Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062971

ARTICLE

ERBM: A Machine Learning-Driven Rule-Based Model for Intrusion Detection in IoT Environments

Arshad Mehmmod[1,#], Komal Batool[1,#], Ahthsham Sajid[2,3], Muhammad Mansoor Alam[2,3], Mazliham MohD Su'ud[3,*] and Inam Ullah Khan[3]

[1]Department of Information Security and Data Science, Riphah Institute of Systems Engineering, Riphah International University, Islamabad, 46000, Pakistan

[2]Faculty of Computing, Riphah International University, Islamabad, 46000, Pakistan

[3]Faculty of Computing and Informatics, Multimedia University, Cyberjaya, 63100, Malaysia

*Corresponding Author: Mazliham MohD Su'ud. Email: mazliham@mmu.edu.my

[#]These authors contributed equally to this work

Received: 31 December 2024; Accepted: 04 March 2025; Published: 19 May 2025

ABSTRACT: Traditional rule-based Intrusion Detection Systems (IDS) are commonly employed owing to their simple design and ability to detect known threats. Nevertheless, as dynamic network traffic and a new degree of threats exist in IoT environments, these systems do not perform well and have elevated false positive rates—consequently decreasing detection accuracy. In this study, we try to overcome these restrictions by employing fuzzy logic and machine learning to develop an Enhanced Rule-Based Model (ERBM) to classify the packets better and identify intrusions. The ERBM developed for this approach improves data preprocessing and feature selections by utilizing fuzzy logic, where three membership functions are created to classify all the network traffic features as low, medium, or high to remain situationally aware of the environment. Such fuzzy logic sets produce adaptive detection rules by reducing data uncertainty. Also, for further classification, machine learning classifiers such as Decision Tree (DT), Random Forest (RF), and Neural Networks (NN) learn complex ways of attacks and make the detection process more precise. A thorough performance evaluation using different metrics, including accuracy, precision, recall, F1 Score, detection rate, and false-positive rate, verifies the supremacy of ERBM over classical IDS. Under extensive experiments, the ERBM enables a remarkable detection rate of 99% with considerably fewer false positives than the conventional models. Integrating the ability for uncertain reasoning with fuzzy logic and an adaptable component via machine learning solutions, the ERBM system provides a unique, scalable, data-driven approach to IoT intrusion detection. This research presents a major enhancement initiative in the context of rule-based IDS, introducing improvements in accuracy to evolving IoT threats.

KEYWORDS: Rule based; intrusions; IoT; fuzzy prediction

1 Introduction

The Internet of Things (IoT) has transformed how we interact with and connect everyday objects, providing unprecedented convenience and automation. However, IoT devices' rapid proliferation and interconnectivity have introduced significant security challenges and vulnerabilities [1]. IoT-based intrusions refer to unauthorized and malicious activities that exploit weaknesses in IoT systems to gain unauthorized access, manipulate data, disrupt services, or compromise the integrity and privacy of connected devices and networks [2]. Intrusion Detection Systems (IDS) are crucial in securing computer networks and systems by

identifying unauthorized access and malicious activities. Traditionally, the various IDS approaches can be divided into host-based IDS (HIDS), Network-Based IDS (NIDS), and Hybrid IDS. Specialized solutions are needed to overcome challenges in IoT environments, including unique types of attacks [3]. HIDS detects threats to a network's devices (hosts) by monitoring system logs, file integrity, and process behaviors. Unlike network intrusion detection systems, HIDS is concerned with a host's internal State, allowing it to detect threats that do not generate observable network traffic, such as unauthorized file modifications and privilege escalation. Yet, it focuses on host behavior, leaving it vulnerable to network-based attacks [4]. NIDS inspects network incursion, e.g., they watch network traffic in real-time, examining incoming and outgoing data packets on an application of the network. They can detect external threats like DDoS attacks, port scans, and the shipment of malware. By placing it at critical points along the network infrastructure, NIDS provides an overall view of network-based threats but has difficulty recognizing intrusions inside encrypted traffic or within the confines of the host itself [5]. Hybrid IDS represents a middle ground between HIDS and NIDS, combining both strengths to provide a complete security overview. Hybrid systems can detect internal or external threats by encompassing host-based and network-based monitoring. This integration improves detection capabilities but comes at the cost of increased system complexity and resource requirements. The Internet of Things (IoT) brings new security challenges and freshness from the sheer number of devices, diversity of architectures, and resource constraints. Such IDS for IoT environments need to consider these challenges [6]. Examples of common attack types in the IoT context are:

- **Spoofing Attacks:** Attackers falsify IP addresses or identities to deceive systems, imitating legitimate sources to gain unauthorized access.
- **Web Attacks:** These attacks target web applications and infrastructure, exploiting vulnerabilities to breach and manipulate digital platforms.
- **Passive Attacks:** Intruders monitor a system to collect information without altering system resources, often leading to privacy violations.
- **Active Attacks:** These involve direct interactions with the system, such as data modification or disruption of services, potentially causing significant harm.

Several factors contribute to the vulnerabilities of IoT systems, including device heterogeneity, limited computing resources, the growing number of IoT devices, and the lack of standardization [7]. The diversity of devices poses challenges for ensuring consistent security across the ecosystem, as different devices often have varying security features and update capabilities [8]. Traditional rule-based intrusion detection methods are limited in performance [9], particularly when processing the vast amounts of data generated by IoT devices and addressing the evolving nature of cyber threats [9].

This article focuses on improving rule-based intrusion detection techniques tailored to IoT environments. The goal is to improve detection accuracy and overall system performance. Fuzzy logic is a practical approach for handling uncertainty and imprecision, particularly in real-world datasets where binary decisions may fail to capture the data's complexity [10]. This research implements fuzzy logic over standard IoT Intrusion Detection (INID-2019) and CICIOT23 Datasets, which involves predicting a new label feature after fundamental data analysis. Unlike the binary or multi-classification nature of traditional/classical logic, fuzzy logic allows for true classification flexibility, which is crucial for making real-world predictions. Fig. 1 shows how fuzzy logic works.

Using newly generated labels, machine learning models were used to score the accuracy, precision, recall, F1 Score, and false optimistic rate prediction. The proposed hybridization fuses the input of fuzzy logic to handle the vagueness. Due to its highly predictive nature, it applies machine learning models as a holistic way of appraising system performance [10]. The objective is to enhance the prediction accuracy by managing the uncertainty implicitly available in these IoT Network Intrusion Detection (INID) and

CICIOT23 Dataset [7] with fuzzy logic. The overlap in IoT intrusion records has been removed using fuzzy logic, which ultimately improved the result.

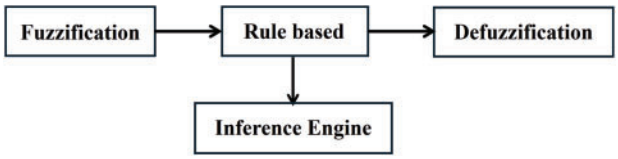


Figure 1: Working of fuzzy logic

The ERBM uses a fuzzy set representation of the important aspects, builds rules using IF-THEN type statements in a few steps, and then uses an inference engine to predict the output. The output values are converted to predicted attack types as part of the defuzzification process, and new features are created.

Key Contributions of This Research:

- Examining the Shortcomings of (Traditional Rule-Based Approaches): A critical review of traditional rule-based approaches highlights their inherent limitations, especially in capturing the intricate relationships in IoT environments.
- The design of the ERBM: The ERBM, based on fuzzy logic/machine learning, improves detection accuracy and adaptability for IoT-based intrusion detection.
- Conducting a Comparative Analysis of Pre-ERBM and Post-ERBM: A performance comparison showing significant improvements in accuracy, precision, and reduction in false-positive rates with the ERBM than without (ERBM) over IoT Network Intrusion Detection-(INID-2019) and CICIOT23 Datasets.

2 Related Work

The rapid growth of IoT and the increasing use of internet-connected devices lead to significant privacy and security challenges. The sophistication of Cyber attacks has highlighted the need for advanced network intrusion detection technologies, with prioritized intelligent testing and verification methodologies, especially in industries like healthcare [11]. This study presents an IDS for IoT networks using supervised machine learning methods, with the UNSW-NB15 dataset normalized using the min-max method to ensure no data loss, which is valuable for addressing IoT security and privacy issues [12]. The proliferation of IoT devices complicates security efforts, with researchers exploring ML algorithms for developing IoT security models despite challenges like high processing overhead and privacy concerns. FL and DL algorithms are also considered for improving security while preserving privacy [13].

The research compares rule-based methods for detecting IoT attacks, focusing on a framework using Bayesian optimization, Gaussian Process, and decision trees, which performed well in detecting botnet attacks in IoT environments [14]. This article reviews IoT-specific security concerns, including standard attacks and threat models. It discusses various IDS approaches, analyzing data on commonly used ML techniques for network security and highlighting unresolved issues [15]. IoT adoption in industrial settings has led to significant changes in automation and control systems, raising security concerns. This research proposes a unique IDS model using PSO, the Bat algorithm, and a Random Forest classifier for IoT networks, showing promising results [16]. Another study presents an interpolation logic detection method to address IoT botnet attacks, achieving a high detection rate and reducing false positives [17]. The rising number of cyberattacks on IoT devices necessitates real-time IDS, with one research proposing a deep learning approach using a multilayered FC network, which showed strong performance in preventing IoT attacks [18].

