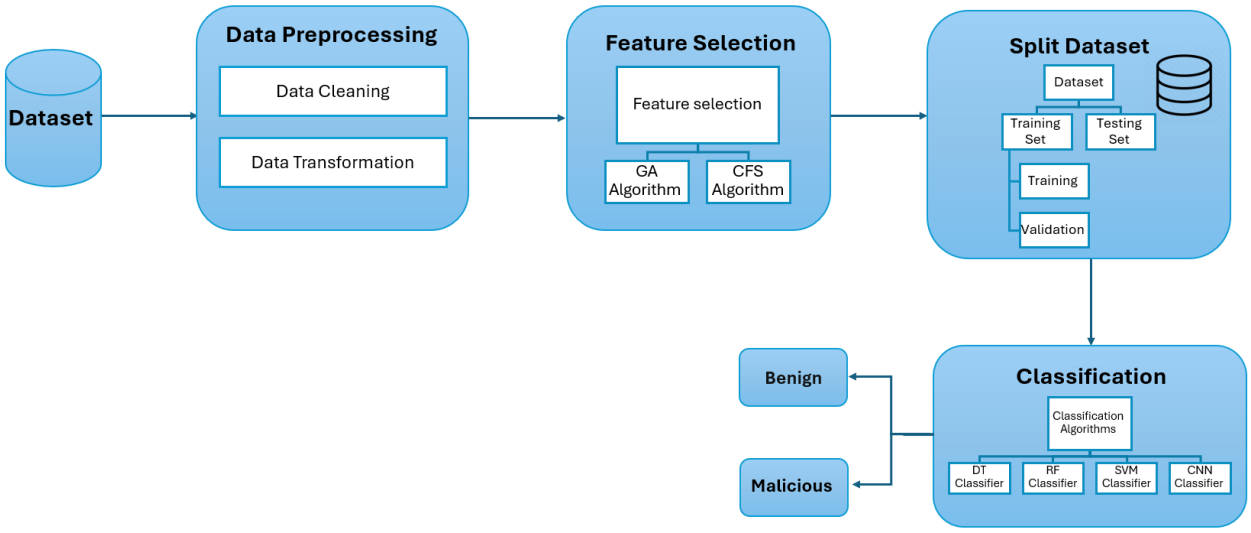

Fig. 2. Evaluation of the performances.

2) Machine learning algorithms solutions and selected dataset: We applied four machine learning algorithms to the IoT23 dataset: DT, RF, SVM, and CNN. In Table I, we clarify the advantages and disadvantages of the four classifiers.

3) Performance metrics: After implementing and validating the model, we conducted testing to assess its performance. The initial evaluation focuses on accuracy and timeliness in predicting data. Subsequently, the model's training and testing performance are evaluated using metrics such as accuracy, precision, and recall Performance metrics are systematic measures that help assess how well a tool performs. In ML, these metrics are used to evaluate the effectiveness of features, feature selection methods, and ML algorithms. Performance metrics play a crucial role in assessing the effectiveness of four types of ML algorithms: DT, RF, SVM, and CNN. These metrics help evaluate the classification performance of the feature set and different ML algorithms. Some fundamental metrics include True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These basic metrics serve as building blocks for calculating more advanced metrics such as precision, recall, accuracy, F-measure, and AUC. The key advanced metrics considered are precision, accuracy, recall, and F1-score.

- Accuracy: The accuracy formula is as follows.

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

- Precision: The precision formula is as follows.

$$\text{P}recision = \frac{TP}{TP + FP}$$

- Recall: The recall formula is as follows.

$$\text{R}ecall = \frac{TP}{TP + FN}$$

- F-measure: The F-measure formula is as follows.

$$\text{F} - measure = 2x\frac{precision + recall}{precisionxrecall}$$

V. Setup of the Model

To implement the IDS, the model is structured in two main stages: dataset preparation and training with classification algorithms.

A. Data Collection

The IoT23 dataset was selected for this study. It contains 23 PCAP files used in various use cases such as intrusion detection, malware analysis, traffic classification, and anomaly detection in IoT networks. The dataset contains 28 columns called features including duration, orig_bytes, resp_bytes, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes, and label. The model focuses on the 'label' feature, which indicates whether network traffic is benign or malicious.

