THE ASIAN BULLETIN OF BIG DATA MANAGMENT
Vol. 6. Issue 1 (2026)
https://doi.org/10.62019/f7v58427

ASIAN BULLETIN OF BIG DATA MANAGEMENT
http://abbdm.com/
ISSN (Print): 2959-0795
ISSN (online): 2959-0809

Anomaly Detection in IoT Using Machine Learning Techniques: A Comparative Study and Voting-Ensemble Approach

Mediha Maroof*, Ayesha Maroof, Ayesha Bano

Chronicle

Article history
Received: Nov 8, 2025
Received in the revised format: Dec 16. 2025
Accepted: Dec 28, 2025
Available online Jan 18, 2026

Mediha Maroof, is currently affiliated with the School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, P.R. China.
Email: medihach6@gmail.com

Ayesha Maroof is currently affiliated with the Department of Computer System Engineering, Mirpur University of Science and Technology, Mirpur, Pakistan.
Email: ayeshamaroof03@gmail.com

Ayesha Bano is currently affiliated with the University of Agriculture, Peshawar, Pakistan.
Email: ayeshaabano345@gmail.com

Corresponding Author*

Abstract

The widespread adoption of Internet of Things systems has connected large numbers of devices through shared networks and cloud services, which has improved automation, monitoring, and operational efficiency. However, this growing connectivity also enlarges the security boundary and creates more opportunities for attackers to inject harmful or abnormal traffic. As a result, it becomes increasingly important to detect malicious behavior early and reliably so that IoT-oriented networks can remain stable, trustworthy, and resilient. In this study, machine learning is used to identify malicious activity from network traffic data. Five supervised learning models are examined, including k nearest neighbor, decision tree, naive Bayes, multilayer perceptron, and support vector machine. Since each classifier has different strengths and weaknesses, an additional voting-based ensemble model is developed to combine their predictions into a single decision. The evaluation is carried out on the KDD Cup 1999 benchmark dataset using a conventional training and testing procedure. To ensure a fair comparison, performance is assessed through accuracy, precision, recall, and F score, which together provide a balanced view of detection quality. The experimental findings show that the voting ensemble model produces the most reliable performance among the tested approaches. In the full label evaluation setting, the ensemble achieves 98.87 percent accuracy, exceeding the best single classifier performance. In a filtered class setting that removes rare classes to reduce extreme imbalance, the ensemble accuracy increases to 99.87 percent. These findings indicate that combining diverse learners improves robustness and reduces the risk of relying on a single model under varying traffic patterns, providing a strong baseline for intrusion detection in IoT network environments.

Keywords: Internet of Things, Intrusion Detection System, Machine Learning, Ensemble Learning, Cybersecurity.

INTRODUCTION

The Internet of Things has become a foundational component of modern computing because it connects large numbers of physical devices to networks and cloud services. Through embedded sensors and communication modules, these devices can observe their surroundings, exchange information, and initiate actions with limited human intervention. As IoT deployments continue to expand in smart homes, industrial automation, healthcare monitoring, agriculture, and transportation, the amount and diversity of generated traffic have increased substantially. Consequently, ensuring the confidentiality, integrity, and availability of IoT services is now a practical necessity for maintaining reliability, privacy, and operational continuity. Despite its benefits, the IoT ecosystem faces persistent security challenges. Many devices are designed with strict

constraints on processing capability, memory, and energy consumption, which limits the feasibility of computationally expensive protection mechanisms. In addition, IoT environments are highly heterogeneous because they integrate different device vendors, operating systems, and communication protocols, often with inconsistent security configurations. Moreover, IoT devices are frequently deployed in unattended or semi trusted settings, where weak authentication, delayed patching, and misconfiguration can be exploited. As a result, IoT networks have become attractive targets for adversaries, and they are exposed to a wide range of threats, including denial of service behavior, probing and scanning activities, unauthorized access attempts, and other anomalous traffic patterns. Intrusion detection systems are widely used to monitor network activity and identify suspicious behavior. However, conventional IDS techniques often struggle in IoT scenarios. Signature-based systems can perform well when attack patterns are already known, yet they typically adapt slowly when adversaries change tactics or introduce new variants.

