International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.4 (2025) pp. 9526-9539
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Anomaly Detection for the Internet of Things Using Machine Learning Techniques

Lotfi Hazzam[1]*, Samir Fenanir[2]

[1]Faculty of Medicine, Ferhat Abbas Setif1 University, Algeria
* **Corresponding Author Email:** l.hazzam@univ-setif.dz- **ORCID:** 0000-0002-5247-0050

[2]Laboratory of Artificial Intelligence, Computer Science Department, Faculty of Sciences, Ferhat Abbas Setif 1 University, Algeri
Email: samir.fenanir@univ-setif.dz - **ORCID:** 0000-0002-5247-7950

Article Info:

DOI: 10.22399/ijcesen.4166
Received : 25 November 2025
Revised : 15 December 2025
Accepted : 20 December 2025

Keywords

IoT Security,
Intrusion Detection System (IDS),
Machine Learning,
Feature Selection,
NSL-KDD,
Cybersecurity.

Abstract:

The rapid growth of the Internet of Things (IoT) has introduced considerable security challenges, making Intrusion Detection Systems (IDS) essential for protecting connected devices from cyber threats. This paper presents an Artificial Intelligence (AI)-driven Intrusion Detection System (IDS) designed for IoT environments, utilizing machine learning and anomaly detection techniques. The system leverages the NSL-KDD dataset to distinguish between normal and anomalous network traffic. We implemented multiple feature selection techniques, including correlation-based selection, Recursive Feature Elimination (RFE), and LASSO, to optimize model performance. Various machine learning classifiers, including Random Forest (RF), Bagging, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), were evaluated using key performance metrics such as accuracy, recall, precision, and AUC-ROC. Experimental results indicate that Random Forest and Bagging outperformed other models, achieving the highest accuracy of 99.89%, with an AUC-ROC of 99.88%, outperforming other models in intrusion detection. Additionally, feature selection methods effectively reduced dataset dimensionality while maintaining high detection performance. Despite these promising results, challenges such as high training time, false positives, and real-time adaptability remain critical concerns for practical deployment in IoT systems.

1. Introduction

Over the past decade, connected devices such as computers, sensors, and smartwatches have profoundly transformed our daily lives, forming what is now known as the Internet of Things (IoT) [1]. The IoT represents a revolutionary technological paradigm aimed at enhancing multiple facets of human life by integrating interconnected devices embedded with sensors, actuators, and communication technologies [2]. These devices serve as bridges connecting the physical and digital worlds, enabling real-time monitoring, automation, and interaction with the surrounding environment [3].The IoT offers significant opportunities across multiple domains, including agriculture, healthcare, surveillance, transportation, and industrial automation. Technological advancements in these fields enable increased efficiency, improved decision-making, and new possibilities for innovation [4]. However, the widespread adoption and seamless integration of IoT devices into our daily lives introduce major challenges, particularly regarding data security and privacy [5][6].Due to their connectivity and often limited computational capabilities, IoT devices are highly vulnerable to cyberattacks. These devices frequently collect, transmit, and process sensitive information without explicit user awareness, making them prime targets for adversaries. Common attack types include:

- **Remote-to-User (R2U) Attacks**: Unauthorized access attempts that allow attackers to extract confidential data.
- **Denial-of-Service (DoS) Attacks**: Disruptions that overwhelm a service, rendering it unavailable to legitimate users [7].

- **Probing Attacks**: Scans that exploit vulnerabilities to bypass network security mechanisms, particularly targeting devices with publicly accessible IP addresses [8].

To mitigate these security risks, **Intrusion Detection Systems (IDS)** have become an **essential pillar** of modern cybersecurity frameworks. **Engineered to monitor, detect, and respond**, an IDS proactively identifies **malicious activities** within a system or network, enhancing overall security and threat mitigation. It serves as a secondary layer of defense, reinforcing preventive measures such as encryption and authentication mechanisms. The primary goal of an IDS is to precisely detect attacks while ensuring optimal resource utilization, particularly in resource-limited environments like IoT networks.

IDS can be generally classified into two primary categories:

- **Signature-based IDS (Misuse-based IDS)**: Detects intrusions by analyzing incoming data and comparing it to a predefined database of known attack signatures. While highly efficient at recognizing familiar threats, it has limitations in identifying emerging or evolving attack patterns [9].
- **Anomaly-based IDS**: Detects intrusions by recognizing anomalies that deviate from established normal behavior. This approach is well-suited for recognizing novel threats but tends to generate higher false-positive rates due to unpredictable variations in network behavior [10].

To improve detection accuracy and adaptability, modern IDS solutions increasingly integrate machine learning (ML) techniques. These approaches enable real-time analysis of network traffic, automatic pattern recognition, and adaptive threat mitigation, making them particularly valuable for complex and dynamic environments such as IoT ecosystems [11].In this paper, we present a machine learning-driven Intrusion Detection System (IDS) specifically designed for IoT environments. Our system utilizes the NSL-KDD dataset to categorize network traffic as either normal or abnormal. To enhance detection efficiency, we employ five feature selection methods (two filter-based, two embedded, and one wrapper method) and evaluate multiple classifiers using established performance metrics. Our goal is to determine the most effective solution for securing IoT networks against cyber threats.The rest of this paper is structured as follows: in Section 2, we analyze the state-of-the-art in IoT and IDS technologies. Section 3 examines key technologies for intrusion detection in IoT environments. In Section 4, we define and evaluate machine learning techniques. Section 5 presents the design and deployment of a customized IDS optimized for our architecture.. Finally, we conclude this work in Section 6 by addressing the growing security challenges in a world increasingly dominated by connected devices.

2. Related work

Anomaly detection and intrusion detection systems have been widely studied in the context of IoT security. Researchers have investigated a range of machine learning and deep learning techniques to enhance detection accuracy and mitigate cyber threats. This section reviews significant contributions in this field.Several studies have focused on anomaly detection in IoT environments. For instance, [12] proposed an anomaly detection method for smart home systems by analyzing user behaviors and home conditions. The approach demonstrated the effectiveness of behavioral data in identifying operational anomalies. Similarly, [13] explored machine learning techniques for detecting anomalies in IoT-based healthcare systems, highlighting the importance of security in medical applications. Additionally, [14] introduced a federated learning approach for detecting intrusions in industrial control systems, addressing privacy concerns by decentralizing data processing.In the context of machine learning-based IDS, [15] presented an IDS specifically designed for mobile IoT networks, utilizing supervised learning techniques to enhance security. Likewise, [16] proposed a deep neural network model for classifying network traffic into two categories, demonstrating high accuracy in detecting malicious activities. Further, [17] analyzed the performance of various machine learning models in detecting network intrusions, comparing their effectiveness in different scenarios.The use of benchmark datasets for IDS evaluation has been another key area of research. A comprehensive review by [18] examined various machine learning algorithms used on the KDD-99 and NSL-KDD datasets, discussing their strengths and limitations. Additionally, [19] provided an in-depth survey of machine learning and deep learning applications for detecting anomalies in IoT systems, highlighting emerging trends and challenges.Moreover, researchers have investigated challenges and potential future directions in IoT security. In [20], a survey highlighted the primary challenges faced by researchers when applying machine learning in IoT security, including dataset limitations,

computational constraints, and evolving attack patterns. Similarly, [21] emphasized the need for adaptive and robust detection mechanisms to address the dynamic nature of IoT threats.In summary, existing research underscores the importance of integrating advanced machine learning techniques into IDS and anomaly detection frameworks for IoT security. However, challenges such as high false-positive rates, scalability issues, and adversarial attacks remain open areas for further investigation.