1) Load dataset: To train the system for detecting malicious activity in the IoT network, we used the IoT23 dataset due to its comprehensive feature set. First, we downloaded the dataset from its official website and saved it locally. Then, we loaded the dataset by locating the path to our dataset file, storing it in the variable "path," and assigning the dataset to the data frame object "df" with the help of the Pandas library. Finally, we printed the dataset using the df.head() function.

B. Dataset Preprocessing

In this stage, we enhance the dataset and make it more specific by performing a series of manipulations, including cleaning, encoding, scaling, and noise removal.

1) Cleaning: In the data cleaning process, we removed null values and their corresponding entries. Missing values can be challenging to handle for ML algorithms, potentially leading to incorrect predictions[26]. We utilized the isnull().sum() function to check for null values.

2) Encoding: First, we used the "df['label'].value_counts()" function to determine how frequently benign and malicious packet traffic appeared. Second, as included in the dataset, it classified them as benign (normal) or attack types (Attack, PartOfAHorizontalPortScan, PartOfAHorizontalPortScan, C&C, C&C-FileDownload, C&C-HeartBeat, C&C-HeartBeat-FileDownload, C&C-Mirai, C&C-Torii, DDoS, FileDownload, Okiru, and PartOfAHorizontalPortScan). From these attack types, we selected only those relevant to our study—specifically intrusion, DDoS, and botnet attacks. Finally, we converted all labels to 1 or 0 using the "lambda" function and the "apply" method.

Our dataset is noise-free, as it was generated through the simulation of our system. Therefore, there is no need for noise removal steps.

C. Feature Selection

We must identify which features are important for the training and testing phases and eliminate those that are unnecessary. GA is an algorithm used for feature selection. The GA is an optimization algorithm using the wrapper method which starts with a set of points coded as an alphabet rather than one real parameter set. Additionally, there is no need for a calculation step size or derivative information in GA. When applied to our dataset, GA selected 14 out of 28 features as the most relevant for training the system.

CFS was also employed as an alternative feature selection method. Features are selected and analyzed based on their relationship with the target variable and their interrelationships. After executing the CFS algorithm, the top-ranked features were prioritized for application within our dataset.

D. Split the Dataset into Training and Testing Sets

The dataset was divided into two subsets: training and testing. This is a crucial step in the ML process, as it directly impacts the model's accuracy. The training data were used to fit the model, while the testing data evaluated its performance. Then, the testing data is used to evaluate the performance of the model. We used the train_test_split() function to divide the dataset into training and testing sets with a 67/33 split, using random_state=42 for reproducibility. We set the test size to 33% of the dataset and the remaining 67% was used for training. In addition, we used the "random_state" parameter with a value of 42 to ensure reproducibility. After splitting the dataset, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to baa lance the training data.

E. Training with Classification Algorithms

The most critical stage of our project involves training and testing the model using ML algorithms. Once the training and testing sizes were known, we worked with classification algorithms. We selected four classification algorithms—DT, RF, SVM, and CNN—based on their effectiveness of algorithms in distinguishing between benign and malicious traffic. This section compares and evaluates the four algorithms using performance metrics such as confusion matrix, accuracy, precision, recall, and F1-score. We then identified the best-performing algorithm among the four.

1) Decision tree: DT is a classification algorithm used to determine whether the traffic is an attack or not. It has a tree-like structure that is easy to understand and mimics human decision-making. It consists of internal decision nodes and leaf nodes for outcomes, working by recursively splitting the dataset using attribute selection measures. In this study, the DT model was trained using the GA algorithm with Scikit-Learn's DecisionTreeClassifier.

2) Random forest classifier training: The RF model was implemented by creating randomly selected subsets of the training data to generate multiple decision trees. The method is based on building multiple DTs from a randomly selected subset of the training set and then combining all votes for every tree to make a final prediction. Additionally, the RF model effectively identifies the class of each input instance. Each decision tree processes input data to determine the corresponding category. The RF generates multiple trees and then averages the predictions, which helps mitigate overfitting [26].