Meanwhile, the dynamic and large-scale nature of IoT traffic makes static rules difficult to maintain and increases the risk of both missed detections and false alarms. These limitations have motivated the growing use of machine learning for intrusion detection because data driven models can learn complex relationships among traffic features and can support automated decision-making beyond handcrafted signatures. Even so, relying on a single classifier can introduce practical weaknesses. Some models may favor majority classes, others may overfit to the training distribution, and several may degrade when traffic characteristics shift over time. Furthermore, no single learning algorithm is consistently best across different types of attacks or network conditions. For instance, certain models capture nonlinear behavior effectively, whereas others offer faster predictions or better interpretability. Therefore, ensemble learning has emerged as an appealing strategy because it combines complementary models and reduces dependence on one decision boundary.

In particular, voting-based ensembles can improve robustness by balancing the errors of individual learners and improving overall stability. In this study, supervised machine learning is used to examine intrusion detection for IoT-oriented traffic through a structured pipeline that converts network records into security decisions. Several widely used classifiers are evaluated alongside a voting-based ensemble that aggregates model predictions to improve robustness. Experiments are conducted using the KDD Cup 1999 benchmark to provide a reproducible baseline for comparative evaluation; however, the broader goal is to inform intrusion detection design choices for IoT settings where traffic diversity and resource constraints are critical. The main contributions of this work are summarized as follows.

- A structured intrusion detection pipeline for IoT-oriented network traffic that organizes the full workflow from data preparation and feature handling to classification and evaluation, enabling consistent and reproducible experimentation.

- A systematic comparative study of widely used supervised learning models for intrusion detection, including k nearest neighbor, decision tree, naive Bayes, multilayer perceptron, and support vector machine, analyzed under the same experimental conditions.

- A voting-based ensemble intrusion detection model that integrates the outputs of multiple base learners to improve robustness and reduce dependence on any single classifier, thereby strengthening detection stability across diverse traffic patterns.

- A comprehensive multi-metric evaluation using accuracy, precision, recall, and F score, supported by confusion matrix analysis, to provide a detailed understanding of both overall performance and error behavior.

BACKGROUND AND RELATED WORK

The Internet of Things, also called IoT, refers to a communication environment where many devices are connected through wired and wireless networks to exchange data and provide services. These devices are not limited to computers. They include consumer products such as smartphones, wearable devices, and home appliances, as well as industrial equipment used for monitoring, control, and automation. A key idea in IoT is that devices can be uniquely identified, which allows different forms of communication, such as communication between people, between people and devices, and between devices themselves [1],[2]. Because of this capability, IoT has expanded quickly and has become an important part of modern information and communication systems. At the same time, global investment in IoT is expected to exceed one trillion United States dollars, which shows the high economic and strategic value of these technologies. Vermesan et al. described IoT as a strong connection between the physical world and the digital world, where sensing and communication allow software services to observe real environments and support automated actions [3].This link is useful because it enables smart services that respond to data in real time. However, it also increases security exposure. As more devices connect to the network, there are more entry points for attackers, and security weaknesses in one device can affect many others. For this reason, IoT security has become a major research topic, especially for systems that handle personal information, safety critical operations, or large scale industrial processes.

A. Security challenges in IoT networks

IoT systems are vulnerable for several practical reasons. First, many IoT devices are built with limited processing power, small memory, and low energy capacity. These limits reduce the ability of devices to run complex security tools or continuous monitoring functions. Second, IoT systems are highly diverse because devices come from different vendors, run different operating systems, and use different communication standards. This diversity makes it difficult to apply one unified security policy across all devices. Third, IoT devices are often deployed in unattended or semi trusted environments. In such settings, weak authentication, unsafe default configurations, delayed updates, and poor access control can be exploited.