This research presents an intrusion detection framework for IIoT systems, focusing on collaboration between IoT and edge devices to minimize energy consumption and communication overhead while maintaining detection accuracy [19]. Another study introduces a novel anomaly-based IDS using deep learning models, tested on the CICIDS2017 dataset, which effectively classifies multiple attack types in IoT networks [20]. Despite extensive research, there is still a need for more robust and reliable IDSs for IoT environments, with one study highlighting key factors in developing such systems. Adapting traditional IDS methods to IoT is challenging due to the unique characteristics of IoT devices. A paper reviews the literature on IDS for IoT, categorizing them based on detection methods, deployment strategies, and security threats [21]. Further research introduces a new IDS using a political optimizer and cascade forward neural networks, which perform superiorly on a reference dataset [22]. IoT presents new security challenges that traditional IDS cannot fully address. Another paper analyzes IDS studies on IoT and categorizes proposed IDSs by detection method, deployment process, and other criteria [23]. The need for effective IDS is growing as IoT forms the backbone of future smart cities. A study discusses the trade-offs in balancing detection accuracy with performance overheads, identifying obstacles in achieving effective intrusion detection [24]. Another study focuses on the impact of feature extraction models on the effectiveness of ML-based IDS in IoT, comparing various ML models and feature extractors like VGG-16 and DenseNet [25]. Another study proposes a novel deep learning-based approach to attack detection in IoT networks, showing improved performance over existing methods when tested on the Bot-IoT dataset [26].

RDTIDS combines decision tree and rule-based classifiers, and its integrated approach improves IoT attack detection, outperforming the state-of-the-art [27]. Last, in [28], an article classified recent research into IDS developed for IoT architecture and highlighted essential points for implementing these systems. With the Industrial Internet of Things (IIoT) bringing virtual tools into direct contact with physical systems, it is not surprising that the security challenges this substantial data drawn from large volumes of sensor data are accumulating. Cyberattacks threaten ioT reliability, which led to the growth of network intrusion detection systems (NIDSs). Alongside applying a hybrid feature selection method with the help of a rule-based feature selection method and deep learning, this study focuses on IIoT intrusion detection [29]. The proposed approach learned daily is higher than other techniques for classifying IIoT-related network threats using NSL-KDD and UNSW-NB15 datasets. As a result, it proved to be a promising technique [30].

IoT systems, particularly with the RPL protocol, are very complex, making them weaker regarding data protection. The research focuses on creating an anomaly intrusion detection system based on machine learning and an anomaly dataset produced by the Cooja simulator. The findings revealed increased accuracy and a reduced incidence of false paths, which emphasizes the fundamentals of tackling security concerns in the IoT [29]. Due to improvements in IoT security techniques, especially ML and deep learning (ML/DL), mitigation of zero-day attacks has been achieved with intelligent monitoring systems [31]. This work provided an overview of ML/DL approaches to increasing IoT security and highlighted risks and attack surfaces in IoT architecture, including some recommendations to ensure better security practices [32]. Poor resources and various architectures of IoT networks oversimplify their security challenges. ML and DL strategies are essential for making IoT devices and networks intelligent, but there are also vulnerable and security issues associated with such strategies cited in [10]. Performance optimization and cost-cutting are vital formulas in the 4th Industrial Revolution (4IR), and IoT has become one of its most prominent cornerstones. Analyzing IoT Security Concerns: The Role of AI in Fortifying Online Security: provides business implications and avenues for further research regarding the impact and challenges of IoT security [28]. The potential for widespread IoT adoption is hindered by privacy and security concerns. This work surveyed existing security approaches and provided recommendations for improving IoT security to facilitate mainstream adoption [33]. The rise in IoT usage has increased security risks, with AI playing a crucial role in cybersecurity.

However, fraudsters also exploit AI. A comprehensive review of the existing relevant literature is given in this study; the study further reviews some of the significant literature related to IoT and AI and the daze surrounding it [34]. Traditional intrusion detection systems (IDS) face challenges due to the specific traits of IoT devices. In this paper, we discussed some of rule-based strategies accessible for IoT systems and their advantages and disadvantages, they identified how to improve their performance level for IoT-based intrusion detection [35].

While rule-based approaches help identify deviations caused by attacks from the typical behavior of legitimate users for better IoT intrusion detection, they also have drawbacks. This work evaluated the performance of rule-based strategies against multiple types of intrusions and compared rule-based strategies with other IDS approaches, recommending improvements [36] and using a rule-based system with machine learning for IoT intrusion detection. We next propose its practical applicability by proposing a hybrid framework and validating its effectiveness in real-world IoT deployments because the relevancy of either approach largely depends upon the context, which has direct implications in overall security prisms [37].

Heavy load balancing, environment monitoring, and multiple static rules and dynamic rules to domestic things using anomalies based known and unknown threats detection with a low false positive rate in a rule-based approach for healthcare IoT systems have been proposed which provide a simple option for securing healthcare institutions based on static rule to security prescription and non-prescription and also based on positive recommendation and counter recommendation to vague prescription and non-prescriptio [1]. A rule-based intrusion detection system was designed to secure IoT networks in smart home environments. It could detect accurately with minimal false positives, thus ensuring smart homes are protected sufficiently [38]. In this study, rule-based strategies for protecting IoT-empowered transportation networks were discussed. It showed that the rule-based approach (e.g., the system used through kernel-ids) actually got a very low false positive rate with general detection performance and still works feasibly at a high rate in transportation [39]. It is based on rule-based techniques adopted in the brilliant grid security analysis to compare signature [40] and anomaly-based [41] approaches. The rule-based strategy had perceived accuracy and reliability regarding threat detection, making the results helpful in the generation of file-based smart grid intrusion detection [42].

IDS are among the most essential tools for helping network security by detecting and mitigating threats. The classic IDS techniques sometimes lack adaptability and precision against advanced and evolving cyberattacks. Anomaly-based IDS methods have emerged as an effective solution to these challenges because they can pinpoint yet-unknown malice. This study proposes an anomaly detection system using fuzzy logic, demonstrating that fuzzy logic can improve the certainty of IDS by appropriately managing the uncertainties of network flow characteristics. The method uses fuzzy logic to deal with uncertainty and generate more robust anomalous behavior classification from the data in the network environments [43].

Due to the circumstances above, many researchers have focused on hybrid methodologies that integrate various computational approaches to address the challenge of adequate and accurate IDS. Authors: Mohammed Ishaque, Md Gapar Md Johar, Ali Khatibi, Muhammed Yamin (2023). Such a combination of the fuzzy neural network along with genetic algorithms, which is a new hybrid technique, contributes to enhancing IDS performance. Their research identifies that conventional IDS handles large datasets poorly and struggles to adapt to changing attack patterns. This hybrid approach combines the advantages of all three techniques to enhance detection accuracy while decreasing false favorable rates [44].

Conclusion of related work: Intrusion detection in IoT environments has been extensively studied, with various solutions proposed to enhance security. Traditional rule-based IDS approaches rely on predefined signatures and heuristic rules to identify known threats. Though suitable for identified target patterns, these approaches often perform poorly at generalizing to new or changed attacks. With the advancement of

technology, ML (machine learning)-based IDS models, data-driven methodologies focusing on typical and atypical behavior, have become a feasible decision alternative. Training supervised and unsupervised learning algorithms have improved detection and false positive rates. Nonetheless, such models are usually affected by dataset imbalance, feature selection, and computational constraints for IoT devices. Fuzzy logic has been used in IDS to improve decision-making in uncertain cases. Fuzzy logic combines rule-based reasoning with ML, making it suitable for cases where data is uncertain and unclear, making the detection of anomalies more robust. Despite these benefits, integrating ML and fuzzy logic for IoT intrusion detection is still a work in progress, with issues in scalability, settled efficiency, and flexibility to contemporary threats. Research Gap:- Sect 1—Essential There is a growing need to provide an effective solution to bridge the limitations of existing IDS models. A holistic approach that balances precision, computational efficiency, and capability should be integrated to improve Intrusion Detection in IoT environments.

3 Methodology

The proposed (ERBM) will enhance the effectiveness of rule-based techniques in detecting intrusions in IoT networks. The research methodology consists of several key steps, including dataset analysis, data preprocessing, rule-based model development, evaluation, and performance analysis, as illustrated in Fig. 2.

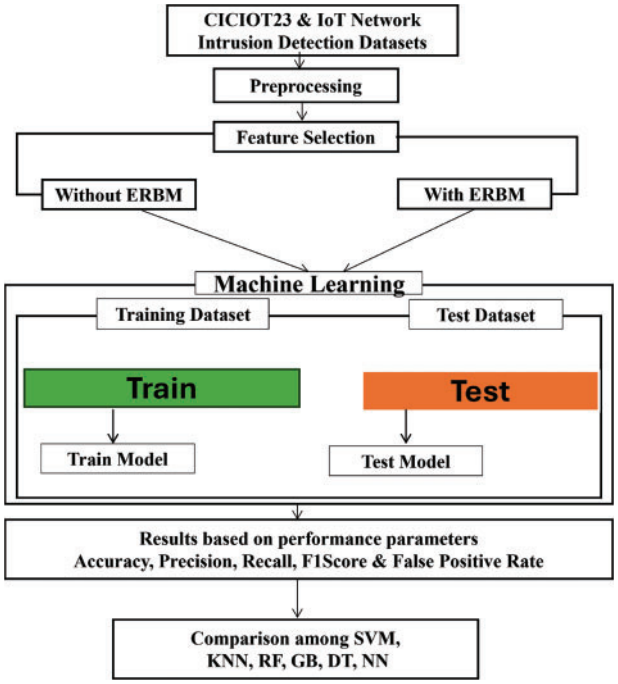


Figure 2: ERBM structural diagram

The implementation of these steps is further explained in Algorithm 1 as Pseudoscope. **IoT Network Intrusion Detection-(INID)** Dataset: The INID-2019 dataset is an IoT network intrusion detection dataset that captures traffic from various IoT devices under normal and attack conditions. It includes multiple attack types, such as DDoS, Botnets, and Exploits, with features like packet size, duration, and protocol types. However, the dataset exhibits class imbalance, with a significantly higher proportion of regular traffic than attack instances.