To implement the classifier, a random sample of size n is selected for building each decision tree. Two essential criteria are carried out: 1) m features are randomly selected from all features, and 2) the data is split based on the m features. A total of k trees are constructed, and the final prediction is obtained by averaging their outputs [26].

3) Support vector machine training: The SVM algorithm is primarily used for classification but can also handle regression problems and both continuous and categorical variables. SVM aims to separate classes by constructing a hyperplane in multidimensional space. Iterative SVM algorithms help minimize errors. The algorithm seeks to define a maximum marginal hyperplane to split the dataset into multiple classes. To implement SVM, we separate the dataset by identifying the margin—the distance between classes. This is known as the margin. The goal is to determine a hyperplane that offers the widest margin between support vectors [26].

We implemented the SVM model by importing it and then creating a support vector classifier object using an SVC(). The SVC() object includes the gamma parameter to control the influence of individual training examples. The model was trained using the train set with the fit() function and made predictions on the test set using the "predict()" function. After training, predictions were made on the test data, and the SVM model was compared with other classifiers.

4) Convolutional Neural Network (CNN): is a powerful DL algorithm, effective for both image classification and sequential or network traffic data like intrusion detection. CNN includes convolutional layers like Conv1D, which automatically extract important patterns from input data, followed by pooling and flattening layers. The dense layer, a key component, is responsible for performing the final classification. CNN was applied to enhance the detection of attacks within network traffic data.

In addition, by using learning deep feature representations we can improve classification accuracy.

5) Evaluate the performance: The evaluation phase was based on the confusion matrix. We compared the results of classifier algorithms with GA and CFS feature selection algorithms. Furthermore, to evaluate the classifier algorithms, we chose to use accuracy, precision, recall, and the F1 score.

VI. Results

This section presents and analyzes the experimental results, including the training and testing times of various classifiers, and evaluates their performance using key metrics such as accuracy, precision, recall, and F1-score.

A. Training and Testing Time

To evaluate the computational efficiency of the selected classifiers, we measured both training and testing times using two different feature selection techniques: GA and CFS. Table V and Fig. 3 present a comparative analysis of these times across the four classifiers: DT, RF, SVM, and CNN.

From the results, DT exhibited the fastest performance in both training and testing phases across both feature selection methods. Specifically, training with GA took only 0.6821 seconds, and testing took 0.0168 seconds. With CFS, the training and testing times increased slightly to 1.1308 and 0.0285 seconds, respectively.

RF demonstrated moderate computational costs, with GA yielding faster results compared to CFS. Notably, the training time for RF with GA was 33.4947 seconds, while testing took 1.1156 seconds. CFS led to slightly increased times at 43.9254 seconds (training) and 1.0211 seconds (testing).

SVM showed the highest variation in computational time. With GA, SVM required significantly higher resources, recording 5448.912 seconds for training and 1005.804 seconds for testing. However, with CFS, there was a remarkable reduction to 158.8664 seconds for training and 94.2506 seconds for testing.

On the other hand, CNN had a moderate-to-high training and testing time, with GA being faster than CFS. CNN took 188.9341 seconds for training and 6.1233 seconds for testing when GA was applied. Under CFS, training time increased to 387.9729 seconds, while testing time rose to 24.252 seconds.

In Fig. 3, GA generally yielded lower testing times, especially for DT and RF, making it suitable for time-sensitive environments. However, CFS demonstrated better training time efficiency for high-complexity models like SVM, highlighting its utility in scenarios where training overhead needs to be minimized.

From these results, we can conclude that the best-performing classifier in terms of both training and testing time is the DT algorithm. It consistently achieved the lowest execution times across both GA and CFS feature selection methods, making it an ideal choice for real-time IoT anomaly detection scenarios where computational efficiency is critical. This finding highlights the practicality of DT for deployment in lightweight and resource-constrained IoT environments.