3. Background

In this section, we explore the technologies used in Intrusion Detection Systems to overcome security challenges in IoT environments.

3.1. Internet of Things

According to Pierre-Jean Benghozi et al. [22], the Internet of Things (IoT) is a system of interconnected networks that facilitates the precise and unique identification of digital entities and physical objects using **standardized electronic identification systems** and **wireless mobile devices**. This technology facilitates the **seamless collection, storage, transfer, and processing of data,** ensuring continuous interaction between the **physical and virtual worlds**. Figure 1 illustrates the IoT architecture, which is structured into six distinct layers: The IoT architecture is composed of six layers, each essential to ensuring the functionality and security of IoT systems.

Perception Layer (Physical Layer): As the foundational layer, it is responsible for sensing and gathering data from the physical environment. It includes sensors, actuators, RFID tags, cameras, and other smart devices that capture environmental parameters like temperature, motion, light, and humidity. The collected data is then forwarded to the network layer or additional analysis and handling.

Network Layer: This layer transfers data from the perception layer to the processing unit or cloud servers. It relies on communication technologies, including Wi-Fi, Bluetooth, Zigbee, 5G, LPWAN (LoRa, NB-IoT), and satellite networks. Ensures secure and efficient data transmission to prevent unauthorized access.

Middleware layer (Data Processing): acts as a bridge between the Network Layer and the Application Layer. It is responsible for data filtering, processing, and storage before delivering insights to end-users. This layer integrates edge computing, cloud computing, and IoT gateways to manage massive data flows efficiently. It also ensures interoperability between different IoT devices and platforms while implementing security mechanisms like encryption and access control. By reducing latency and optimizing data management, the Middleware Layer plays a crucial role in facilitating real-time decision-making and ensuring system scalability.

Application Layer: Provides user-specific applications based on the processed IoT data. Examples include smart home automation, healthcare monitoring, industrial IoT (IIoT), and intelligent transportation systems. It ensures that IoT services are accessible and user-friendly.

Business Layer: focuses on **data interpretation, decision-making, and monetization** of IoT services. It transforms raw data into **actionable insights** using AI and analytics, defines **business models and strategies,** ensures **regulatory compliance** (e.g., GDPR, HIPAA), and enhances **user experience** through dashboards and applications. This layer plays a crucial role in aligning IoT solutions with **industry needs,** optimizing operations in **smart cities, healthcare, and industrial IoT, an**d driving overall business value.Security Layer (Trust & Privacy): This layer ensures data confidentiality, integrity, and authentication across all IoT layers. It includes encryption, access control, anomaly detection, intrusion detection systems (IDS), and blockchain-based security mechanisms. Protects IoT networks from cyber threats, unauthorized access, and data breaches. Each layer in this architecture plays a fundamental role in ensuring the efficiency, scalability, and security of IoT ecosystems.

3.2. Intrusion Detection System (IDS)

An IDS is a cybersecurity solution that combines both hardware and software to monitor network traffic in a stealthy and real-time manner. Its main role is to identify and assess attempts of unauthorized access to a network or host system. By identifying potential threats, an IDS helps to prevent intrusions that could compromise key security principles such as confidentiality, data integrity, and system availability. This proactive defense mechanism plays a crucial role in cybersecurity, enabling organizations to respond swiftly to potential cyber threats [24]. Generally, Intrusion Detection Systems are classified into two primary types: Network-based IDS, which monitor network traffic for security threats, and Host-based

IDS, which focus on detecting intrusions at the host level [24, 25].

3.2.1. Network-based IDS (NIDS)

An NIDS monitors network traffic to detect intrusions by analyzing packets across the network, transport, and application layers. It operates in stealth mode by placing network interface cards in promiscuous mode, allowing it to passively monitor traffic without being detected. NIDS sensors are positioned both outside the network to analyze attack attempts and inside the firewall to inspect internal traffic. Common NIDS tools include Snort, Bro, Suricata, Enterasys, Check Point, and Tipping Point.The advantages of NIDS include its ability to detect scans efficiently using signatures, remain invisible to attackers, and enhance network security. However, it has limitations, such as ineffectiveness if improperly positioned, an inability to inspect encrypted traffic, and bandwidth issues when handling high traffic volumes.

3.2.2. Host-based IDS (HIDS)

An HIDS is deployed on individual hosts (workstations, servers) to analyze incoming and outgoing network traffic and system logs (syslog, lastlog, wtmp) for suspicious activities such as denial-of-service (DoS) attacks, unauthorized access attempts, malicious code execution, Trojan horses and buffer overflow attacks. Notable HIDS solutions include AIDE, Chkrootkit, DarkSpy, Fail2ban, IceSword, OSSEC, and Rkhunter. HIDS excels in detecting attacks in encrypted traffic, preventing damage, and acting as a last line of defense for threats missed by NIDS. However, it is vulnerable to DoS attacks, consumes significant system resources (CPU, memory), and struggles with scan detection. Other IDS types include Hybrid IDS, Knowledge-based Intrusion Prevention Systems (KIPS), Collaborative IDS (CIDS), and Wireless IDS (WIDS).

3.3. Machine learning techniques

Machine Learning (ML), also known as **automated learning** or **artificial learning**, is a subset of Artificial Intelligence (AI) that allows systems to learn from data rather than relying on explicit programming [26]. Unlike traditional rule-based programming, ML algorithms iteratively analyze data, identifying patterns, and generate predictions or decisions autonomously, without human involvement.The **machine learning process** involves training algorithms on large datasets to build predictive models. As these algorithms ingest training data, they refine their accuracy, improving their ability to generate reliable insights. A **machine learning model** is the final product of this training phase-once trained, it can process new input data and produce **intelligent outputs [27].**

Figure 2. shows a structured overview of machine learning, categorizing it into three primary types: supervised learning, unsupervised learning, and reinforcement learning.

- **Supervised learning**, which relies on labeled data, includes **classification** (e.g., **decision trees, neural networks**) and **regression** (e.g., **linear regression, support vector machines**).
- **Unsupervised learning**, based on unlabeled data, involves **clustering** (e.g., **k-means, hierarchical clustering**) and **dimensionality reduction** (e.g., **PCA, LDA**).
- **Reinforcement learning** focuses on **decision-making through trial and error**, utilizing techniques such as **Q-learning and SARSA**.

4. Proposed model

This section presents our intrusion detection model, which is based on machine learning techniques and thoroughly evaluated using the most well-known performance metrics.Figure 3 illustrates the architecture of our model, depicting a machine learning classification pipeline structured into three main steps.

1. **Data Preprocessing (Step 1):** The raw dataset undergoes preprocessing, which includes handling missing data, removing duplicates, transforming data, scaling, and encoding categorical variables. This step ensures that the dataset is clean and structured for model training.
2. Training Phase (Step 2): The preprocessed dataset is split into a training set, where various machine learning classifiers, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression (LR), among others, are trained to identify patterns and make accurate predictions.
3. Testing Phase (Step 3): The trained models are evaluated on a separate testing set to assess their performance, where various classifiers, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and others, are applied to the test data.

4. Evaluation and Classification Result: The models' performance is assessed using classification metrics, and the final classification results are generated to determine the overall effectiveness of each model.