Algorithm 1: Comparison between pre test without (ERBM) & post test with (ERBM) over IoT network intrusion detection (INID) and CICIOT23 datasets

1: **START**
2: Read dataset *DS*
3: Apply preprocessing on *DS*:
4: a. Handle missing data
5: b. Normalize/Standardize features
6: c. Encode categorical features
/* Pre Test without (ERBM) */
7: Apply Machine Learning on *DS*:
8: a. Split *DS* into training and testing sets
9: b. Train a Machine Learning model (e.g., Random Forest, Gradient Boosting, SVM, KNN, Decision Tree, Neural Network) on the training set.
10: c. Test the model on the testing set and evaluate performance
/* Post Test with (ERBM) */ State Create fuzzy sets:
11: a. Define fuzzy sets for each relevant feature (e.g., "Low," "Medium," "High" for numeric values) State Create fuzzy rule conditions:
12: a. Define fuzzy rules based on feature sets
13: **/* Rule Example */**
14: **if** (Total_Bwd_packets≥16)AND (Total_Bwd_packets ≤ 29)AND (Average_Packet_Size ≤ 174) **then**
15: Attack = "Dictionary Attack"
16: **end if**
17: **for** each data point in *DS* **do**
18: Match fuzzy rules using defined conditions
19: **if** data point matches fuzzy rule **then** State Predict new feature "Attacks" using fuzzy prediction
20: **end if**
21: **end for**
22: Compare Results:
23: Compare ML model results vs. fuzzy rule-based prediction (e.g., accuracy, precision, recall, F1 Score, False Positive Rate)
24: **END**

Certain types of attacks are underrepresented, skewing the models towards majority classes. Moreover, feature dependencies, such as depending on specific IP addresses, might restrict the transferability of models trained on this dataset. Because the traffic is collected from specific IoT devices, the dataset cannot fully represent various IoT network environments. It may thus result in bias in the actual field of application. The dataset suits researchers who want to develop and test intrusion detection systems (IDS) explicitly targeting IoT environments. It is used in the paper mentioned earlier, which is network traffic data captured under every day and attack scenarios from IoT devices. This dataset exhaustively covers features representing different aspects of network traffic, including packet size, protocols in use, source and destination IP addresses, and timestamps. It also labels whether the traffic is benign or some attack. DoS distributed DoS and any other type of offense commonly dealt with by IoT framework vulnerabilities. These researchers hope to provide a dataset for evaluating machine learning algorithms for detecting security threats in IoT networks. We have removed duplicate rows from the data during preprocessing. The feature with the same value is removed, so the size of the datasets is decreased. CICIoT23 dataset: The CICIoT23 dataset, released by the

Canadian Institute for Cybersecurity, provides a rich collection of IoT network traffic with diverse types and attack scenarios (i.e., DoS, DDoS, Botnets, and Exploits). It is rich in network flow features such as TCP flags and entropy-based features. However, like INID-2019, CICIoT23 also suffers from class imbalance, with certain types of attacks dominating the dataset while others are underrepresented. Moreover, specific attacks are synthetically synthesized (in lab-controlled settings) and might not be adequately emulated in real-life attack behaviors. Also, there is a possible bias on the dataset due to network protocols: the models trained on this dataset can hardly be adapted to an IoT environment with different architectures or communication protocols. CICIOT23: IoT Cyber Intrusion Detection DataSet. It usually consists of network traffic data generated through IoT, allowing for several applications, including intrusion detection, anomaly detection, and network behavior analysis.

Preprocessing: **IoT Network Intrusion Detection-(INID)** dataset referenced as (https://ieee-dataport.org/open-access/iot-network-intrusion-dataset) (accessed on 03 March 2025). It contains initially contains 625,784 rows and 86 features. Raw network traffic data is cleaned to remove any irrelevant or redundant information. This includes filtering out noise and correcting any inconsistencies in the data, such as missing or erroneous values. The dataset is then transformed into a format suitable for analysis by normalizing and scaling numerical features to ensure they contribute equally to the model's performance. Categorical features, such as protocol types, are encoded into numerical values to facilitate their use in machine learning algorithms. After processing, the dataset has 311,110 rows and 59 features. Feature Selection: based on domain knowledge and existing data, the most significant features are selected for further usage and rule creation, as shown in Table 1.

Table 1: List of most significant features in IoT network intrusion detection (INID) dataset

Feature	Description
Flow duration	The total time duration of a network flow is measured from the first to the last packet within the flow.
Flow_IAT_Mean	The average inter-arrival time (IAT) between packets within a flow. IAT is the time interval between consecutive packets.
Fwd_Header_Len	The total length of the headers of all forward packets in a flow. Forward packets are those sent from the source to the destination.
Fwd_IAT_Mean	The average inter-arrival time between forwarding packets in a flow.
Bwd_Header_Len	The total length of the headers of all backward packets in a flow. Backward packets are those sent from the destination back to the source.
Flow_Pkts_s	The number of packets per second within a flow, indicating the packet transmission rate.
Idle_Mean	The average time duration when there is no packet transmission within a flow.
Protocol	The communication protocol used in the flow, such as TCP, UDP, ICMP, etc.
Fwd_Act_Data_Pkts	The total number of forwarding packets that contain actual data payload in a flow.

CICIOT23 dataset referenced as (https://www.unb.ca/cic/datasets/iotdataset-2023.html) (accessed on 03 March 2025). It contains 992,381 rows and 85 features, and the IoT Network Intrusion Detection dataset contains 625,784 rows and 86 features. Raw network traffic data is cleaned to remove any irrelevant or redundant information. This includes filtering out noise and correcting any inconsistencies in the data, such as missing or erroneous values. The dataset is then transformed into a format suitable for analysis by

normalizing and scaling numerical features to ensure they contribute equally to the model's performance. Categorical features, such as protocol types, are encoded into numerical values to facilitate their use in machine learning algorithms. After processing, the dataset has 121,028 rows and 72 features. Feature Selection: based on domain knowledge and existing data, the most significant features are selected for further usage and rule creation, as shown in Table 2.

Table 2: List of most significant features in CICIOT23 dataset

Feature	Description
Total Fwd packets	The total number of packets sent from the source to the destination in a flow.
Total Bwd packets	The total number of packets sent from the destination back to the source in a flow.
Down/Up ratio	The ratio of packets sent from the destination to the source (downstream) to those sent from the source to the destination (upstream).
Average packet size	The average size of packets in a flow is calculated over all packets in both directions.
Fwd segment size avg	The average size of TCP segments in the forward direction.
Bwd packet/bulk avg	The average number of packets in a bulk transfer in the backward direction.
Flow Packets/s	The rate of packets per second in a flow, considering packets in both directions.
Protocol	The communication protocol used in the flow, such as TCP, UDP, ICMP, etc.
Subflow fwd packets	The number of forward packets in a subflow is a subset of the main flow.
Packet length variance	The variance in the length of packets within a flow.
SYN flag count	The number of packets with the SYN flag set used to initiate TCP connections.
RST flag count	The number of packets with the RST flag set are used to reset TCP connections.
URG flag count	The number of packets with the URG flag set indicates urgent TCP data.
ECE flag count	The number of packets with the ECE flag set are used in TCP Explicit Congestion Notification (ECN).
PSH flag count	The number of packets with the PSH flag set, indicating the receiver should pass the data to the application immediately.

Fuzzy sets are first defined to partition important feature values into a limited number of classes, e.g., low, medium, and high before rules can be set up. Through this categorization, the system can later understand and make decisions between these data continuous values but within these ranges in a fuzzy logical manner. To cater to the subjectivity and vagueness of real-world data, fuzzy sets provide a framework on which rule-based classification is built. Mainly, Listing 1 down below summarizes the membership functions for each fuzzy set, which are designed intelligibly according to data distribution and the unique nature of every feature. These fuzzy sets play a crucial role in developing rules that cover a broad range of input scenarios. For instance, features such as `Total_Fwd_Packet`, `Total_Bwd_Packets`, and `Down_Up_Ratio` are assigned to fuzzy sets labeled as "low," "medium," and "high." These fuzzy sets capture the varying levels of intensity or value that each feature may exhibit.

The system can create nuanced rules based on defined thresholds and conditions by mapping feature values to specific fuzzy sets. This approach allows for identifying patterns that signal particular behaviors or classifications. For example, specific configurations of these features within the "low" or "high" ranges can trigger conditions in the fuzzy system that classify an input as a "Benign" or an "Attack" scenario. Thus,

constructing fuzzy sets and the rules derived from them provide a flexible and effective way for the system to interpret and respond to a wide array of data inputs. The code snippet is just a basic example whereby Listing 1 shows how fuzzy sets can be built based on the initial feature ranges so that they can set orientation to rules. Such a structured approach ensures that each rule appropriately uses the range definitions to classify input data into different buckets under the fuzzy system, thus enabling intelligent decision-making.

```
membership_functions = {
    'Total Fwd Packet': {
        'Low': lambda x: low(x, 1, 6),
        'Medium': lambda x: medium(x, 6, 8, 8, 44),
        'High': lambda x: high(x, 44, 21252)
    },
    'Total Bwd packets': {
        'Low': lambda x: low(x, 1, 16),
        'Medium': lambda x: medium(x, 16, 20, 20, 43),
        'High': lambda x: high(x, 43, 19493)
    },
    ....................................
    }
```

Listing 1: Membership functions dictionary

Fuzzification refers to converting crisp inputs into degrees of membership for fuzzy sets. It is a key concept in fuzzy logic and fuzzy rule-based systems. After creating rules, the sci-kit learn library is used to apply these rules to each dataset row and predict an attack or standard class. Scikit-fuzzy provides fuzzy logic capabilities such as fuzzy membership functions, fuzzy sets, and fuzzy inference systems. Integrating fuzzification into machine learning workflows can enhance models with fuzzy rules, improving interpretability and handling uncertainty. The library base and create a new feature named "Attack," as shown in Listing 2.

```
# Define fuzzy rules
rules = [
    # Rule R1
    "IF ((Total_Fwd_Packet >= 44) and (Total_Bwd_packets > 44) and" +
    "(Total_Bwd_packets < 19493) and (Down_Up_Ratio >= 4)" +
    "and (Down_Up_Ratio <= 16)) THEN Attack = 'Benign'",

    # Rule R2
    "IF ((Total_Bwd_packets >= 16) and (Total_Bwd_packets <= 29) and " +
    "(Average_Packet_Size <= 174)) THEN Attack = 'Dictionary Attack'",

    # Rule R3
    "IF ((Protocol == 17) and (Fwd_Segment_Size_Avg >= 24) and " +
    "(Fwd_Segment_Size_Avg <= 82) and (Bwd_Packet_Bulk_Avg >= 8))" +
    "THEN Attack = 'DNS Flooding'",
]
```

Listing 2: Fuzzy rules in Python

ML is applied on non (ERBM) dataset and ERMB dataset with label "Attack". The dataset was split into training and testing sets to facilitate model evaluation. The train_test_split function from sci-kit-learn was used to divide the data into training (70%) and testing (30%) sets, ensuring a robust evaluation framework. The algorithms SVM, KNN, RF, GB, DT, and NN are used to evaluate, and the following performance metrics are used.