TABLE V. Training and Testing Times (seconds) for Different Classifiers

Classifier	Training Time (s)		Testing Time (s)	
	GA	CFS	GA	CFS
DT	0.6821	1.1308	0.0168	0.0285
RF	33.4947	43.9254	1.1156	1.0211
SVM	5448.912	158.8664	1005.804	94.2506
CNN	188.9341	387.9729	6.1233	24.252

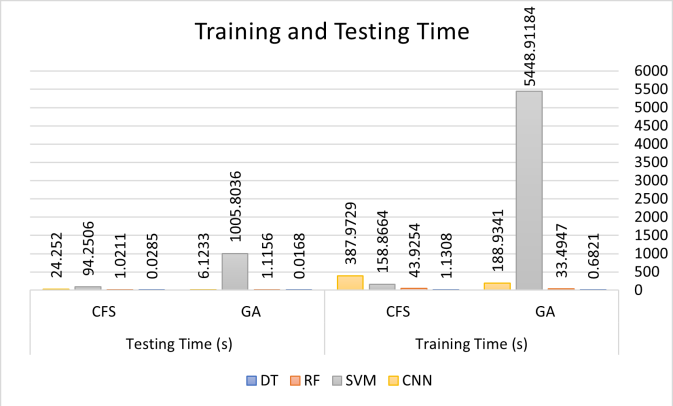

Fig. 3. Training and testing times.

1) Evaluate the performance: This section evaluates the performance of the classifiers using key metrics: Accuracy, Precision, Recall, and F1-score. The results are expressed in percentages for clarity, and both GA and CFS were applied for feature selection.

a) Accuracy: Accuracy evaluates the overall precision of the classifier. As illustrated in Table VI and Fig. 4, both the DT and RF attained the highest accuracy of 99.9% when utilizing features selected by CFS. SVM and CNN also exhibited remarkable performance, increasing from 98.6% and 97.9% (with GA) to 99.7% (with CFS), respectively. These findings validate the efficacy of the CFS method in improving overall classification accuracy.

TABLE VI. Accuracy Results

Classifier	With GA	With CFS
DT	0.993	0.999
RF	0.993	0.999
SVM	0.986	0.997
CNN	0.979	0.997

b) Precision: Precision assesses the ratio of accurate positive identifications. As illustrated in Table VII and Fig. 5, DT and RF achieved the highest precision rate of 99.9% when utilizing CFS. SVM and CNN also exhibited enhancements, increasing to 99.5% with CFS, in contrast to their previous rates of 98.5% and 98.6% with GA. This indicates that the models improved in their ability to reduce false positives when employing CFS.

c) Recall: Recall reflects the model's capacity to identify all genuine instances of attacks accurately. As shown in Table VIII and Fig. 6, every classifier attained a recall rate of 99.9%, except for the CNN model utilizing GA-selected

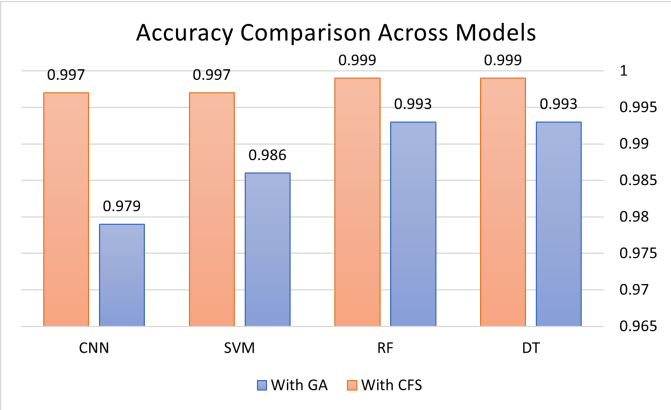

Fig. 4. Accuracy results.

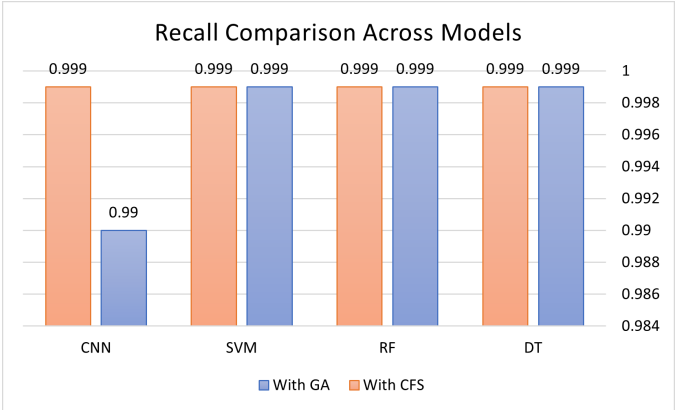

Fig. 6. Recall results.