Because of these conditions, IoT networks face many kinds of attacks. Common examples include denial of service attacks that aim to overload devices or networks, probing activities that scan services and ports to find weaknesses, and remote to local attacks that attempt to gain access by exploiting exposed services or weak credentials [4]. In addition, attackers may target privacy by collecting sensitive data from insecure devices, or they may use compromised devices to form botnets and attack other systems. Therefore, IoT security requires protection measures that can detect threats early and respond to them without placing heavy burden on constrained devices [5],[7].

B. Intrusion detection and anomaly detection

Intrusion detection systems are widely used to monitor traffic and identify suspicious behavior. In general, intrusion detection methods can be grouped into signature based detection and anomaly based detection. Signature based detection uses

known patterns of attacks, so it can work well when attackers repeat known behaviors. However, it often becomes less effective when attackers change their methods or create new variants. In contrast, anomaly detection focuses on identifying deviations from normal behavior. This approach can be helpful when new attack behaviors appear, but it can also produce false alarms if normal traffic changes frequently. Anomaly detection methods are often classified into supervised and unsupervised learning approaches [8]. In supervised learning, models are trained using labeled data, and then they classify new traffic records based on what they learned during training [8].

This approach is useful when labeled datasets are available and when the goal is to distinguish normal traffic from attack traffic, or to classify different attack categories. In unsupervised learning, there are no labels, so the model attempts to learn what normal behavior looks like and then flags unusual behavior as possible attacks [9].Recent work also applies unsupervised deep learning for early network traffic anomaly detection [10]. This approach reduces the need for manual labeling, yet it can be more sensitive to changes in normal traffic and may raise more false alarms. For IoT networks, the choice between supervised and unsupervised methods often depends on the available data and the level of variation in device behavior.

C. Machine learning based intrusion detection for IoT

Machine learning has become an important direction for IoT intrusion detection because it can learn patterns from data and can support automatic classification beyond fixed rules. Researchers have studied many approaches, including classical machine learning algorithms, feature selection methods, and multi stage detection frameworks. In IoT settings, model selection is not only about accuracy. It is also about computational cost, response time, and the ability to remain stable when traffic patterns change. Ammarah Irum et al. reviewed detection and prevention methods for denial of service threats in IoT systems and highlighted the growing use of learning based solutions for detecting large scale attacks and changing traffic behavior [22]. Meidan et al. studied unauthorized device detection and showed that supervised learning can identify IoT device types using traffic features, including results with Random Forest models trained on traffic collected from multiple device categories [11].

In another study, Makkar et al. proposed a machine learning framework for spam detection in IoT environments and compared multiple classifiers using performance measures, with validation on smart home datasets such as REFIT[12],[4]. These studies show that machine learning supports a wide range of IoT security tasks, including intrusion detection, device identification, and message filtering. Some work has focused on specific IoT application areas where reliability is especially important. Liu and Xiao et al. proposed a trust joint light probe defense mechanism for industrial IoT and studied methods to detect malicious nodes under intermittent attack behavior while reducing error rates [13]. Ukil et al. explored anomaly detection in IoT healthcare analytics, where abnormal behavior detection is critical for system safety and service reliability [14]. These application focused studies confirm that intrusion detection in IoT must be accurate and also practical for real deployments. A major challenge in intrusion detection research is the detection of minority and low frequency attacks. These attacks appear less often in training data, so models may learn them poorly and may misclassify them as normal. Pajouh et al. proposed a two layer approach that combines dimensionality reduction with tier based

classification to improve the detection of low frequency categories such as user to root and remote to local attacks, using datasets such as NSL KDD and models such as Naive Bayes and k nearest neighbor [15]. Horng et al. combined hierarchical clustering with support vector machine classification and reported improved detection performance, especially for denial of service and probe attacks on the KDD Cup 1999 dataset [16]. Mukherjee et al. introduced a layered multi classifier method designed to improve minority attack detection and reported improved precision together with reduced false positive rates [17].

These studies show that attack imbalance, feature representation, and classifier design have a strong influence on intrusion detection performance. In addition to proposing models, broader reviews have discussed wider IoT security challenges that affect intrusion detection performance. Xiao and Liang et al. reviewed machine learning based IoT security and emphasized issues such as authentication, access control, malware detection, and secure offloading [18], [19] which shows that intrusion detection should be considered as one part of a complete security framework [4]. This body of work supports the view that learning based intrusion detection can provide advantages over static rule based systems, especially when IoT networks are diverse and attacker behavior changes over time.