4 Simulation Setup

We also performed the simulation of the Enhanced Rule-Based Model (ERBM) simulation to evaluate the hybrid method's performance in which fuzzy logic is used for feature selection. In contrast, a machine learning algorithm is used for classification. This simulation is based on the IoT Network Intrusion Detection (INID) and CICIOT23 Datasets with the primary objective of identifying and classifying any malicious activities in the network. It was implemented in Python 3.9 with great data-processing and machine-learning libraries (NumPy, Pandas, and Scikit-learn) and had its fuzzy logic component implemented with the SciKit-Fuzzy library.

All experiments are run on a machine with Intel Core i7-7600U CPU and 16 GB memory. Simulated real-world Internet of Things (IoT) network of benign and malicious network traffic: The new IoT Network Intrusion Detection (INID) and CICIOT23 Datasets. To preprocess the data, we normalized features and imputed missing values. Fuzzy logic was used to discover the frequencies with the highest impact separating regular network traffic from malicious traffic. A fuzzy inference system was defined with membership functions for each feature. This system divided input data into fuzzy sets with the corresponding low-medium-high outputs based on thresholds found in the data distribution. The threshold critical features were determined using domain knowledge; the system then used fuzzy rules to determine the significance of each feature in an intrusion detection problem.

After selecting the best features using fuzzy logic, various ML algorithms such as Decision Trees, Random Forests, and SVM were trained with the dimensionally reduced set of features. To ensure rigorous evaluation, all the data were formed into a training set (70%) and a testing set (30%). The training data was used to train the machine learning models, and performance metrics were used to evaluate accuracy, precision, recall, F1 Score, false-positive, etc.

A comparative analysis of each model's performance was carried out to identify the most effective algorithm for intrusion detection and false-positive rate.

4.1 Mathematical Model for ERBM and Performance Metrics

The Enhanced Rule-Based Model (ERBM) leverages fuzzy logic to preprocess and select key features and then applies machine learning classifiers to identify potential intrusions. Here is a high-level mathematical representation of the ERBM methodology:

1. Fuzzy Set Definition:
   Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of features in the dataset. Define fuzzy sets F_i for each significant feature x_i, with membership functions representing "Low," "Medium," and "High" as shown in Eqs. (1)–(3):

$$\mu_{\text{Low}}(x_i) = \begin{cases} 1, & x_i \le a \\ \dfrac{b - x_i}{b - a}, & a < x_i < b \\ 0, & x_i \ge b \end{cases} \tag{1}$$

$$\mu_{\text{Medium}}(x_i) = \begin{cases} 0, & x_i \le a \text{ or } x_i \ge d \\ \dfrac{x_i - a}{b - a}, & a < x_i < b \\ \dfrac{d - x_i}{d - c}, & c < x_i < d \end{cases} \tag{2}$$

$$\mu_{\text{High}}(x_i) = \begin{cases} 0, & x_i \leq c \\ \dfrac{x_i - c}{d - c}, & c < x_i < d \\ 1, & x_i \geq d \end{cases} \tag{3}$$

2. Fuzzy Rules for Intrusion Detection:
   Define a rule R_j as:

   R_j : IF $(F_1$ is Low$) \wedge (F_2$ is Medium$) \wedge \ldots$ THEN Attack = Type

   For each data point, compute membership values and apply rules to classify the point as either "Attack" or "Benign."
3. Feature Selection and Classification:
   Selected features, $S = \{s_1, s_2, \ldots, s_m\} \subset X$, are used for training machine learning models. Let $\mathbf{x} = (s_1, s_2, \ldots, s_m)$ represent the feature vector for a data point. Train a classifier $f : \mathbf{x} \to y$ where $y \in \{\text{Attack, Benign}\}$ based on labeled training data using algorithms like Decision Tree (DT), Random Forest (RF), and Neural Network (NN). This mathematical model is a structured framework for the ERBM approach. It guides fuzzy set creation, rule application, feature selection, and model evaluation. Integrating this model into the paper would enhance its technical clarity and rigor.

4.2 Rule Optimization Using Fuzzy Inference

To optimize the fuzzy rules, let R_j represent the j-th rule defined as:

R_j : IF $(F_1$ is Low$) \wedge (F_2$ is Medium$) \wedge \ldots$ THEN Attack = Type

We calculate the degree of rule activation, α_j, by aggregating the membership values using Eq. (4):

$$\alpha_j = \min\left(\mu_{F_1}(x), \mu_{F_2}(x), \ldots\right) \tag{4}$$

We derive the output for each rule using the Sugeno or Mamdani inference approach. For Mamdani inference, the defuzzified output y is computed using Eq. (5):

$$y = \frac{\sum_j \alpha_j \cdot y_j}{\sum_j \alpha_j} \tag{5}$$

where y_j represents the output for rule R_j.

4.3 Feature Importance Calculation Using Entropy

To assess feature importance in the fuzzy system, we use entropy to measure the information gained from each feature using Eq. (6):

$$H(X) = -\sum_i P(x_i) \log P(x_i) \tag{6}$$

where $P(x_i)$ is the probability of feature x_i occurring in the dataset. Higher entropy values suggest more informative features.

4.4 Optimization Problem for Parameter Tuning

The ERBM parameters can be optimized as a constrained optimization problem using Eq. (7):

$$\min_{\theta} \mathcal{L}(\theta; X, Y) = \sum_{i} (y_i - f_\theta(x_i))^2 \tag{7}$$

where θ represents the model parameters, X is the feature set, Y is the output label set, and $\mathcal{L}$ is the loss function minimized to fine-tune the ERBM parameters for optimal performance.

This mathematical formulation enhances the ERBM's precision, robustness, and overall performance, making it well-suited for IoT intrusion detection in complex environments.

4.5 Evaluation Metrics

Accuracy (ACC): The proportion of correctly classified instances over the total number of instances using Eq. (8).

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

Precision (PRE): The proportion of correctly identified intrusions over the total number of instances classified as intrusions using Eq. (9).

$$PRE = TP/(TP + FP) \tag{9}$$

Recall (REC): The proportion of correctly identified intrusions over the total number of actual intrusions as in Eq. (10).

$$REC = TP/(TP + FN) \tag{10}$$

F1 Score: A metric that balances precision and recall, providing a single measure of the model's overall performance using Eq. (11).

$$F1 = 2 * ((PRE * REC))/(PRE + REC) \tag{11}$$

False Positive Rate (FPR): The proportion of normal activities incorrectly classified as intrusions using Eq. (12).

$$FPR = FP/(FP + TN) \tag{12}$$

The simulation demonstrated that the hybrid approach, using fuzzy logic for feature selection followed by machine learning classification, significantly improved the accuracy of intrusion detection compared to the traditional rule-based method.

5 Result Analysis

Accuracy

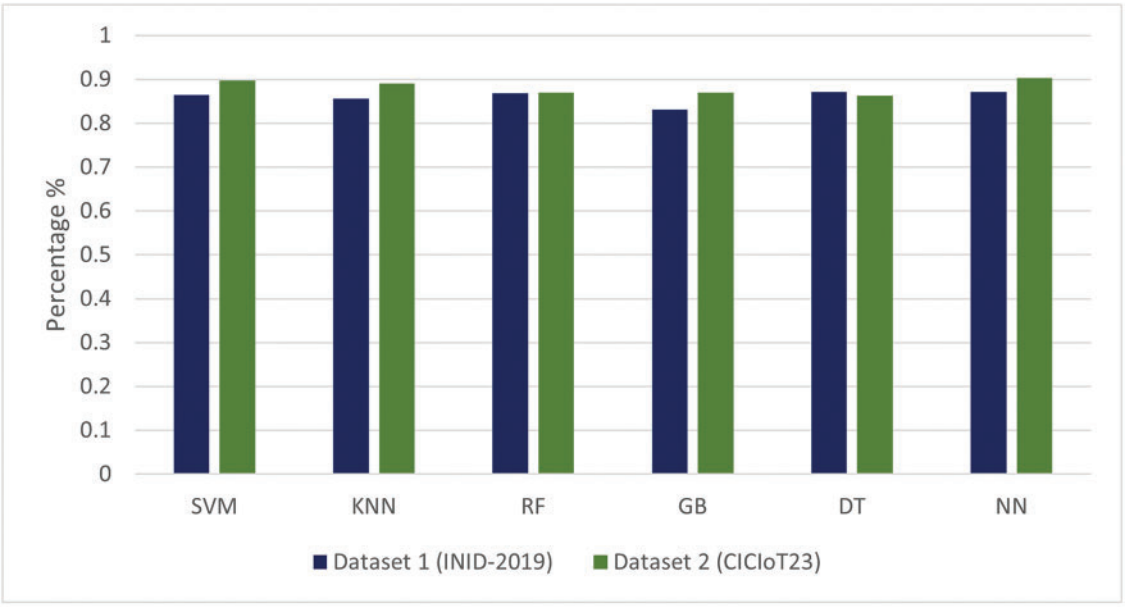
The Extended Regularization-Based Model has proven its effectiveness through the superior performance of machine learning models on the IoT Network Intrusion Detection (INID-2019) and CICIoT23 datasets. Differentially, the Support Vector Machine (SVM) model improved by 15.6% with ERBM, while the K-Nearest Neighbors (KNN) model enhanced by 16%. The Random Forest (RF) model had a strong baseline performance, achieving 0.869 accuracy on both datasets before ERBM.