TABLE VII. PRECISION RESULTS

Classifier	With GA	With CFS
DT	0.993	0.999
RF	0.993	0.999
SVM	0.985	0.995
CNN	0.986	0.995

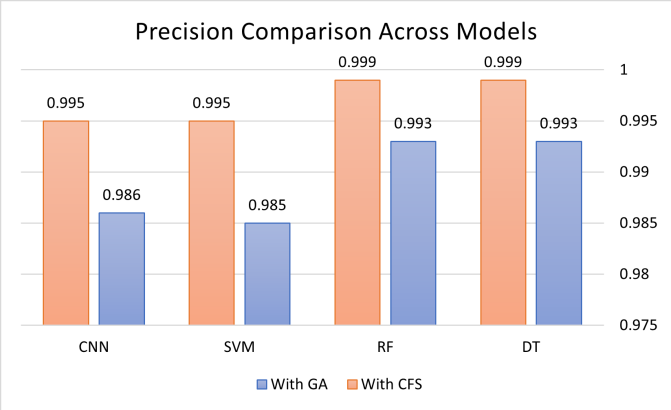

Fig. 5. Precision results.

features, which experienced a minor decrease to 99.0%. The consistently elevated recall values indicate that the models are exceptionally proficient in recognizing actual attacks.

TABLE VIII. RECALL RESULTS

Classifier	With GA	With CFS
DT	0.999	0.999
RF	0.999	0.999
SVM	0.999	0.999
CNN	0.990	0.999

d) F1 Score: The F1-score represents the harmonic mean of precision and recall, establishing a balance between these two metrics. As demonstrated in Table IX and Fig. 7 DT and RF achieved the highest F1-score of 99.9% when utilizing CFS, whereas SVM and CNN attained scores of 99.7%, an increase from 99.2% and 98.8% with GA, respectively. These findings validate the robustness of the models, especially when employing CFS.

TABLE IX. F1 SCORE RESULTS

Classifier	With GA	With CFS
DT	0.996	0.999
RF	0.996	0.999
SVM	0.992	0.997
CNN	0.988	0.997

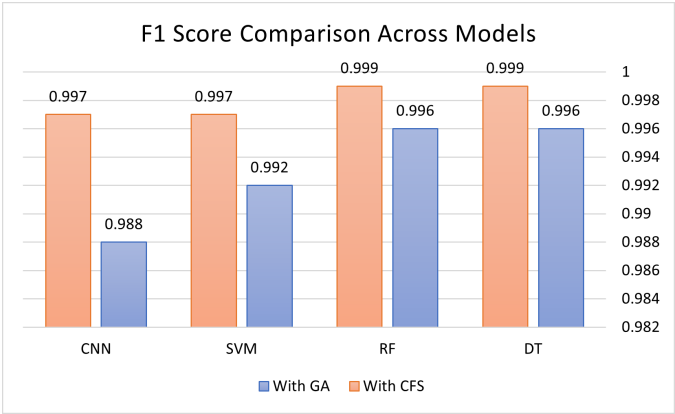

Fig. 7. F1 score.

VII. DISCUSSION

This section interprets the experimental results, highlights the key findings, and compares our approach with existing studies to provide deeper insights into the significance of the work.

A. Key Findings

According to the comprehensive results derived from our experiments, the classifiers that exhibited the highest performance in terms of accuracy, precision, recall, and F1-score were DT and RF, particularly when trained with features selected through CFS. Both models attained an impressive 99.9% across all metrics, underscoring their exceptional reliability and effectiveness in identifying malicious activities within IoT networks.

In terms of computational efficiency, DT surpassed all other classifiers, achieving the quickest training and testing durations with both GA and CFS. This outcome affirms that DT is particularly well-suited for real-time applications and scenarios that necessitate rapid processing with minimal computational overhead.