D. Motivation for ensemble learning and comparative evaluation

Although individual machine learning models can achieve strong results, a single classifier may still fail under certain traffic conditions. For example, some models may overfit training data, some may favor majority classes, and some may lose accuracy when the traffic distribution changes. At the same time, different algorithms often perform well in different situations. Some classifiers capture nonlinear behavior effectively, while others provide faster prediction or clearer decision rules. For this reason, ensemble learning has become a popular strategy in intrusion detection research because it combines multiple classifiers and can reduce the impact of errors made by any single model. Based on this motivation, this work focuses on a comparative evaluation of widely used machine learning classifiers together with an ensemble based intrusion detection approach for IoT networks. The aim is to provide a clear baseline that explains how classical models behave under consistent settings and how a voting ensemble can improve reliability by combining complementary model decisions.

At the same time, the reviewed studies highlight continuing challenges for practical deployment, such as handling rare attacks, reducing false alarms, and maintaining performance when network behavior changes. These challenges support the need for careful evaluation and robust detection strategies that are effective and also feasible for IoT environments. In the Internet of Things (IoT), millions of devices are connected through wired or wireless media that have the ability to transfer data and information. IoT has spread rapidly within a few decades and is considered one of the largest networks in the world, as it enables human-to-human, human-to-things, and things-to-things communication by providing a unique identity to each device [1]. IoT is defined as a network of interconnected computing devices capable of communicating with each other through the Internet. These devices include everyday consumer devices such as smartphones and wearables, smart home devices such as smart meters, and smart industrial machines. These intelligent devices are capable of collecting, exchanging, and analyzing data to produce informed behavior accordingly. Global IoT investment is projected to exceed one trillion U.S. dollars. Vermesan et al. described IoT as an association between the real and virtual worlds,

where the virtual world interacts with the physical environment through sensors and actuators. IoT devices have become an essential part of Information and Communication Technology (ICT) infrastructure and are widely used in daily life activities. However, as millions of IoT devices are interconnected, the probability of cyber-attacks increases significantly. IoT systems are susceptible to network-level, physical, and application-layer attacks, as well as privacy leakage involving objects, services, and networks. Reliable transmission and intelligent processing within IoT networks are required to ensure secure operation. Despite the advantages offered by wireless communication, numerous vulnerabilities have been introduced into IoT environments. These vulnerabilities allow attackers to exploit personal information and launch cyber-attacks such as phishing, denial-of-service (DoS), probing, and remote-to-local (R2L) attacks [20].

Intrusion detection models employ different approaches to uncover anomalous or abnormal activities. Anomaly detection techniques are generally categorized into supervised and unsupervised learning methods [9]. In supervised anomaly detection, models are trained using labeled datasets [8], whereas unsupervised techniques rely on assumptions about normal behavior and do not require labeled training data [21]. Numerous studies have explored the application of machine learning techniques for intrusion detection in IoT systems. Ammarah Irum et al. [22] reviewed and compared various DDoS detection and prevention techniques in IoT environments to identify optimal solutions. Their analysis highlighted the effectiveness of intelligent learning-based approaches and examined whether specific techniques support post-attack analysis. Meidan et al. [11] demonstrated that unauthorized IoT devices can be accurately detected using supervised machine learning techniques. They applied Random Forest classifiers to extract network traffic features and evaluated their approach using traffic data collected from multiple IoT devices representing different device categories.

Makkar et al. [12] proposed a machine learning framework for spam detection in IoT systems. Their model evaluated multiple classifiers using various performance metrics to compute a spam score that reflects device authenticity. The REFIT smart home dataset was used for validation. Liu and Xiao et al. [13] proposed a Trust Joint Light Probe- Based Defense (TLPD) mechanism to improve security in industrial IoT systems against ON–OFF attacks. The proposed mechanism enhanced malicious node detection accuracy while reducing error rates. Ukil et al. [14] investigated anomaly detection in IoT- based healthcare analytics, emphasizing the importance of detecting abnormal behavior in medical monitoring systems using smartphone-based solutions.