This structured approach ensures effective training, testing, and evaluation of machine learning classifiers for classification tasks.

4.1. Dataset

In this work, we used the NSL-KDD dataset, chosen for its enhancements over the original KDD'99 dataset. While it still has some limitations, as noted by McHugh [28], it remains a widely accepted benchmark for evaluating IDS due to the lack of publicly available datasets. One of its key advantages is the balanced number of records in both training and testing sets, allowing researchers to conduct experiments without the need for random sampling, ensuring consistency and comparability of results across different studies [29, 30].Compared to KDD'99, NSL-KDD offers several improvements: it removes redundant records from the training set, preventing classifiers from developing a bias toward frequently occurring samples. Moreover, it eliminates duplicate entries in the test set, reducing the advantage of models that perform well on repeated patterns. The dataset also balances the number of records across different difficulty levels, enhancing the evaluation of machine learning techniques. These improvements contribute to a more reliable and standardized benchmark for IDS research [28, 31]. Figure 4 illustrates the different types of attacks in the NSL-KDD dataset, grouped into four main categories:

Probing Attack (Prob): The attacker gathers information about the network to identify vulnerabilities and bypass security measures.

Denial of Service Attack (DOS): This attack disrupts services, rendering them unavailable to legitimate users.

Remote to User Attack (R2L): The attacker exploits security weaknesses to gain unauthorized access as a local user.

User to Root Attack (U2R): An attacker escalates privileges from a standard user account to obtain root access to the system.The NSL-KDD dataset contains 125,972 rows and 43 columns of different types. However, it is not in a suitable state for processing, requiring preprocessing.

4.2. Data Preprocessing

Preprocessing is an essential step in preparing datasets for machine learning models. It helps address issues such as missing values, inconsistencies, and noise, ensuring cleaner and more reliable data. By refining the dataset, machine learning models can achieve better accuracy, improved generalization, and faster processing times. Techniques like feature selection, normalization, and outlier handling contribute to maintaining consistency and enhancing interpretability. Well-processed data allows models to learn effectively and make more precise predictions.

4.2.1. Missing Data Handling

Missing data, or absent values, occur when no value is recorded for a specific variable in a given observation. In other words, a feature lacks an assigned value. This can negatively impact data processing, leading to inaccurate results.

4.2.2. Handling Duplicates

A duplicate value occurs when all the attributes of at least one row are identical to those of another row. The NSL-KDD dataset is an enhanced version of KDD99, where duplicates have been removed to prevent artificially inflated prediction accuracy.

4.2.3. Data Transformation

This step involves reviewing the available data to determine which features are useful, which are not, and which may be considered anomalies. Initially, we removed the "level" column, which describes the difficulty level of the attack, as it is not needed for our classification.

4.2.4. Scaling

Features in machine learning can have significantly different values, which may impact classification performance. This variation in scale can lead to reduced accuracy, making scaling essential to adjust features to a common range.

4.2.5. Encoding

This step involves converting the categorical features into numerical values to ensure proper functionality and prevent errors. After completing the preprocessing, the target variable (‘attack’) is categorized using a binary classification (0, 1) by introducing a new feature called ‘intrusion.’ If an intrusion is detected, it is labeled as 0; otherwise, it is assigned a value of 1. This transformation allows for a clearer visualization of detected attacks through a pie chart, effectively representing the data and enhancing interpretation.Figure 5 shows that the TCP protocol is consistently affected by DoS attacks, while the ICMP protocol is primarily

targeted by Probe attacks. This indicates that attacks aiming to render IoT services unavailable generally infect reliable transport protocols, such as TCP, which operate in a connected mode. On the other hand, attacks designed to gather information from IoT devices typically target the ICMP protocol, which is responsible for managing error-related information in connected machines.

4.3. Classifiers

Table 1 illustrates a selection of the most frequently used classifiers for developing machine learning-based IDS, as referenced in [29].In our work, we selected the following classifiers: Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT) , Neural Networks (MLP), Random Forest (RF), Logistic Regression (LR), AdaBoost and Bagging.

4.4. Evaluation Metrics

Table 2 illustrates a selection of widely used evaluation metrics for IDS models utilizing machine learning, as referenced in [29].
Our selection includes 'training time,' 'testing time,' 'confusion matrix', 'accuracy,' 'recall,' 'precision,' 'F-measure, 'Receiver Operating Characteristic - Area Under Curve (ROC-AUC)', and 'ROC curve'.

4.5. Feature Selection

Feature selection aims to identify the most relevant features that contribute to building effective machine learning models for a given problem. This process enhances model performance by reducing dimensionality, improving interpretability, and minimizing overfitting. Feature selection techniques are generally classified into three categories:

- **Filtering methods:** Evaluate feature relevance using statistical measures such as information gain and correlation coefficient.
- **Wrapper methods:** Utilize iterative processes like Recursive Feature Elimination (RFE) to select the optimal feature subset.
- **Embedded methods:** Integrate feature selection within the model training process, employing techniques such as LASSO regularization and random forest feature importance.

5. Evaluations and results

This section outlines the results of our experiments. First, we evaluate the classification performance metrics of the dataset using all available features. Next, we reduced the dataset dimensions using the three well-known feature selection methods mentioned earlier and assessed each algorithm using the optimal hyper-parameters.

5.1. Performance Evaluation without Feature Selection

We evaluate the dataset's classification performance metrics without applying feature selection, utilizing all available features.
Table 3 illustrates the results of classifiers without any feature selection method in terms of various metrics.
Figure 6 displays the experimental results in terms of accuracy without applying any feature selection method.
The experimental results showed that the Random Forest (RF) model outperformed all other models, achieving an accuracy of 99.89% and an AUC-ROC of 99.86. It also had a fast testing time of 14 seconds and correctly classified 99.97% of the samples. However, its training time was higher (220 seconds) compared to the Bagging model (26 seconds), which achieved nearly the same performance. The Bagging model correctly classified 99.96% of the samples, with an accuracy of 99.87% and an AUC-ROC of 99.87.

5.2. Performance Evaluation with Feature Selection

Now, we apply three feature selection techniques to reduce the dataset dimensions and evaluate each algorithm with the best hyper-parameters.

5.2.1. Correlation coefficient method

Correlation quantifies the linear relationship between two or more variables, allowing one variable to be predicted based on another. In feature selection, correlation is used under the assumption that relevant features should exhibit a strong correlation with the target variable while remaining uncorrelated with each other.To apply this method, an absolute correlation threshold (e.g., 0.5) is typically set for selecting features. If two predictive variables are highly correlated, the one with the lower correlation with the target variable can be discarded to minimize redundancy and improve model efficiency.Table 4 illustrates the results of classifiers with the correlation coefficient method in terms of various metrics.Figure 7 presents the experimental results in terms of accuracy using the correlation coefficient method.
After applying feature selection using the correlation coefficient, the feature set was narrowed down to 27 variables, which had a coefficient lower than 0.5. The results showed that the Bagging and Random Forest models were both efficient and fast in testing time (14s and 1s, respectively). They outperformed other models, achieving the same

accuracy of 99.87% and an identical AUC-ROC of 99.86. Additionally, they correctly classified nearly the same percentage of samples (99.97% and 99.96%). However, there was a significant difference in training time between the Random Forest and Bagging models.