While the CNN and SVM models demonstrated commendable detection accuracy when CFS was utilized, they exhibited longer training and testing durations, particularly with GA. Furthermore, SVM with GA selection recorded the lowest precision, likely attributable to the algorithm's sensitivity to high-dimensional data and correlated features, which can adversely affect its robustness when handling larger or imbalanced datasets.

Overall, the CFS feature selection method has proven to be more efficient and stable across all classifiers, delivering superior performance compared to GA in most instances. This highlights the critical role of appropriate feature selection in improving classifier accuracy and minimizing computational expenses.

B. Comparative Analysis between Our Proposed Approach and an Existing Study

This section provides a comparative analysis of our proposed ML based IoT security framework in relation to the approach outlined by Nazir et al. [6]. While both studies focus on improving IoT security using ML techniques, they exhibit notable differences in their methodologies, system architectures, and performance results.

Nazir et al. [6] proposed a full-stack collaborative threat intelligence framework that integrates blockchain, ML, and human feedback via an iOS control center. This approach emphasizes real-time, collaborative detection and mitigation, utilizing advanced models like RF, DT, Long LSTM, and CNN. Their study focused on reducing false negatives and improving adaptability in IoT security, primarily using the IoT23 dataset for model training and evaluation. In contrast, our approach is a more streamlined ML-based anomaly detection framework that relies on feature selection to improve model efficiency without requiring additional infrastructure like blockchain. We used four classifiers (DT, RF, SVM, and CNN) and employed two feature selection methods—GA and CFS to enhance detection accuracy and reduce computational overhead.

Our results showed a clear advantage in detection accuracy compared to Nazir et al.'s approach. As illustrated in Fig. 8, our DT and RF models achieved 99.9% accuracy, precision, recall, and F1-score when using CFS, outperforming the 95.0% accuracy reported for the best-performing model (RF) in Nazir et al.'s study. Similarly, our SVM and CNN models reached 99.7% accuracy with CFS, significantly higher than the 92.0% achieved by the CNN model in their study. This marked improvement highlights the effectiveness of our feature selection strategy in boosting model performance, even for DL classifiers like CNN.

Optimizing computational efficiency is essential for real-time IoT applications, and our research revealed notable benefits in this aspect. Our DT model achieved the quickest training and testing durations, finishing training in 0.6821 seconds (GA) and 1.1308 seconds (CFS), with testing times of 0.0168 seconds (GA) and 0.0285 seconds (CFS). In contrast, Nazir et al. did not provide specific training and testing durations, but their use of blockchain and collaborative methods indicates a more resource-demanding strategy. Additionally, our RF model demonstrated effective processing, with training times of 33.4947 seconds (GA) and 43.9254 seconds (CFS), and testing times of 1.1156 seconds (GA) and 1.0211 seconds (CFS). This performance is considerably quicker than what one might anticipate from a comprehensive system like that of Nazir et al., which includes blockchain overhead and real-time feedback loops. On the other hand, our SVM model showed a distinct trade-off, needing 5448.912 seconds (GA) and 158.8664 seconds (CFS) for training, and 1005.804 seconds (GA) and 94.2506 seconds (CFS) for testing. This underscores the computational demands of SVM, despite its impressive accuracy, making it less feasible for real-time applications without additional optimization. The CNN, while also achieving high accuracy, required moderate training and testing times, clocking in at 188.9341 seconds (GA) and 387.9729 seconds (CFS) for training, and 6.1233 seconds (GA) and 24.252 seconds (CFS) for testing.

Although both studies used the IoT23 dataset, our method showed better generalization, consistently achieving high performance across all metrics. This is especially important for real-world IoT settings, where accurately identifying anomalies from various attack types is crucial. While Nazir et al.'s framework is innovative, it might encounter scalability issues due to the extra overhead from its collaborative and blockchain features.

Our research emphasizes the significance of effective feature selection in lowering computational costs and enhancing detection accuracy. The notable performance improvements seen with CFS indicate that future research should focus on this method, particularly in resource-limited IoT environments. Furthermore, our framework's efficient design, which does not depend on external infrastructure, provides a more feasible solution for real-time IoT security.