Pajouh et al. [15] introduced a two-layer dimension reduction and two-tier classification framework to detect low-frequency attacks such as user-to-root (U2R) and remote-to-local (R2L) attacks. Their experiments were conducted using the NSL-KDD dataset with Naive Bayes and K-Nearest Neighbor classifiers. Horng et al. [16] proposed a hierarchical clustering and SVM-based intrusion detection system. Evaluation using the KDD Cup 1999 dataset demonstrated improved detection performance for DoS and Probe attacks. Clustering has also been used to model attacker activities for intrusion detection [23]. Mukherjee et al. [17] proposed a layered multi- classifier approach to enhance minority attack detection in intrusion detection systems. Their approach achieved improved precision and reduced false positive rates. Xiao and Liang et al. [4] reviewed machine learning-based IoT security techniques and discussed key challenges related to authentication, access control,

malware detection, and secure offloading. Overall, existing studies confirm that machine learning- based intrusion detection systems outperform traditional security mechanisms in IoT environments. However, individual classifiers often suffer from limitations such as bias, overfitting, and poor generalization. These limitations motivate the use of ensemble learning approaches, which combine multiple classifiers to improve robustness and detection accuracy. This work builds upon existing literature by presenting a comprehensive comparative analysis of multiple machine learning models and an ensemble-based intrusion detection approach for IoT networks.

PROPOSED METHODOLOGY

This section explains the proposed machine learning based intrusion detection system for Internet of Things networks. The goal is to detect malicious behavior in network traffic with high accuracy, while still keeping the overall process efficient and suitable for large volumes of data. The methodology follows supervised learning, where models learn from labeled traffic records and then classify unseen records as normal or malicious. The workflow is organized into sequential stages that include data acquisition, preprocessing, feature handling, model training, evaluation, and final comparison of results.

A. Overall System Architecture

The intrusion detection system is designed as a modular pipeline that converts raw network traffic into a final security decision. It starts with collecting or importing traffic records, then prepares the data for learning, trains multiple classifiers, and finally produces predictions for detection. Related intelligent architectures combining event processing and machine learning for IoT attack detection have also been reported [26]. This modular structure makes the system easy to extend. For example, additional classifiers can be included later, or new features can be

Table 1.
Summary Of Related Work on Machine Learning-Based Iot Intrusion Detection

author	ml technique	performance	target	dataset	evaluation measures
irum et al. [22]	detection & prevention	optimal ddos handling	ddos attacks	survey	Survey
meidan et al. [11]	supervised ml	accurate iot identification	unauthorized devices	local iot traffic	Accuracy
makkar et al. [12]	ml framework	spam score evaluation	spam detection	refit	accuracy, precision, recall
diro et al. [24]	deep learning	improved classification	multiple attacks	nsl-kdd	Accuracy
hasan et al. [25]	ml models	performance comparison	iot anomalies	ds2os	accuracy, precision, recall, f1
liu et al. [13]	tlpd	reduced error rate	network attacks	synthetic	Simulation
pajouh et al. [15]	two-layer ml	low-frequency detection	u2r, r2l	nsl-kdd	detection rate
horng et al. [16]	svm-based ids	improved dos detection	dos, probe	kdd 1999	Accuracy
mukherjee et al. [17]	multi-classifier	minority attack detection	mixed attacks	nsl-kdd	precision, recall
xiao et al. [4]	ml-based iot security	attack modeling	cyber attacks	multiple	Accuracy

tested, without changing the complete pipeline. In practical terms, the architecture can be explained in four connected layers. First, the data layer contains the traffic records collected from IoT environments under normal conditions and attack conditions. Second, the preprocessing layer improves data quality and converts it into a learning ready form by cleaning duplicates, handling missing values, normalizing numeric attributes, and encoding categorical attributes. Third, the learning layer trains different machine learning models on the same prepared data so that their strengths and weaknesses can be compared fairly. Fourth, the decision layer generates predictions and reports performance using standard evaluation metrics so that the

best model can be identified for intrusion detection. Fig. 1 illustrates the overall architecture of the proposed IDS pipeline.