The IoT and CICIoT23 datasets significantly improved with the Gradient Boosting (GB) model post-ERBM, achieving 0.999 and 0.914, respectively. The Decision tree (DT) model performed similarly, reaching the perfect score after going from 0.871 to high accuracy retention. Although the Neural Network (NN) model performed well initially, a 0.999 accuracy model was obtained after ERBM (Table 3).

Table 3: Accuracy comparison between pre test without (ERBM) and post test with (ERBM) on IoT network intrusion detection (INID) and CICIOT23 datasets

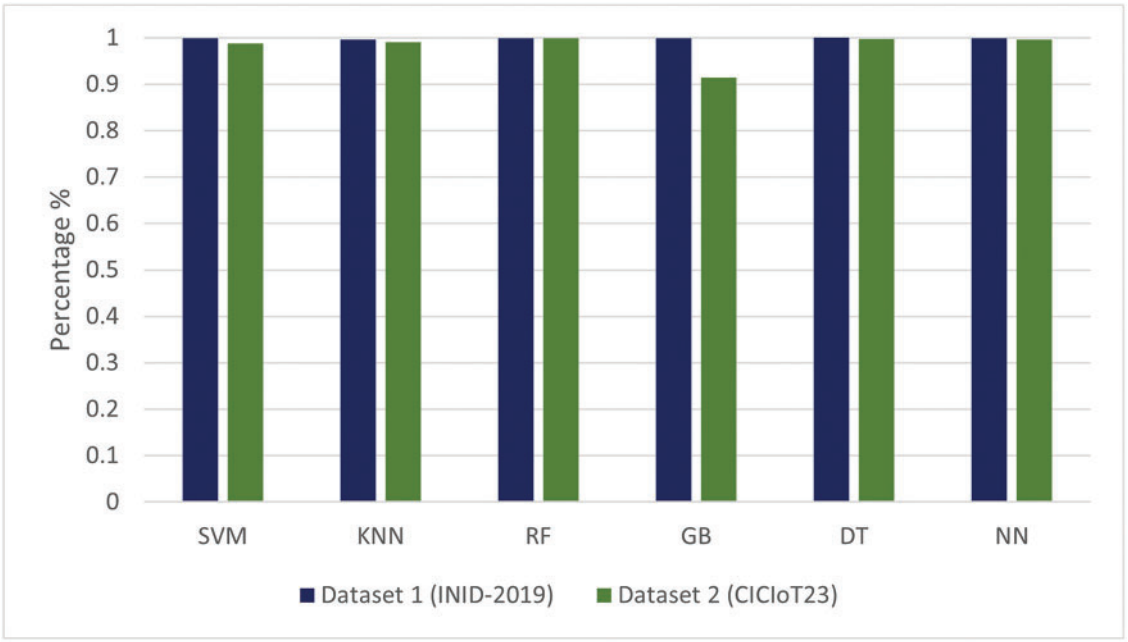
Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
SVM	0.864	0.898	0.999	0.988
KNN	0.857	0.891	0.997	0.991
RF	0.869	0.869	0.999	0.999
GB	0.831	0.869	0.999	0.914
DT	0.871	0.863	1.000	0.998
NN	0.871	0.904	0.999	0.996

ERBM significantly enhances model accuracy across both datasets. The substantial improvements suggest that the ERBM approach bolsters model performance and provides a more robust framework for tackling the complexities of IoT network intrusion detection and analysis, as depicted in Fig. 3a,b.



(a)

Figure 3: (Continued)



(b)

Figure 3: (a) Pre test without (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23). (b) Post test with (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23)

Precision

As we can see from Table 4, the models implemented with the ERBM approach have shown better precision on the IoT Network Intrusion Detection (INID) Dataset and CICIOT23 Datasets. Overall, the Support Vector Machine (SVM) model experienced a particularly noteworthy increase in precision, giving SVM a score of 0.864 on the test and 0.898 on the post-test. The K-Nearest Neighbors (KNN) model also experienced an impressive increase with the pre-test precision scores of 0.857 and 0.891, having jumped to 0.997 and 0.991 at the post-test. In both datasets, RF exhibited an ideal precision of 0.999 during the post-test, confirming that ERBM can strengthen model robustness. Gradient Boosting (GB) had a slightly lower precision but improved to 0.999 and 0.994 in the post-test. The precision of the Decision Tree (DT) model had a unique trend with a precision of 0.871 on the test and 0.970 on the post-test.

In a nutshell, the analysis underscores the work of the proposed ERBM approach in enhancing the accuracy of different machine learning models to predict true positive cases effectively in both IoT Network Intrusion Detection (INID) and CICIOT23 Datasets as shown in Fig. 4a,b, respectively.

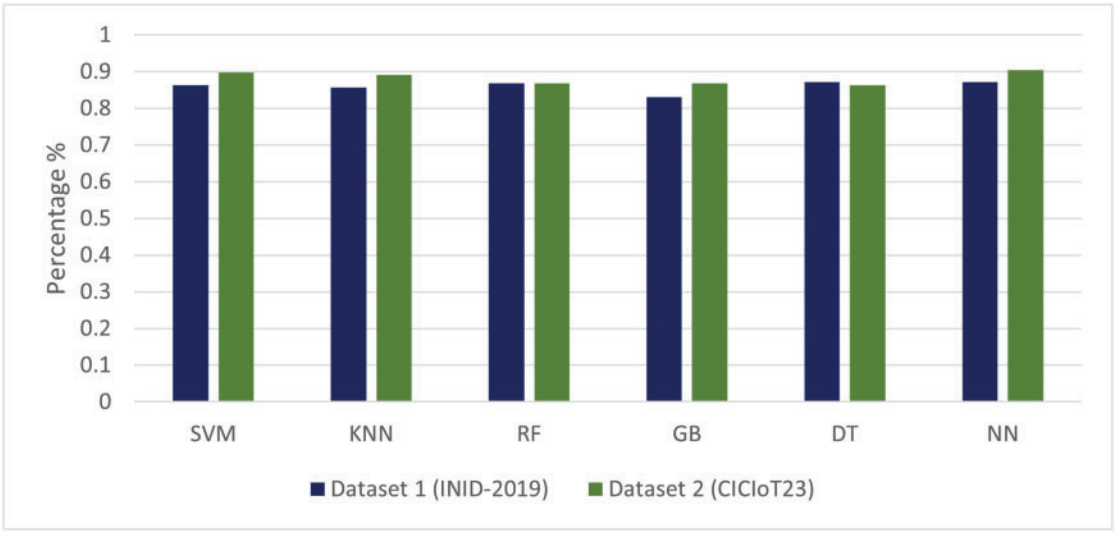
Table 4: Precision comparison between pre test without (ERBM) and post test with (ERBM) on IoT network intrusion detection (INID) and CICIOT23 datasets

Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
SVM	0.864	0.898	0.999	0.988
KNN	0.857	0.891	0.997	0.991
RF	0.869	0.869	0.999	0.999

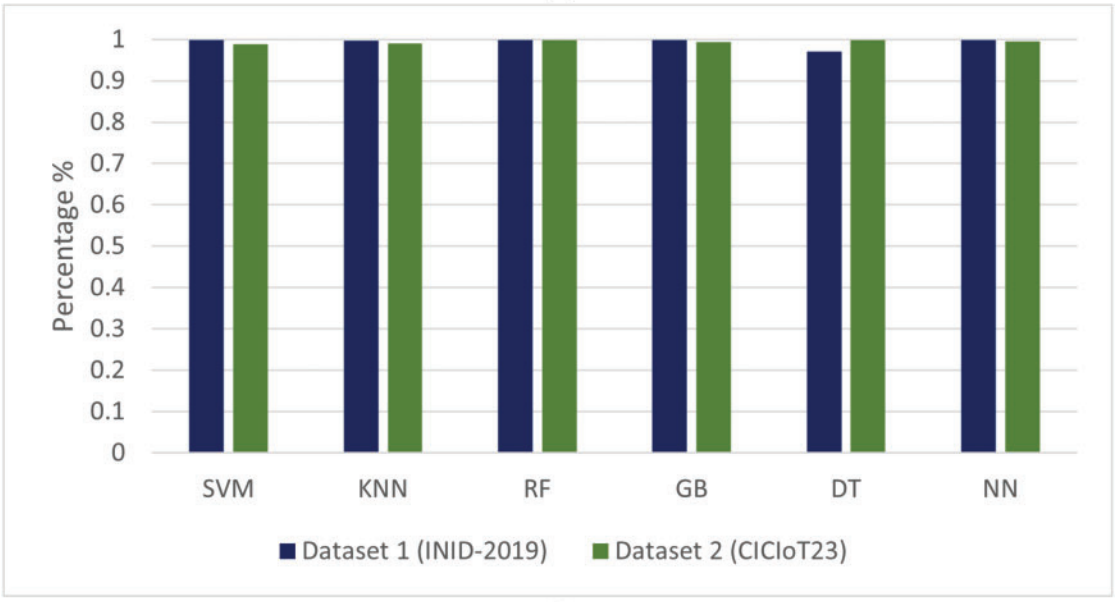
(Continued)

Table 4 (continued)

Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
GB	0.831	0.869	0.999	0.994
DT	0.871	0.863	0.970	0.998
NN	0.871	0.904	0.999	0.996



(a)



(b)

Figure 4: (a) Precision result: pre-test without (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23). (b) Precision result: post test with (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23)

Recall

Recall metrics are the validity of discovering actual positive instances, which indicates the model's effectiveness concerning discovering true positives shown in Table 5 for the IoT Network Intrusion Detection (INID) and CICIOT23 Datasets. With ERBM, the performance (recall) of the models (SVM, KNN, and RF) in the test stage was also considerably strong. Upon ERBM, the Gradient Boosting (GB) model saw a slight decrease in recall scores but an impressive boost overall. The recall of Decision Tree (DT) in the Pre Test was 0.871, and 0.980 in the Post Test, indicating a slightly reduced performance compared to other models.