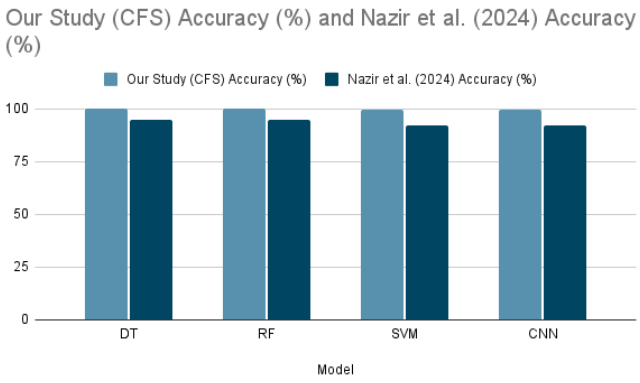

Fig. 8. Accuracy comparison between this study and Nazir et al. for different ML models.

VIII. Conclusion and Future Work

In this paper, we proposed a robust framework to enhance the security of IoT environments by leveraging ML-based

anomaly detection techniques. The framework was designed to address key security challenges in IoT networks, particularly those related to detecting sophisticated attacks such as botnets and DoS. We implemented four widely used classifiers—DT, RF, SVM, and CNN on the IoT23 dataset, a comprehensive resource designed for IoT threat detection.

To further optimize the performance of our classifiers, we incorporated two feature selection algorithms: GA and CFS. The comparative evaluation of these algorithms revealed that CFS consistently outperformed GA in terms of classification accuracy and execution speed. This confirmed the importance of effective feature selection in reducing model complexity and improving detection reliability.

Our experimental results showed that both DT and RF classifiers achieved exceptional performance across all metrics—accuracy, precision, recall, and F1-score—reaching 99.9% when paired with CFS. Moreover, DT exhibited the lowest training and testing times, making it the most suitable model for real-time and resource-constrained IoT environments. While CNN and SVM also showed strong performance, they were more computationally intensive, especially when used with GA.

Despite these promising results, there are some limitations to consider. The study relies on the IoT23 dataset, which may not encompass all possible IoT attack scenarios, and model performance could vary in different network environments or device configurations. Additionally, potential biases in the training data could affect detection accuracy. Addressing these limitations in future work will further strengthen the generalizability and reliability of the proposed framework.

In conclusion, our contribution lies in designing and validating a machine learning-based anomaly detection framework that is both accurate and computationally efficient. Future work will focus on extending the framework to support additional datasets, addressing adversarial threats, and enabling dynamic adaptation in evolving network environments.

For future work, we will conduct and evaluate the performance of our proposed model using other datasets, such as the IoTID20 dataset, as it is recommended for the same condition under which we focus. Moreover, we will try to test it again by selecting other selection algorithms and try to select fewer features with new attack types collected from real-world IoT environments. Evaluating and implementing other types of AI algorithms, such as ML classifiers, will add value for future projects.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [GRANT 252792]. The authors would like to thank the anonymous reviewers for their insightful scholastic comments and suggestions, which improved the quality and clarity of the paper.

REFERENCES

[1] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of internet of things (iot): A survey," *Journal of Network and Computer Applications*, vol. 161, p. 102630, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301041

[2] T. Mazhar, D. B. Talpur, T. A. Shloul, Y. Y. Ghadi, I. Haq, I. Ullah, K. Ouahada, and H. Hamam, "Analysis of iot security challenges and its solutions using artificial intelligence," *Brain Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2076-3425/13/4/683

[3] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[4] R. Ahmad and I. Alsmadi, "Machine learning approaches to iot security: A systematic literature review[formula presented]," 6 2021.

[5] M. A. Amanullah, R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, E. Ahmed, A. Nainar, N. Akim, and M. Imran, "Deep learning and big data technologies for iot security," *Computer Communications*, vol. 151, 02 2020.

[6] A. Nazir, J. He, N. Zhu, A. Wajahat, F. Ullah, S. Qureshi, X. Ma, and M. S. Pathan, "Collaborative threat intelligence: Enhancing iot security through blockchain and machine learning integration," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, 2 2024.

[7] K. Kiran, R. Devisetty, N. Kalyan, K. Mukundini, and K. Ramesh, "Building a intrusion detection system for iot environment using machine learning techniques," *Procedia Computer Science*, vol. 171, pp. 2372–2379, 01 2020.