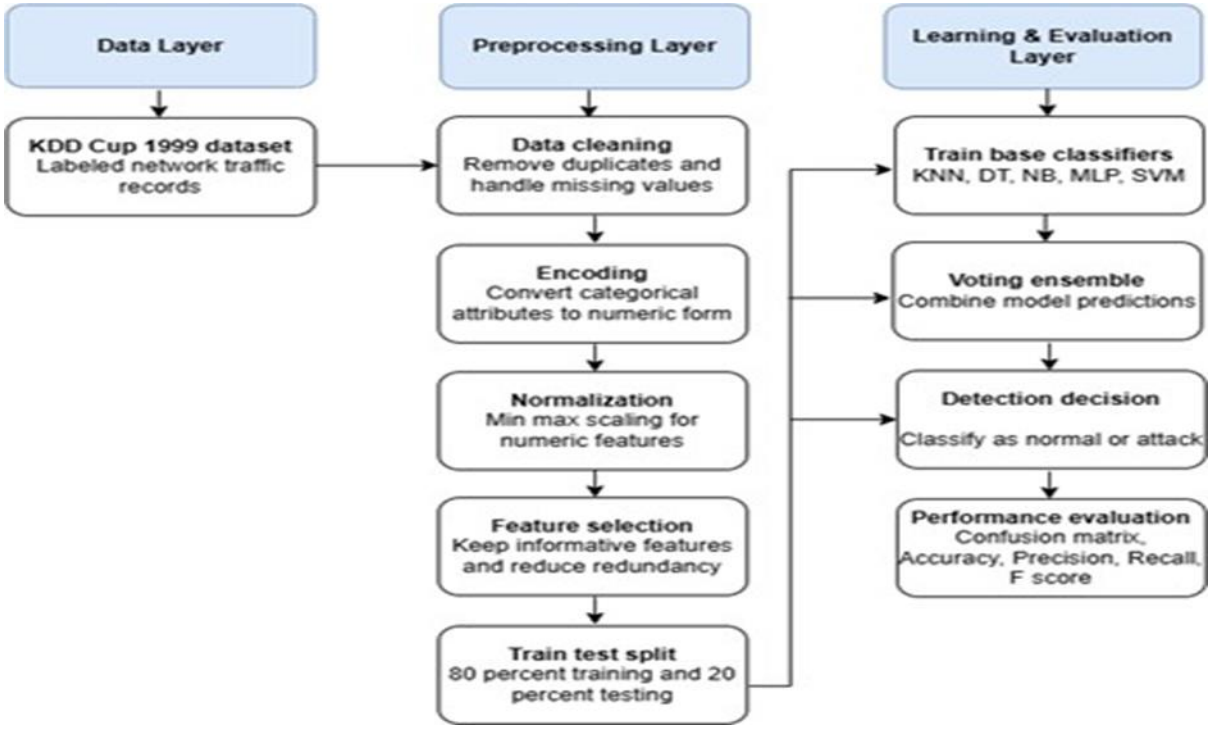

Figure1.
Overall architecture of the proposed machine learning based IoT intrusion detection system

B. Dataset Description

To evaluate the proposed intrusion detection system, a benchmark dataset is used that contains labeled network traffic records captured under normal operation and during attack events. Each record represents a network connection described by protocol based, statistical, and behavior related features, and each record includes a class label indicating whether the traffic is normal or malicious. Since the data are labeled, supervised learning is suitable for this problem. The dataset includes attack behaviors that are commonly studied in intrusion detection and are also relevant to IoT style deployments, such as denial of service attempts, probing activities, and other abnormal patterns.