Table 5: Recall comparison between pre test without (ERBM) and post test with (ERBM) on CICIOT23 and IoT network intrusion detection datasets

Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
SVM	0.864	0.864	0.999	0.999
KNN	0.857	0.857	0.997	0.997
RF	0.869	0.869	0.999	0.999
GB	0.831	0.831	0.999	0.999
DT	0.871	0.871	0.980	0.980
NN	0.871	0.871	0.999	0.999

Overall, the analysis highlights the positive impact of the ERBM approach on the precision of various machine learning models, enhancing their ability to accurately predict true positive instances in both the IoT Network Intrusion Detection (INID) and CICIOT23 Datasetsas shown in Fig. 5a,b.

F1 Score

In Table 6, the F1 Score is a critical metric that balances precision and recall, providing a single measure of a model's accuracy in classifying positive instances. In this analysis, we examine the F1 Score for various models on the IoT Network Intrusion Dataset and CICIoT23, contrasting the results before and after implementing the ERBM approach. Support Vector Machine (SVM) exhibited an F1 Score of 0.864 for both datasets within the pre-testing phase, indicating a relatively effective balance between precision and recall. Following the ERBM application, SVM's F1 Score soared remarkably to 0.999 for both datasets, reflecting a pronounced improvement in classification performance and confirming enhanced complexities handling.

Similarly, K-Nearest Neighbors (KNN) began with an F1 Score of 0.857 in pre-testing for both datasets. Post ERBM, its F1 Score improved impressively to 0.997, showcasing a highly maintained balance between precision and recall effectiveness.

Random Forest (RF) started strongly with an F1 Score of 0.869 for both datasets. After ERBM introduction, it achieved an astounding F1 Score of 0.999, indicating reliability in True positive prediction while minimizing false positives. Gradient Boosting (GB) exhibited slightly lower F1 Scores in pre-testing, scoring 0.831 for both datasets. However, GB's F1 Score improved drastically to 0.999 in post-testing, mirroring overall observed enhancement trends. Decision Tree (DT) initially recorded an F1 Score of 0.871 for both datasets, but this decreased slightly to 0.970 after the ERBM application. While this minor drop suggests a reduced balance between precision and recall, DT maintained robust performance. Lastly, the Neural Network (NN) model showed consistent results, with an F1 Score of 0.871 in pre-testing, which improved remarkably to 0.999 in post-testing for both datasets. This demonstrates NN's strong adaptability and enhanced classification performance with ERBM.

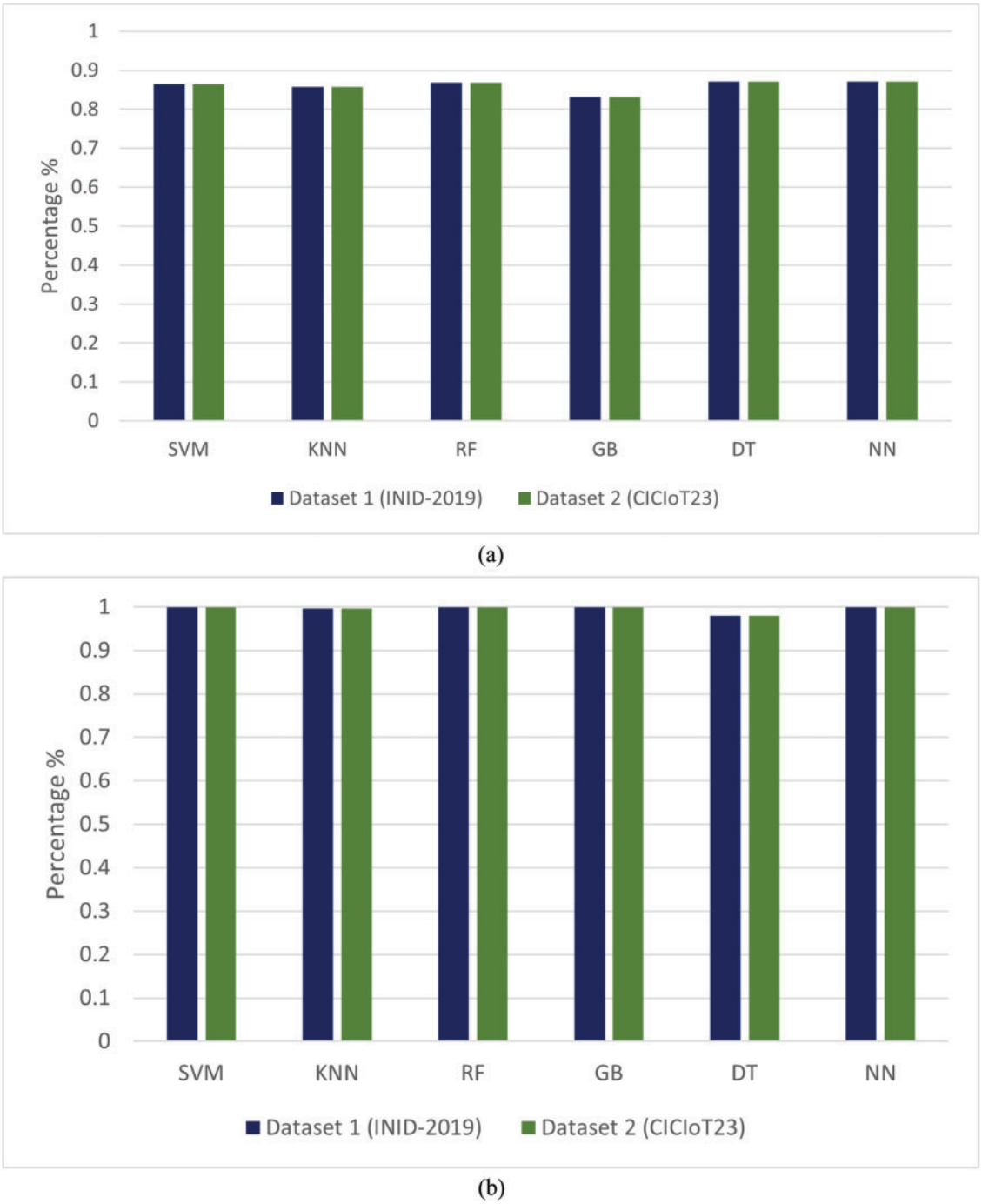


Figure 5: (a) Recall result: pre-test without (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23). (b) Recall result: post test with (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23)

Table 6: F1 Score comparison between pre test without (ERBM) and post test with (ERBM) on IoT network intrusion detection (INID) and CICIOT23 datasets

Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
SVM	0.864	0.864	0.999	0.999
KNN	0.857	0.857	0.997	0.997
RF	0.869	0.869	0.999	0.999
GB	0.831	0.831	0.999	0.999
DT	0.871	0.871	0.970	0.970
NN	0.871	0.871	0.999	0.999

The analysis compares F1 Scores of various models on the IoT Network Intrusion Detection (INID) and CICIOT23 Datasets before and after the ERBM approach. SVM, KNN, and RF showed high F1 Scores, with SVM significantly improving post-ERBM. GB also improved after ERBM, while DT remained strong. The NN model's F1 Score increased from 0.871 to 0.999, as illustrated in Fig. 6a,b.

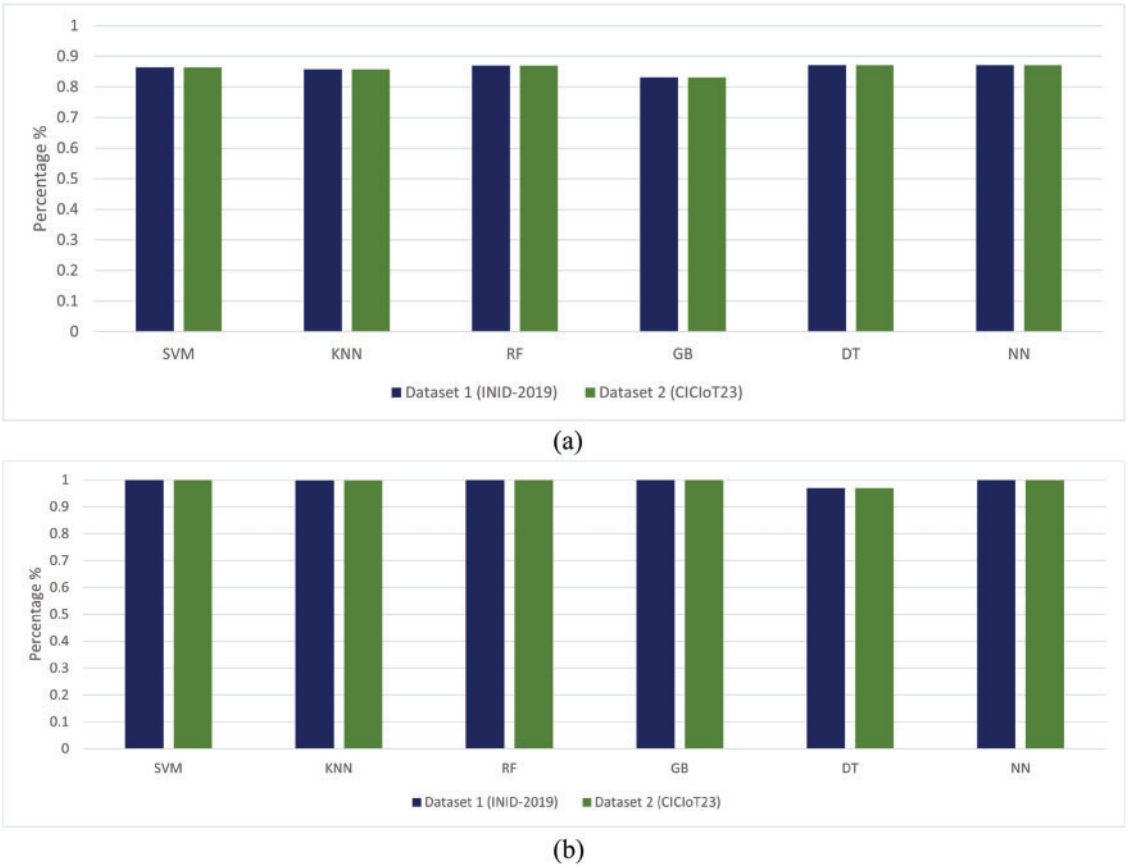


Figure 6: (a) F1 Score result: pre-test without (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23). (b) F1 Score result: post test with (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23)