5.2.2. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a feature selection technique that iteratively reduces the feature set to identify the most relevant variables. It operates by training an estimator that assigns weights to features, such as the coefficients of a linear model or the feature_importances attribute of tree-based models.The process begins with the full set of features, where the estimator evaluates their importance. The least significant features are then systematically removed in each iteration, refining the subset until only the most influential features remain. This method helps enhance model performance by eliminating irrelevant or redundant variables.Table 5 illustrates the results of classifiers using the Recursive Feature Elimination in terms of various metrics. Figure 8 presents the experimental results in terms of accuracy using RFE selection.The experimental results of the proposed method showed that Random Forest is the most effective model, achieving the best performance among all classifiers. It obtained an accuracy of 99.89%, an AUC-ROC of 99.85, and correctly classified 99.97% of the samples. However, the training time increased compared to previous feature selection methods.

5.2.3. LASSO Regularization

Regularization involves adding a penalty to the model parameters to limit its complexity and prevent overfitting. In linear model regularization, this penalty is applied to the coefficients associated with each predictor. Among the different types of regularization, LASSO (L1 regularization) has the unique property of shrinking some coefficients to zero. As a result, the corresponding features can be effectively removed from the model, improving interpretability and reducing dimensionality.LASSO selection reduced the feature set to 92. The performance of classifiers using the LASSO Regularization in terms of various metrics are represented in Table 6. Figure 9 presents the experimental results in terms of accuracy using LASSO Regularization.The results obtained after LASSO selection showed that Random Forest and Bagging achieved almost identical performance, with the same accuracy (99.89%), AUCROC = 99.88, and nearly the same number of well-classified samples (99.97% and 99.96%). The only difference between them was the training time, which was significantly higher for Random Forest (212 s) compared to Bagging (18 s). This indicates that Random Forest is highly effective, while Bagging is both effective and fast.