Table II summarizes the main characteristics of the dataset used in this study, including the total number of records, the number of features, and the distribution of normal and attack traffic. The attack labels are grouped into four major categories. Denial of service attacks attempt to disrupt services by overwhelming network or system resources. Probe attacks focus on scanning and reconnaissance to identify potential vulnerabilities. Remote to local attacks aim to obtain local access from a remote machine without authorization. User to root attacks attempt to escalate privileges from a normal user account to administrator level control. In the experiments, two evaluation settings are considered to reflect practical intrusion detection requirements. In the first setting, the task is treated as a multi class classification problem where all available attack classes are retained and each attack type is learned as a separate class. In the second setting, very rare classes are removed by excluding attack types with fewer than one thousand samples. This reduces extreme class imbalance and supports more stable learning and testing, leaving eleven classes out of thirty eight for evaluation.

Table 2.
Dataset Statistics for The Kdd Cup 1999 Traffic Records Used For Intrusion Detection

Attribute	Description
Total Records	311029 records
Number of Features	41 features per connection record
Normal Traffic	60593 records
Attack Traffic	250436 records
Attack Categories	Four categories DOS U2R R2L PROBE

C. Data Preprocessing

Raw traffic data often includes duplicate rows, missing fields, inconsistent values, and features that use different numeric ranges. These issues can reduce model accuracy and also make training unstable. For this reason, preprocessing is applied before any model training. The preprocessing stage includes the following steps.

- **Data Cleaning:** Duplicate records are removed, and missing or inconsistent values are handled so that the dataset reflects valid traffic behavior rather than repeated or corrupted entries.

- **Encoding:** Categorical attributes are converted into numeric form using label encoding so that they can be processed by machine learning algorithms. Examples of such attributes can include protocol type, service type, and status flags in network flow records.

- **Normalization:** Numeric attributes are scaled using min max normalization. This step is important because features such as byte counts and connection durations may have large ranges, while other features may be much smaller. Scaling helps prevent large range features from dominating learning, especially for distance based methods.

- **Dataset Splitting:** Dataset Splitting: The dataset is divided into training and testing sets using an 80 to 20 ratio. The training set is used to learn model parameters, while the testing set is used only for final evaluation to estimate real performance on unseen traffic.

Together, these steps improve learning stability and support fair comparison across models by ensuring all algorithms learn from the same prepared data.

D. Feature Engineering and Selection

After preprocessing, the next step is to prepare the feature set so that it represents network behavior clearly and supports stable learning. Feature handling is important because intrusion detection performance depends on how well the input attributes describe traffic behavior. In this stage, the analysis focuses on features that reflect packet flow patterns, connection behavior, and time related trends, because these characteristics often separate normal activity from attacks. At the same time, some attributes may add little information or may repeat what other attributes already describe. Keeping too many weak or repeated features can increase computation and may also reduce generalization, which matters in IoT environments where resources are limited. Therefore, redundant or weak features are removed to reduce unnecessary inputs and keep the model efficient. Feature selection is carried out using simple, practical checks.

First, constant features are removed because they do not help distinguish between normal and malicious records. Next, highly correlated features are removed to avoid repetition and to reduce redundancy. Finally, the remaining features are kept based

on their stronger relationship with the attack labels, so the learning algorithms focus on the attributes that contribute most to detection. This process improves generalization, reduces training time, and can also reduce overfitting for some classifiers.

E. Machine Learning Model Implementation

Multiple supervised learning classifiers are implemented to study their effectiveness for intrusion detection and to understand how different learning principles behave on the same traffic data.

1) K-Nearest Neighbor (KNN): KNN assigns a label to a test record by comparing it with the most similar training records and using majority voting among the nearest neighbors. Its performance depends on the distance measure and the choice of the neighbor count. KNN can perform well when classes are well separated, but it can become computationally heavy when the dataset is large because it needs many comparisons during prediction [27].

2) Decision Tree (DT): Decision trees classify traffic by learning a hierarchy of rules. Each split is chosen to separate classes using feature based conditions. Decision trees are easy to interpret because the rules can be inspected, but they can overfit if the tree grows too deep. Controlling depth and split criteria helps maintain better generalization.

3) Na¨ıve Bayes (NB): Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that features contribute independently given the class label. Even though this assumption may not always match real traffic, the method is fast and often performs well when feature distributions are informative. It is also suitable when computational cost must remain low.