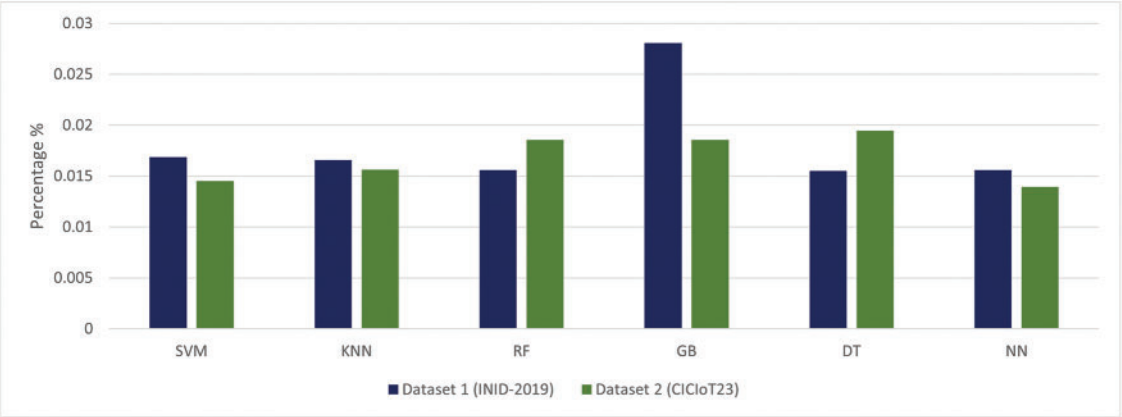
False Positive Rate

Table 7 compares model performance on IoT Network Intrusion Detection (INID) and CICIOT23 Datasets before and after implementing the ERBM approach. The Support Vector Machine (SVM) model significantly decreased error rates post-ERBM, while the K-Nearest Neighbors (KNN) model showed similar reductions. The Random Forest (RF) model showed substantial improvements, while Gradient Boosting (GB) showed higher error rates but improved significantly post-ERBM. The Decision Tree (DT) model had pre-test rates of 0.0155 and 0.0195, which dropped to 0.0003 after ERBM. The Neural Network (NN) model showed the most substantial enhancement in error reduction. Overall, all models showed considerable improvement after ERBM. The false positive rate is reduced due to the nature of the dataset and existing attacks.

Table 7: False positive rate comparison between pre test without (ERBM) and post test with (ERBM) on IoT network intrusion detection (INID) and CICIOT23 datasets

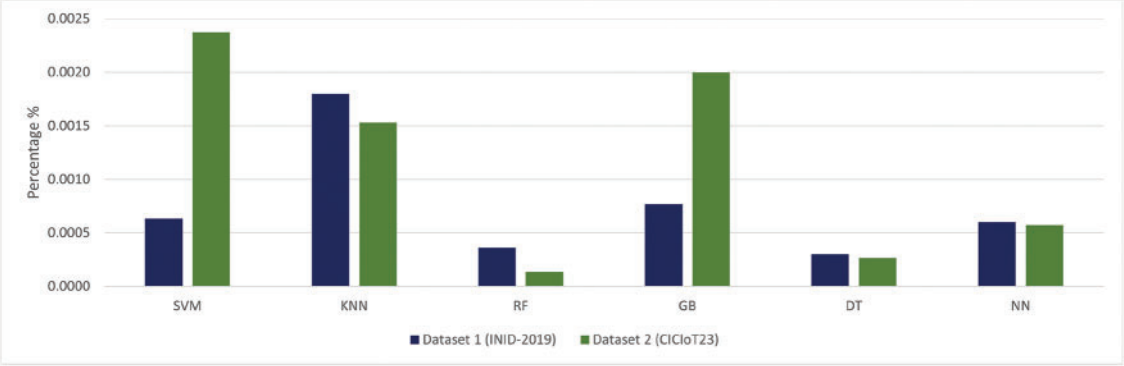
Model	Pre test without (ERBM)		Post test with (ERBM)	
	Dataset 1 (INID-2019)	Dataset 2 CICIoT23	Dataset 1 (INID-2019)	Dataset 2 CICIoT23
SVM	0.0169	0.0145	0.0006	0.0024
KNN	0.0166	0.0156	0.0018	0.0015
RF	0.0156	0.0186	0.0004	0.0001
GB	0.0281	0.0186	0.0008	0.0020
DT	0.0155	0.0195	0.0003	0.0003
NN	0.0156	0.0139	0.0006	0.0006

The analysis reveals that all models significantly improved performance after implementing the ERBM approach IoT Network Intrusion Detection (INID) and CICIOT23 Datasets. SVM, KNN, RF, GB, DT, and NN all showed reduced error rates post-ERBM, with RF achieving the lowest error rates overall. Gradient Boosting displayed the most substantial improvement in error reduction, highlighting the effectiveness of the ERBM technique across various models, as illustrated in Fig. 7a,b.



(a)

Figure 7: (Continued)



(b)

Figure 7: (a) False positive rate result: pre-test without (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23). (b) False positive rate result: post test with (ERBM) over Dataset 1 (INID-2019) and Dataset 2 (CICIOT23)

In summary, the (Post Test with (ERBM) Model) configuration led to dramatic improvements in all performance metrics compared to the Pre Test without (ERBM) Model setup. Each model experienced significant gains in accuracy, precision, recall, and F1 Score, with Random Forest, Decision Tree, and Neural Networks achieving near-perfect results. Additionally, false-positive rates were significantly reduced across the board, enhancing the models' reliability and effectiveness. These enhancements underscore the substantial benefits of the Post Test with (ERBM) Model configuration in optimizing machine learning models for more accurate and reliable performance.

6 Result Discussion

The comparison analysis is shown as only the CICIOT23 dataset is already used in the existing study. Table 8 depicts the post-test with (ERBM) Model models, specifically Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB), consistently outperforming other methodologies across all metrics: accuracy, precision, recall, and F1 Score. These models achieve almost perfect results, with accuracy values 0.999 for DT, RF, and GB and 0.998 for the Neural Network (NN) model. In contrast, models from other studies show relatively lower performance. The Random Forest (8 class) model at 0.994 achieves the highest accuracy in other studies, followed by LSTM with 0.9875. However, the NN model in the RF (8 class) dataset performs poorly, with an accuracy of only 0.67.

Precision in the current study is perfect (1) for DT, RF, and GB and near-perfect (0.998) for NN. In other studies, the MLP model 0.99 shows the best precision, while LSTM follows closely at 0.9866. However, the NN model in the RF (8 class) dataset again lags with a precision of just 0.67. In the current study, recall is also perfect (1) for DT, RF, and GB, with NN achieving 0.998. Other models show lower recall, with MLP performing the best at 0.99, but the NN model in the RF (8 class) dataset shows a significant drop with a recall of 0.90. Finally, the F1 Score in the current study is perfect (1) for DT, RF, and GB and 0.998 for NN. Outside the current study, MLP performs well with an F1 Score of 0.99, followed by LSTM at 0.9859. However, the NN model in the RF (8 class) dataset has the lowest F1 Score of 0.69. In conclusion, the Post Test with (ERBM) Model models consistently achieves the best results, with almost flawless classification performance, while other methodologies, such as MLP, LSTM, and NN (in the RF dataset), show varying degrees of lower performance. The NN model in the RF (8 class) dataset is notably the least effective, demonstrating the lowest scores across all metrics.

Table 8: Comparison of (ERBM) with other studies

Articles	Methodology	Accuracy	Precision	Recall	F1 Score
[5]	DT	0.86	0.82	0.86	0.82
[5]	RF	0.96	0.96	0.96	0.96
[35]	LSTM	0.98	0.98	0.98	0.98
[38]	RF (8 class)	0.99	0.70	0.91	0.71
[38]	NN	0.99	0.67	0.90	0.69
[39]	MLP	0.98	0.96	0.99	0.94
[39]	AE	–	–	–	–
[40]	RF	0.95	0.95	0.97	0.96
[40]	GB	0.96	0.94	0.95	0.94
(ERBM)	DT	0.97	0.97	0.97	0.97
(ERBM)	RF	0.98	0.97	0.97	0.97
(ERBM)	NN	0.98	0.98	0.98	0.98
(ERBM)	GB	0.98	0.96	0.96	0.96

7 Conclusion

The realization of the configuration for the post-test with an Enhanced Random Balance Model has resulted in significant performance improvement for all machine learning models assessed. It resulted in noteworthy improvements in precision, recall, F1 Score, and accuracy, all indicative that the model could make fine adjustments towards sensitivity and specificity. Especially worth noting is that the significant decrease in false favorable rates reflects efficiency in the ERBM for reducing classification errors; this increases the reliability and robustness of the model. The post-test with the ERBM forms a structured approach toward class imbalance handling and the optimization of predictive performance, especially in those scenarios where high accuracy and balanced classification may be required. It does so by balancing a dataset in a strategic way that improves representation for minority classes. Therefore, a more nuanced model results that can generalize well across diverse input data. Improvements in both recall and precision indicate a more accurate classification system and one that can capture the more subtle patterns within complex data distributions.

These findings confirm the post-test with ERBM as an indispensable improvement tool in machine learning workflows due to its enhanced approach to model optimization. This further optimizes the predictive capability of machine learning models to return accurate and robust results, promoting their practical applicability in real-world scenarios.

8 Future Work

Therefore, future research might focus on improving the design in ERBM or exploring other configurations to optimize model performance. For instance, various feature selection methods or hyperparameter manipulation might be used to pursue it. A more accurate assessment of the ERBM-enhanced models' generalization and performance capabilities in various domains may be possible by testing them on a wider range of datasets. Additional improvement opportunities could be found by comparing them with other cutting-edge approaches or optimization strategies. Their resilience would be shown by monitoring optimized models' performance in the real world. This area would progress, and the capabilities of such models would be improved with more work on improving model interoperability, scalability, and computing

efficiency, as well as integrating ERBM with cutting-edge methods like deep learning or transfer learning. Moreover, various attacks like zero-day attack, scalability, and comparison with the Deep Learning model can be made to enhance this model.