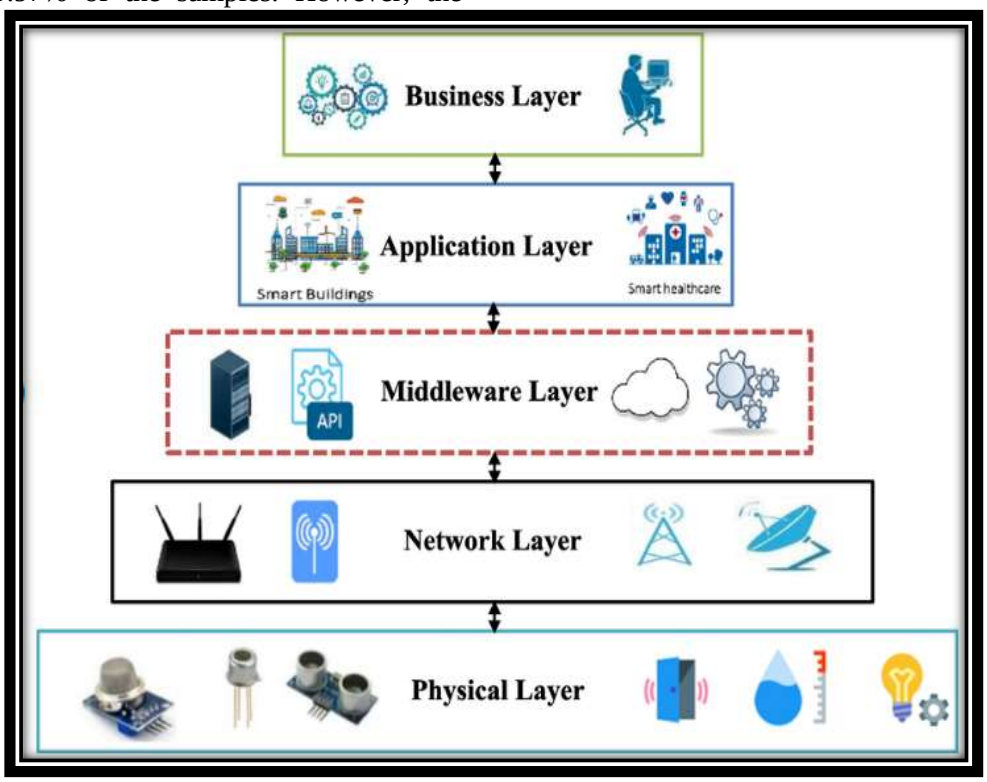

Figure 1. *The IoT architectural model [23]*

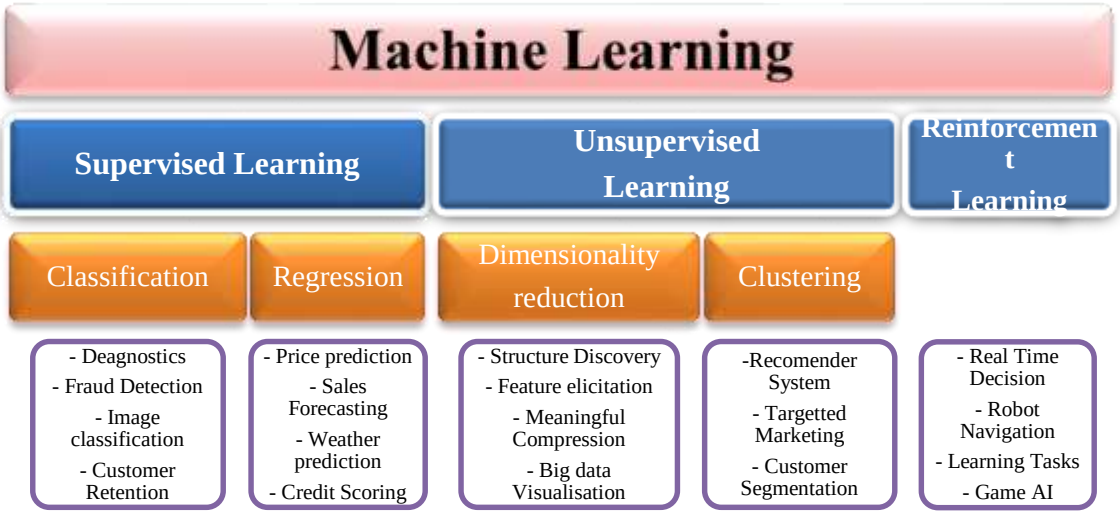

Figure 2: *Machine Learning*

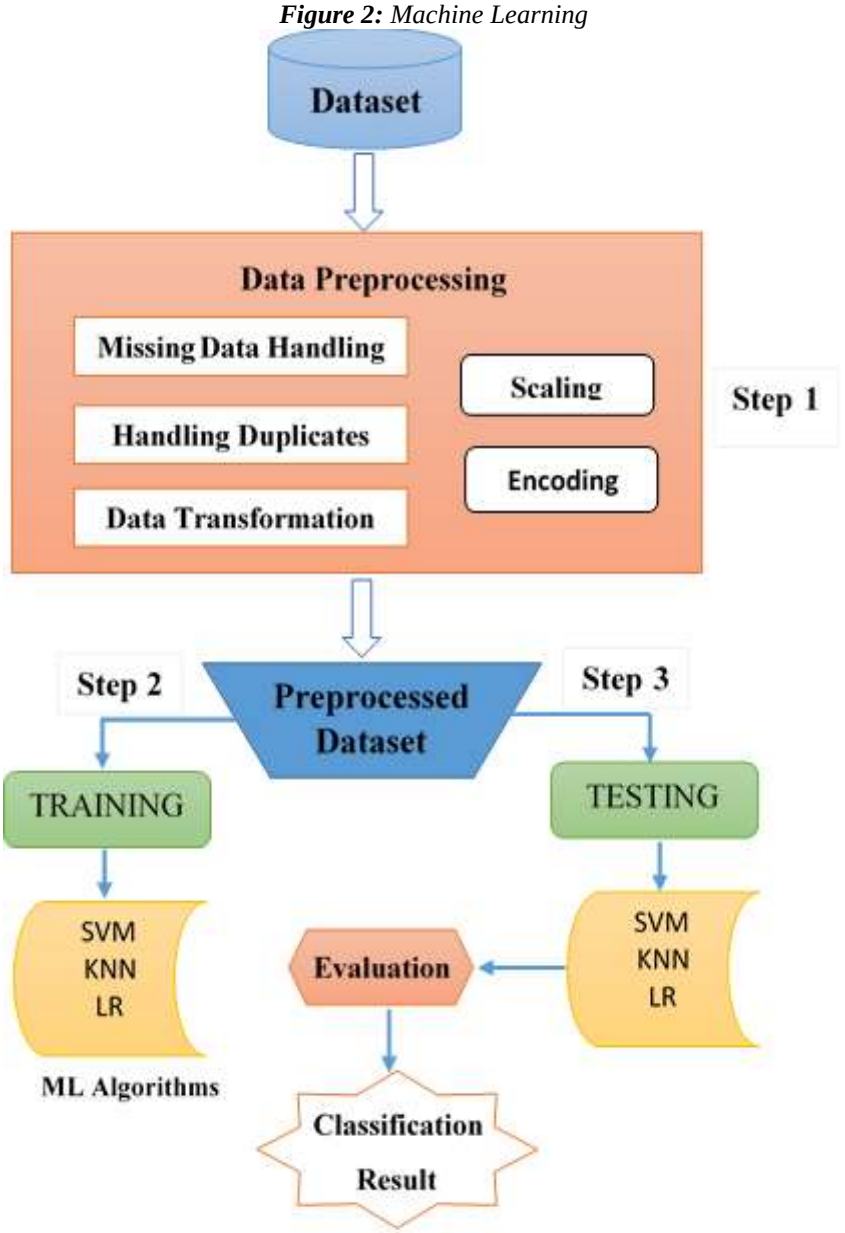

Figure 3: *Model Architectural*

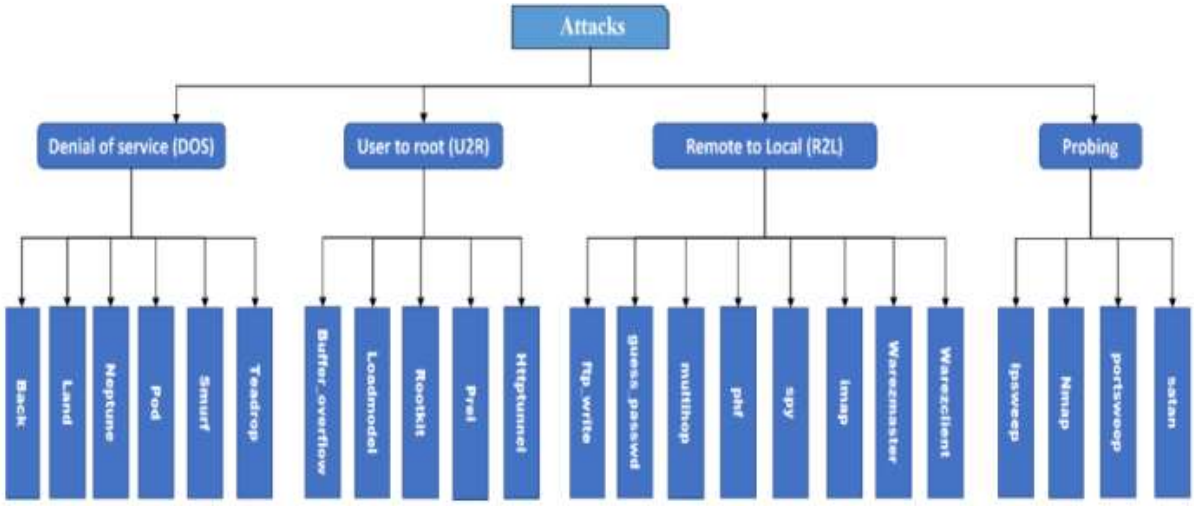

Figure 4. *Categories of attacks in the NSL-KDD dataset [30]*

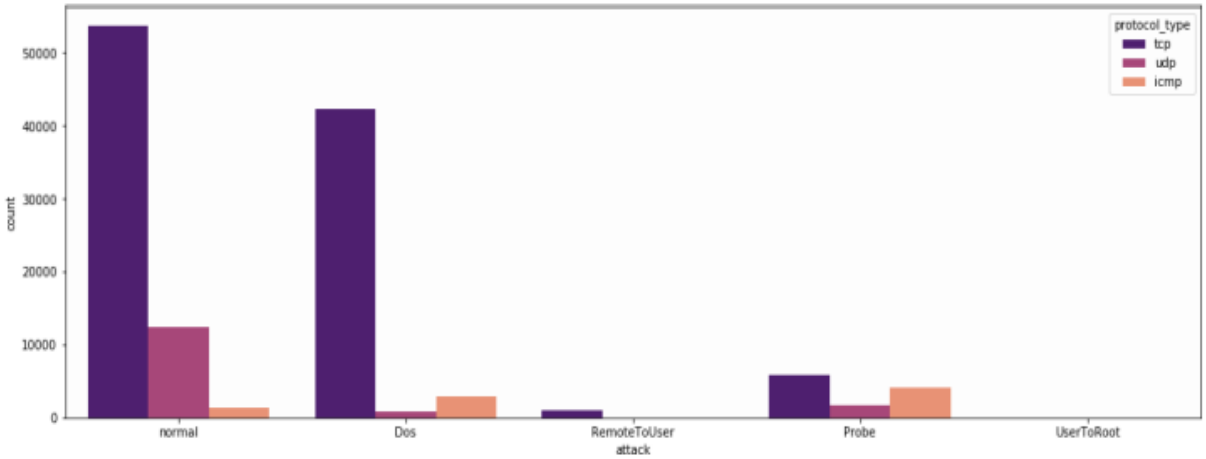

Figure 5. *Protocols affected by the attacks.*

Table 1. *The most commonly used classifiers according to [29].*

Classifier	Article Count
Support Vector Machines	33
Naive Bayes	31
None (Not compared with other methods)	21
Decision Tree(J48)	21
k-nearest neighbors	20
Literature (results are given without experimental comparison)	19
Decision Tree	14
NeuralNetworks(MultiLayerPerceptron)	14
Bayesian Network	13
Random Forest	10
NeuralNetworks(SelfOrganizingMap)	7
NeuralNetworks(RadialBasisFunction)	7
K-Means	6
Rule Based Learner(JRipper)	5
Adaboost	5
Naive Bayes Tree	4
Decision Tree(C4.5)	4
PART	4
Decision Tree(CART)	4
Bagging	3
Random Tree	3

Table 2. *The most commonly used performance metrics according to [33].*

Performance Metric	Article Count
Detection Rate	134
False Positive(False Alarm)	70
Training Time	44
Testing Time	28
ROC-Curve	24
False Negative	22
Confusion Matrix (5 class)	20
True Positive	20
Error Rate	13
Precision	13
F-Measure	13
Recall	12
True Negative	11
Number of Selected Features	10
Correlation Coefficient	9
Cost Per Example	9
ROC-Area Under Curve	8
Sensitivity	7
Specificity	7
Root Mean Square Error	6
None*	5
Memory Usage	5