[8] F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for iot security: Current research and future vision with generative ai and large language models," *Internet of Things and Cyber-Physical Systems*, 01 2024.

[9] F. H. ZAWAIDEH and S. Y. A. BAKER, "Enhanced iot security using cascaded machine learning," *International Journal of Research Studies in Computer Science and Engineering*, vol. 9, pp. 23–34, 2023.

[10] M. Bagaa, T. Taleb, J. B. Bernabe, and A. Skarmeta, "A machine learning security framework for iot systems," *IEEE Access*, vol. 8, pp. 114 066–114 077, 2020.

[11] A. Boukerche and R. W. Coutinho, "Design guidelines for machine learning-based cybersecurity in internet of things," *IEEE Network*, vol. 35, pp. 393–399, 3 2021.

[12] A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim, and M. Alrashoud, "An efficient spam detection technique for iot devices using machine learning," *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1–1, 01 2020.

[13] J. Dhanke, D. Kamalraj, G. Ramesh, K. Sankaran, S. Sharma, and D. Khasim, "A machine learning based iot for providing an intrusion detection system for security," *Microprocessors and Microsystems*, vol. 82, p. 103741, 01 2021.

[14] A. R. Gad, M. Haggag, A. A. Nashat, and T. M. Barakat, "A distributed intrusion detection system using machine learning for iot based on ton-iot dataset," *International Journal of Advanced Computer Science and Applications*, vol. 13, pp. 548–563, 2022.

[15] F. Alrowais, S. Althahabi, S. Alotaibi, A. Mohamed, A. Hamza, R. Marzouk, and M. Ahmed, "Automated machine learning enabled cybersecurity threat detection in internet of things environment," *Computer Systems Science and Engineering*, vol. 45, pp. 687–700, 08 2022.

[16] A. Awajan, "A novel deep learning-based intrusion detection system for iot networks," *Computers*, vol. 12, 2 2023.

[17] U. Tahir, M. Abid, M. Fuzail, and N. Aslam, "Enhancing iot security through machine learning-driven anomaly detection," *VFAST Transactions on Software Engineering*, vol. 12, pp. 01–13, 05 2024.

[18] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in iot," *Journal of Sensor and Actuator Networks*, vol. 12, 4 2023.

[19] M. A. Khatun, S. F. Memon, C. Eising, and L. L. Dhirani, "Machine learning for healthcare-iot security: A review and risk mitigation," *IEEE Access*, vol. 11, pp. 145 869–145 896, 2023.

[20] A. Churcher, R. Ullah, J. Ahmad, S. U. Rehman, F. Masood, M. Gogate, F. Alqahtani, B. Nour, and W. J. Buchanan, "An experimental analysis of attack classification using machine learning in iot networks," *Sensors (Switzerland)*, vol. 21, pp. 1–32, 1 2021.

[21] I. Mukherjee, N. K. Sahu, and S. K. Sahana, "Simulation and modeling for anomaly detection in iot network using machine learning," *International journal of wireless information networks*, vol. 30, no. 2, pp. 173–189, 2023.

[22] P. Bedi, S. Mewada, R. A. Vatti, C. Singh, K. S. Dhindsa, M. Ponnusamy, and R. Sikarwar, "Retracted: Detection of attacks in iot sensors networks using machine learning algorithm," *Microprocessors and Microsystems*, vol. 82, p. 103814, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0141933120309595

[23] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection ids for detecting dos attacks in iot networks based on machine learning algorithms," *Sensors*, vol. 24, no. 2, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/713

[24] B. Alotaibi and M. Alotaibi, "A stacked deep learning approach for iot cyberattack detection," *Journal of Sensors*, vol. 2020, no. 1, p. 8828591, 2020.

[25] A. Huč and D. Trček, "Anomaly detection in iot networks: From architectures to machine learning transparency," *IEEE Access*, vol. 9, pp. 60 607–60 616, 2021.

[26] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection ids for detecting dos attacks in iot networks based on machine learning algorithms," *Sensors*, vol. 24, no. 2, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/2/713