Acknowledgement: The authors are grateful to all the editors and anonymous reviewers for their comments and suggestions.

Funding Statement: A research grant from the Multimedia University, Malaysia supports this work.

Author Contributions: Conceptualization was carried out by Arshad Mehmood and Komal Batool, with Ahthsham Sajid leading the methodology. Arshad Mehmood handled software development, while validation was performed by Mazliham MohD Su'ud, Muhammad Mansoor Alam, and Ahthsham Sajid. Inam Ullah Khan conducted a formal analysis. Arshad Mehmood and Muhammad Mansoor Alam lead the investigation. Komal Batool provided resources, and managed data curation. Arshad Mehmood and Ahthsham Sajid prepared the original draft, also contributed to the review and editing process. Visualization was overseen by Inam Ullah Khan, with supervision provided by Komal Batool. Muhammad Mansoor Alam and Mazliham MohD Su'ud managed project administration. Funding acquisition was secured by Ahthsham Sajid, Muhammad Mansoor Alam, and Mazliham MohD Su'ud. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Hussain F, Hussain R, Hassan SA, Hossain E. Machine learning in IoT security: current solutions and future challenges. IEEE Commun Surv Tutor. 2020;22(3):1686–721. doi:10.1109/COMST.2020.2986444.
2. Khraisat A, Alazab A. A critical review of intrusion detection systems in the Internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets, and challenges. Cybersecurity. 2021;4(1):18. doi:10.1186/s42400-021-00077-7.
3. Abdulganiyu OH, Tchakoucht TA, Saheed YK. Towards an efficient model for network intrusion detection system (IDS): systematic literature review. Wir Netw Cybersecur. 2024;30(1):453–82. doi:10.1007/s11276-023-03495-2.
4. Alsharbaty FS, Ali QI. Smart electrical substation cybersecurity model based on WPA3 and cooperative hybrid intrusion detection system (CHIDS). Smart Grids Energy. 2024;9(1):11. doi:10.1007/s40866-024-00192-7.
5. Muneer S, Farooq U, Athar A, Raza MA, Ghazal TM, Sakib S. A critical review of artificial intelligence based approaches in intrusion detection: a comprehensive analysis. J Eng. 2024;2024(1):3909173. doi:10.1155/2024/3909173.
6. Kizza JM. Internet of things (IoT): growth, challenges, and security. In: Guide to computer network security, texts in computer science. Cham, Switzerland: Springer; 2024. p. 557–73. doi:10.1007/978-3-031-47549-8_25.
7. Ali MH, Jaber MM, Abd SK, Rehman A, Awan MJ, Damaševičius R, et al. Threat analysis and distributed denial of service (DDoS) attack recognition in the Internet of Things (IoT). Electronics. 2022;11(3):494. doi:10.3390/electronics11030494.
8. Kebande VR, Awad AI. Industrial internet of things ecosystems security and digital forensics: achievements, open challenges, and future directions. ACM Comput Surv. 2024;56(5):1–37. doi:10.1145/363503.
9. Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Comput. 2020;23:1397–418. doi:10.1007/s10586-019-03008-x.
10. Jony A, Arnob AK. A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset. J Edge Comput. 2024;3(1):28–42. doi:10.55056/jec.648.

11. Waltl B, Bonczek G, Matthes F. Rule-based information extraction: advantages, limitations, and perspectives. In: Jusletter IT 22; 2018.
12. Nazim M, Mohammad CW, Sadiq M. Fuzzy-based methods for the selection and prioritization of software requirements: a systematic literature review. In: Evolution in Computational Intelligence: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021); 2022; 1st ed. Singapore: Springer Nature Singapore. p. 115–29.
13. Kaur K, Kaur A, Gulzar Y, Gandhi V. Unveiling the core of IoT: comprehensive review on data security challenges and mitigation strategies. Front Comput Sci. 2024;6:1420680. doi:10.3389/fcomp.2024.1420680.
14. Mishra N, Pandya S. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: a systematic review. IEEE Access. 2021;9:59353–77. doi:10.1109/ACCESS.2021.3073408.
15. Kang H, Ahn DH, Lee GM, Yoo JD, Park KH, Kim HK. IoT network intrusion dataset IEEE Dataport. 2019. doi:10.21227/q70p-q449.
16. Butt N, Shahid A, Qureshi KN, Haider S, Ibrahim AO, Binzagr F, et al. Intelligent deep learning for anomaly-based intrusion detection in IoT smart home networks. Mathematics. 2022;10(23):4598. doi:10.3390/math10234598.
17. Mliki H, Kaceam A, Chaari L. A comprehensive survey on intrusion detection based machine learning for IoT networks. ICST Trans Secur Saf. 2021;8(29):e3. doi:10.4108/eai.6-10-2021.171246.
18. Kayode Y, Idris A, Misra S, Kristiansen M, Colomo-palacios R. A machine learning-based intrusion detection for detecting internet of things network attacks. Alex Eng J. 2022;61(12):9395–409. doi:10.1016/j.aej.2022.02.063.
19. Gugueoth V, Safavat S, Shetty S. Security of internet of things (IoT) using federated learning and deep learning—recent advancements, issues and prospects. ICT Express. 2023;9(5):941–60. doi:10.1016/j.icte.2023.03.006.
20. Abdulganiyu OH, Tchakoucht TAit, Saheed YK. A systematic literature review for network intrusion detection system (IDS). Int J Inf Secur. 2023;22:1125–62. doi:10.1007/s10207-023-00682-2.
21. Zhang L, Ma D. A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks. IEEE Access. 2022;10:10852–66. doi:10.1109/ACCESS.2022.3145007.
22. Bhavsar M, Roy K, Kelly J, Olusola O. Anomaly-based intrusion detection system for IoT application. Disc Inter Things. 2023;3:5. doi:10.1007/s43926-023-00034-5.
23. Hidayat I, Ali MZ, Arshad A. Machine learning-based intrusion detection system: an experimental comparison. J Computat Cognit Eng. 2022;2(2):88–97. doi:10.47852/bonviewJCCE2202270.
24. Injadat M, Moubayed A, Shami A. Detecting botnet attacks in IoT environments: an optimized machine learning approach. In: 32nd International Conference on Microelectronics (ICM); 2020; Aqaba, Jordan. p. 1–4.
25. Chen Z. Machine learning-enabled IoT security: open issues and challenges under advanced persistent threats. ACM Comput Surv. 2022;55(5):1–37. doi:10.1145/3530812.
26. Gaber T, Awotunde JB, Folorunso SO, Ajagbe SA, Eldesouky E. Industrial internet of things intrusion detection method using machine learning and optimization techniques. Wirel Commun Mob Comput. 2023;2023(1):3939895.
27. Awajan A. A novel deep learning-based intrusion detection system for IoT networks. Computers. 2023;12(2):34. doi:10.3390/computers12020034.
28. Verma A, Ranga V. Machine learning based intrusion detection systems for IoT applications. Wireless Pers Commun. 2020;111:2287–310. doi:10.1007/s11277-019-06986-8.
29. Arshad J, Ajmal M, Mahmoud M, Salah K. An intrusion detection framework for energy constrained IoT devices. Mech Syst Signal Process. 2020;136:106436. doi:10.1016/j.ymssp.2019.106436.
30. Elnakib O, Shaaban E, Mahmoud M, Emara K. EIDM: deep learning model for IoT intrusion detection systems. J Supercomput. 2023;79:13241–61. doi:10.1007/s11227-023-05197-0.
31. Elrawy MF, Awad AI, Hamed HFA. Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comput. 2018;7(1):1–20. doi:10.1186/s13677-018-0123-6.
32. Bogaz B, Sanches R, Toshio C, De SC. A survey of intrusion detection in internet of things. J Netw Comput Appl. 2017;84:25–37. doi:10.1016/j.jnca.2017.02.009.
33. Alghamdi MI. A hybrid model for intrusion detection in IoT applications. Wireless Commun Mobile Comput. 2022;(1):1–9.

34. Mohamed TS, Aydin S. IoT-based intrusion detection systems: a review. Smart Science. 2022;10(4):265–82. doi:10.1080/23080477.2021.1972914.
35. Spadaccino P, Cuomo F. Intrusion detection systems for IoT: opportunities and taxonomy. 2022. doi:10.3390/app13095427.
36. Alosaimi S, Almutairi SM. An intrusion detection system using bot-IoT. Appl Sci. 2023;13(9):5427. doi:10.3390/app13095427.
37. Lysenko S, Bobrovnikova K, Savenko O, Shchuka R. Technique for cyberattacks detection based on DNS traffic analysis. Comput Sci. 2022;73(540):1–10.
38. Ferrag MA, Maglaras L, Ahmim A. RDTIDS: rules and decision tree-based intrusion detection system for internet-of-things networks. Future Internet. 2020;12(3):44. doi:10.3390/fi12030044.
39. Bhargavi M, Kumar MN, Meenakshi NV, Lasya N. Intrusion detection techniques used for internet of things. Int J Recent Technol Eng. 2019;14(24):4462–6.
40. Bouazza A, Debbi H, Lakhlef H. Machine learning-based intrusion detection system against routing attacks in the internet of things. 2022 [cited 2025 Feb 1]. Available from: http://ceur-ws.orgISSN1613(2022):0073.
41. Al-garadi MA, Mohamed A, Al-ali A, Du X, Guizani M. A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Commun Surv Tutor. 2020;22(3):1646–85. doi:10.1109/COMST.2020.2988293.
42. Fair C. Role of artificial intelligence in the internet of things (IoT) cybersecurity. Discov Internet Things. 2021;1:7. doi:10.1007/s43926-020-00001-4.
43. Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.
44. Almseidin M, Al-Sawwa J, Alkasassbeh M. Anomaly-based intrusion detection system using fuzzy logic. In: 2021 International Conference on Information Technology (ICIT); 2021 Jul 14–15; Amman, Jordan. p. 290–5.