Table 3. *Classification results without feature selection.*

	Accuracy	precision	recall	F1-score	AUC	Well-Class	Misclassified	Train Time	Test Time
Random forest	99.89%	99.83%	99.95%	99.89%	99.86	99.97%	0.03%	220s	14s
Decision tree	99.82%	99.85%	99.82%	99.83%	99.82	99.95%	0.05%	4s	1s
SVM	97.51%	97.34%	98.01%	97.67%	97.47	97.67%	2.33%	478s	43s
KNN	99.60%	99.61%	99.64%	99.62%	99.59	99.73%	0.27%	0.10s	207s
Naive bayes	85.41%	78.57%	99.84%	87.94%	84.40	85.45%	14.55%	0.78s	0.53s
MLP	82.34%	76.22%	97.14%	85.42%	81.30	82.31%	17.69%	678s	5s
Adaboost	99.41%	99.27%	99.63%	99.45%	99.39	99.49%	0.51%	37s	6s
Logistique regression	97.26%	96.48%	98.05%	97.44%	97.20	97.34%	2.66%	3s	0.24s
bagging	99.87%	99.89%	99.87%	99.88%	99.87	99.96%	0.04%	26s	1s

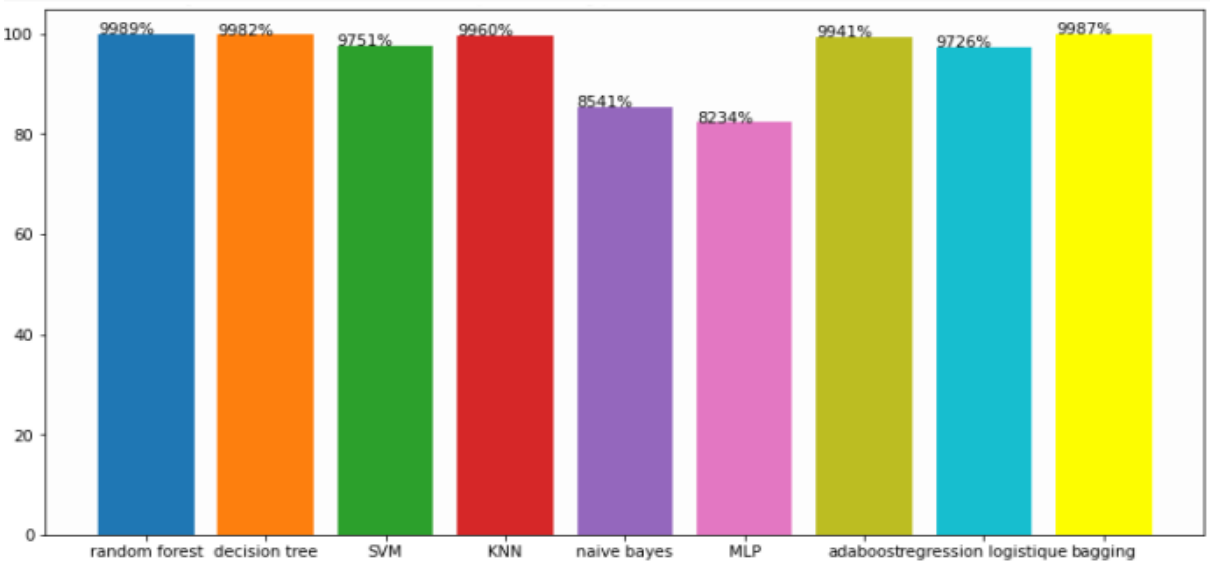

Figure 6. *Model accuracy without applying any feature selection method.*

Table 4. *Performance Evaluation with Feature Selection*

	Accuracy	precision	recall	F1-score	AUC	Well Classified	Misclassified	Train Time	Test Time
Random forest	99.87%	99.82%	99.93%	99.88%	99.86	99.97%	0.03%	499s	12s
Decision tree	99.81%	99.79%	99.86%	99.82%	99.80	99.95%	0.05%	6s	0.14s
SVM	96.84%	96.76%	97.33%	97.05%	96.80	97.06%	2.94%	435s	28s
KNN	99.55%	99.58%	99.56%	99.57%	99.54	99.67%	0.33%	0.15s	148s
Naive bayes	70.72%	64.55%	99.84%	78.41%	68.68	71.06%	28.94%	0.64s	0.41s
MLP	84.50%	77.88%	99.01%	87.19%	83.48	84.53%	15.47%	353s	5s
Adaboost	99.30%	99.19%	99.50%	99.35%	99.28	99.34%	0.66%	54s	4s
Logistique regression	96.67%	96.29%	97.50%	96.89%	96.61	96.80%	3.20%	5s	0.23s
bagging	99.87%	99.87%	99.89%	99.88%	99.86	99.96%	0.04%	16s	1s

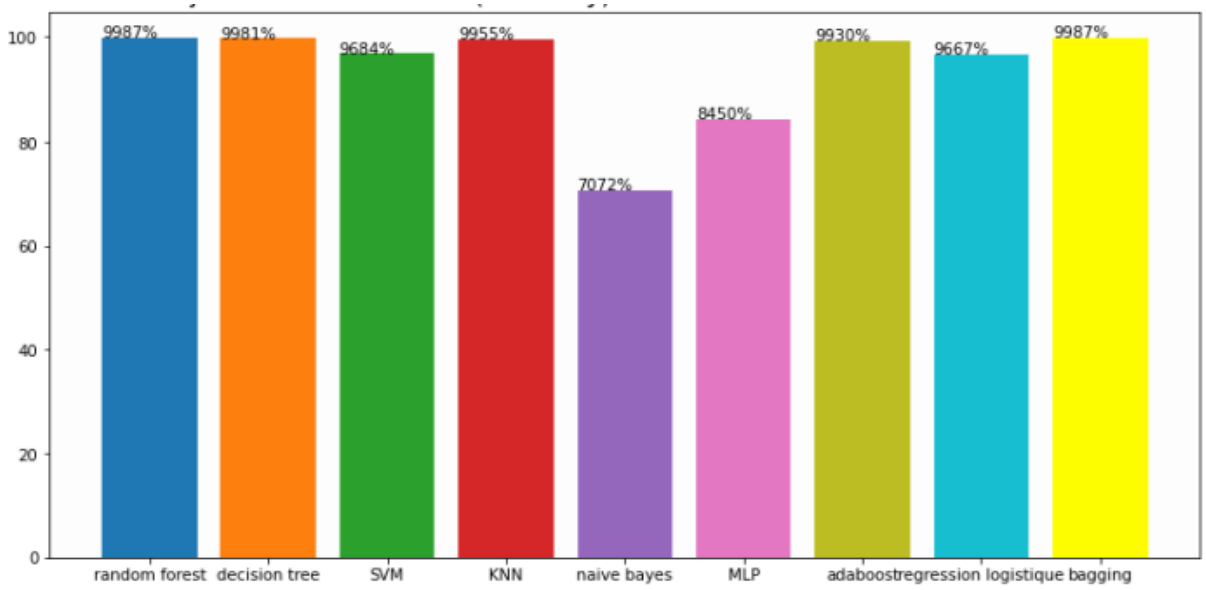

Figure 7. *Model accuracy using correlation coefficient selection.*

Table 5. *Results of classifiers using RFE selection.*

	Accuracy	precision	recall	F1-score	AUC	Well Classifed	Misclassified	Train Time	Test Time
Random forest	99.89%	99.85%	99.94%	99.90%	99.88	99.97%	0.03%	197 s	9 s
Decision tree	99.83%	99.84%	99.83%	99.84%	99.82	99.96%	0.04%	4 s	0.20 s
SVM	97.31%	97.27%	97.69%	97.48%	97.28	97.49%	2.51%	302 s	32 s
KNN	99.58%	99.62%	99.59%	99.61%	99.58	99.71%	0.29%	0.15 s	175 s
Naive bayes	93.38%	90.51%	97.84%	94.03%	93.07	93.44%	6.56%	0.40 s	0.34 s
MLP	67.19%	64.07%	96%	76.85%	67.32	69.10%	30.90%	295 s	2 s
Adaboost	99.25%	99.20%	99.40%	99.30%	99.24	99.37%	0.63%	35 s	3 s
Logistique regression	96.86%	96.47%	97.67%	97.07%	96.80	96.99%	3.01%	3 s	0.24 s
bagging	99.85%	99.89%	99.84%	99.86%	99.85	99.95%	0.05%	18 s	0.82 s

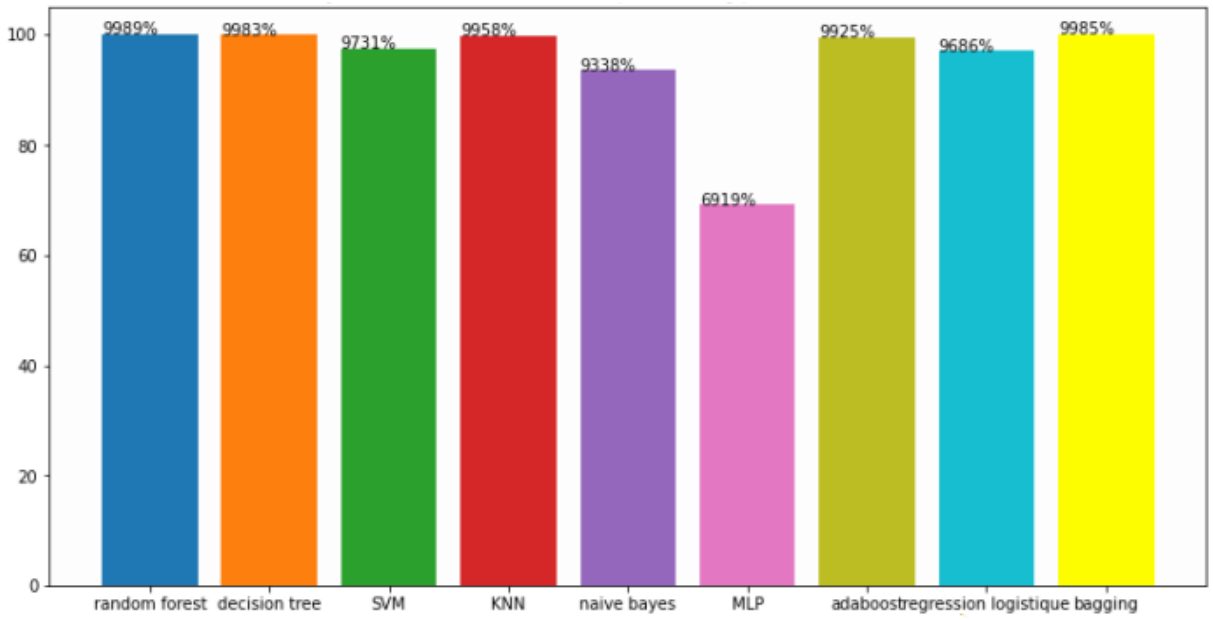

Figure 8. *Comparison of model accuracy using RFE.*

Table 6. *Results of classifiers using LASSO Regularization.*

	Accuracy	precision	recall	F1-score	AUC	well Classified	Missclassified	Train Time	Test Time
Random forest	99.89%	99.85%	99.95%	99.90%	99.88	99.97%	0.03%	212s	11s
Decision tree	99.80%	99.80%	99.83%	99.81%	99.79	99.95%	0.05%	4s	0.20s
SVM	97.53%	97.38%	98.01%	97.69%	97.49	97.66%	2.34%	261s	23s
KNN	99.61%	99.59%	99.67%	99.63%	99.60	99.73%	0.27%	0.14s	163s
Naive bayes	86.83%	80.35%	99.65%	88.96%	85.93	86.82%	13.18%	0.54S	0.38s
MLP	87.24%	80.98%	99.39%	89.25%	86.93	87.19%	12.81%	224s	2s
Adaboost	99.42%	99.28%	99.63%	99.46%	99.40	99.50%	0.50%	57s	4s
Logistique regression	97.22%	96.83%	98%	97.41%	97.17	97.32%	2.68%	4s	0.25s
bagging	99.89%	99.89%	99.90%	99.90%	99.88	99.96%	0.04%	18s	1s

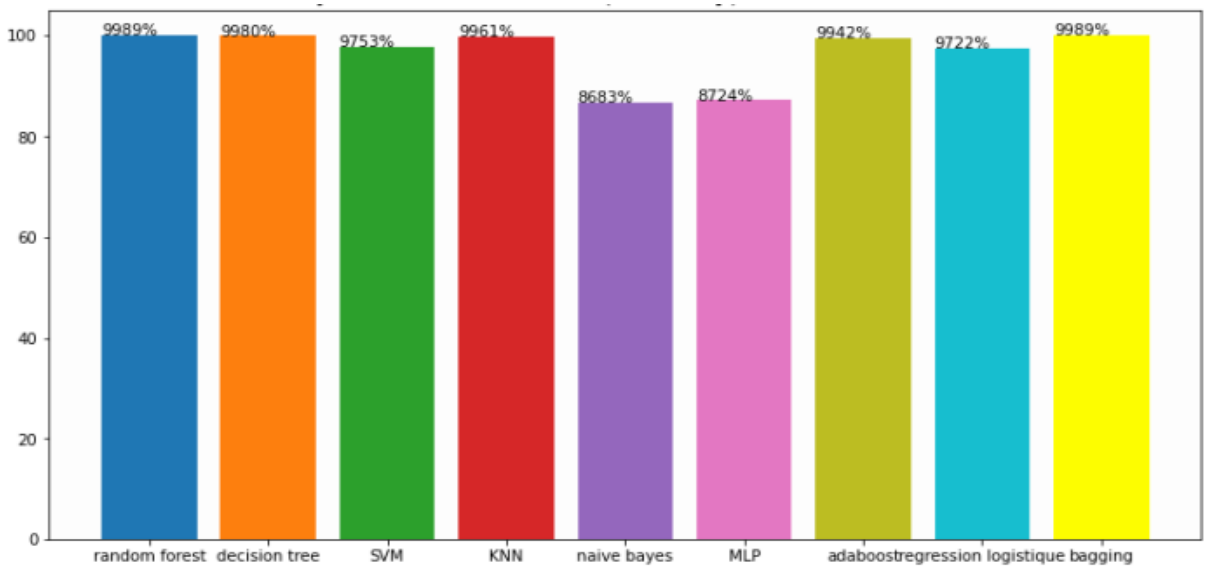

Figure 9. *Model accuracy using LASSO selection.*

6. Conclusions

In this study, we introduced an Intrusion Detection System (IDS) for Internet of Things (IoT) networks, leveraging machine learning techniques to enhance security against cyber threats. We utilized the NSL-KDD dataset to classify network traffic and applied multiple feature selection methods to optimize model performance. Our experiments showed that Random Forest and Bagging classifiers achieved the highest accuracy, with 99.89% accuracy and AUC-ROC of 99.88, proving their effectiveness in detecting anomalies in IoT environments. Additionally, feature selection techniques significantly improved the efficiency of our models by reducing dimensionality while maintaining high detection performance.

Despite the promising results, certain challenges remain, such as high training time for complex models, potential false positives, and the need for real-time adaptability in dynamic IoT environments. Addressing these limitations is crucial for deploying IDS solutions in practical IoT applications.

Future research will focus on enhancing the adaptability and efficiency of IDS in IoT networks by integrating:

- Real-time anomaly detection using online learning techniques to continuously adapt to new attack patterns.
- Federated learning approaches to improve security without centralized data collection, preserving privacy in distributed IoT environments.
- Lightweight models optimized for resource-constrained IoT devices, ensuring faster inference while maintaining high detection accuracy.
- Hybrid IDS solutions combining signature-based and anomaly-based detection to improve detection accuracy and reduce false positives.
- Integration of deep learning techniques, such as graph neural networks (GNNs) or transformers, to enhance feature extraction and improve detection of sophisticated cyberattacks.
- By addressing these aspects, we aim to develop an adaptive, scalable, and efficient IDS solution tailored for the evolving landscape of IoT security.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Kumar, S., Tiwari, P. & Zymbler, M. Internet of Things is a revolutionary approach for future technology enhancement: a review. J Big Data 6, 111 (2019). https://doi.org/10.1186/s40537-019-0268-2

[2] Elgazzar, K., Khalil, H., Alghamdi, T., Badr, A., Abdelkader, G., Elewah, A., & Buyya, R. (2022). Revisiting the internet of things: New trends, opportunities and grand challenges. Frontiers in the Internet of Things, 1, 1073780. https://doi.org/10.3389/friot.2022.1073780

[3] Xia, F., Yang, L. T., Wang, L., & Vinel, A. (2012). Internet of things. International journal of communication systems, 25(9), 1101.

[4] Nižetić, S., Šolić, P., Gonzalez-De, D. L. D. I., & Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. Journal of cleaner production, 274, 122877.

[5] Atzori, L., Iera, A., Morabito, G. (2010). The Internet ofThings: A survey. Computer Network, 54(15): 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010.

[6] Fenanir, S., Semchedine, F., & Baadache, A. (2019). A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things. Revue d'Intelligence Artificielle, 33(3).

[7] Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88, 10–28.

[8] Abomhara, M., & Køien, G. M. (2015). Cyber security and the Internet of Things: Vulnerabilities, threats, intruders and attacks. Journal of Cyber Security and Mobility, 4(1), 65–88.

[9] Jyothsna, V. V. R. P. V., Prasad, R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. International Journal of Computer Applications, 28(7), 26-35.

[10] Kikissagbe, B. R., & Adda, M. (2024). Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review Electronics, 13(18), 3601.

. https://doi.org/10.3390/electronics13183601

[11] Fenanir, S., Semchedine, F., Harous, S., & Baadache, A. (2020). A Semi-supervised Deep Auto-encoder Based Intrusion Detection for IoT. Ingenierie des Systemes d'Information, 25(5).

[12] Yamauchi, M., Ohsita, Y., Murata, M., Ueda, K., & Kato, Y. (2020). Anomaly detection in smart home operation from user behaviors and home conditions. IEEE Transactions on Consumer Electronics, 66(2), 183–192. https://doi.org/10.1109/TCE.2020.2981636

[13] Khan, M. M., & Alkhathami, M. (2024). Anomaly detection in IoT-based healthcare: Machine learning for enhanced security. Scientific Reports, 14, 5872. https://doi.org/10.1038/s41598-024-56126-x

[14] Harahsheh, Maryam & Chen, Chung-Hao. (2023). A Survey of Using Machine Learning in IoT Security and the Challenges Faced by Researchers. Informatica. 47. 10.31449/inf.v47i6.4635.

[15] Amouri, Amar & Alaparthy, Vishwa & Morgera, Salvatore. (2020). A Machine Learning Based Intrusion Detection System for Mobile Internet of Things. Sensors. 20. 10.3390/s20020461.

[16] Abdulla, Shubair & Alashoor, Ahmed. (2020). An Artificial Deep Neural Network for the Binary Classification of Network Traffic. International Journal of Advanced Computer Science and Applications. 11. 10.14569/IJACSA.2020.0110150.

[17] Alsaeedi, Abdullah & Khan, Mohammad. (2019). Performance Analysis of Network Intrusion Detection System using Machine Learning. International Journal of Advanced Computer Science and Applications. 10. 671-678. 10.14569/IJACSA.2019.0101286.

[18] Devi, Ravipati & Abualkibash, Munther. (2019). Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper. International Journal of Computer Science and Information Technology. 11. 65-80. 10.5121/ijcsit.2019.11306.

[19] Rafique, Saida & Abdallah, Amira & Musa, Nura & Murugan, Thangavel. (2024). Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection—Current Research Trends. Sensors. 24. 1968. 10.3390/s24061968.

[20] Harahsheh, Maryam & Chen, Chung-Hao. (2023). A Survey of Using Machine Learning in IoT Security and the Challenges Faced by Researchers. Informatica. 47. 10.31449/inf.v47i6.4635.

[21] Huong, Truong & Ta, Phuong Bac & Long, Dao & Luong, Tran & Dan, Nguyen & Quang, Le & Cong, Le & Thắng, Bùi & TRAN, Kim Phuc. (2021). Detecting Cyberattacks using Anomaly Detection in Industrial Control Systems: A Federated Learning approach. Computers in Industry. 132. 103509. 10.1016/j.compind.2021.103509.

[22] Benghozi, P. J., Bureau, S., & Massit-Folléa, F. (2015). L'Internet des objets/The Internet of Things: Quels enjeux pour l'Europe?/What Challenges for Europe?. Les Editions de la MSH.

[23] Alaghbari, Khaled & Md Saad, Mohamad Hanif & Hussain, Aini & Alam, Muhammad Raisul. (2022). Complex event processing for physical and cyber security in datacentres - recent progress, challenges and recommendations. Journal of Cloud Computing. 11. 10.1186/s13677-022-00338-x.

[24] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36(1), 16-24.* DOI: 10.1016/j.jnca.2012.09.004

[25] Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). *National Institute of Standards and Technology (NIST), Special Publication 800-94.*

[26] Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1, No. 9). New York: McGraw-hill.

[27] Goodfellow, I. (2016). Deep learning.

[28] NSL-KDDDataset, https://www.unb.ca/cic/datasets/nsl.html, accessed on March 1, 2024.

[29] Ozgür, A., & Erdem, H. (2015). *A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015*.

[30] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (2009). *A detailed analysis of the KDD CUP 99 dataset*. In Proceedings of the 2nd IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). IEEE.

[31] Tomer, V., & Sharma, S. (2022). Detecting IoT attacks using an ensemble machine learning model. *Future Internet*, *14*(4), 102. https://doi.org/10.3390/fi14040102