4) Multilayer Perceptron (MLP): MLP is a feedforward neural network that learns nonlinear patterns through hidden layers. It adjusts weights using backpropagation to reduce classification error. Because intrusion patterns can be nonlinear and complex, MLP can capture relationships that simpler models may miss. However, MLP needs careful tuning, such as selecting the number of hidden units and controlling training epochs, to avoid overfitting.

5) Support Vector Machine (SVM): SVM separates classes by finding a boundary that maximizes the margin between them. It performs well in high dimensional spaces and can provide strong generalization. Its performance depends on the choice of kernel and regularization settings. Feature scaling is important for SVM so that the optimization is not biased toward large range features.

6) Ensemble Model: The ensemble combines multiple trained classifiers and produces a final prediction using a voting strategy. In this study, a soft voting classifier is used with model averaging, where each base model contributes equally. The final label is selected by choosing the class with the highest sum of predicted probabilities across models. This approach can improve robustness, although it may not always improve results if one or more base models are overfit.

F. Tools and Implementation Environment

All experiments are implemented in Python using a standard machine learning workflow. Anaconda is used to manage the programming environment and required libraries, and Jupyter Notebook is used for interactive development, testing, and result

inspection. Dataset preparation tasks such as cleaning transforming, and organizing records are performed using pandas. Model construction, training, and evaluation are carried out using scikit learn, including utilities for cross validation, hyperparameter tuning, and performance measurement.

G. Model Training and Validation

Each classifier is trained using the same preprocessed training dataset to keep the comparison fair. The dataset is split into training and testing subsets using an 80 percent to 20 percent ratio, and the testing subset is reserved for final evaluation only. Hyperparameters for each model are selected empirically using the training data to obtain strong performance while avoiding settings that memorize the training set. Cross validation is applied on the training subset to reduce overfitting and to confirm that the learned models remain stable when trained on different subsets of the same data. Since attack classes are imbalanced, validation is performed in a way that keeps class proportions similar across folds so that each fold contains a meaningful number of attack samples. This prevents misleading validation outcomes that can happen when some folds contain very few samples from rare classes. If both evaluation methods are reported, the same training and validation procedure is repeated for the full label setting and for the filtered label setting, so performance differences reflect the label setting and not changes in the training process.

H. Evaluation Metrics

Performance is assessed using standard classification metrics derived from the confusion matrix. The confusion matrix summarizes prediction outcomes in terms of true positives, false positives, true negatives, and false negatives. These values show how many attack records are correctly detected, how many normal records are wrongly flagged as attacks, how many normal records are correctly recognized, and how many attacks are missed and predicted as normal. The study reports four widely used metrics: accuracy, precision, recall, and F score. Accuracy measures the overall proportion of correctly classified records. Precision measures the proportion of predicted attacks that are true attacks, which reflects the false alarm level. Recall measures the proportion of true attacks that are correctly detected, which reflects the missed detection level. F score combines precision and recall using their harmonic mean, providing a balanced view when both false alarms and missed detections matter. Table III summarizes the evaluation metrics used for performance assessment.

Table 3.
EVALUATION METRICS USED FOR PERFORMANCE ASSESSMENT

Metric	Description
Accuracy	Overall correctness of classification
Precision	Proportion of predicted attacks that are true attacks
Recall	Proportion of true attacks correctly detected
F-score	Harmonic mean of precision and recall

Because intrusion detection datasets can be imbalanced, accuracy alone may be misleading when a model performs well on dominant classes but fails on minority attack types. For this reason, precision, recall, and F score are reported alongside accuracy, and confusion matrix analysis is used to provide a clearer view of classifier behavior.

Performance Comparison

A comprehensive performance comparison is conducted to assess how different supervised learning models behave on the same prepared traffic data. The

comparison includes five single classifiers, namely K nearest neighbor, decision tree, naive Bayes, multilayer perceptron, and support vector machine, together with the proposed voting based ensemble. Results are reported under two evaluation settings. In Method 1, all available classes are retained for multi class classification. In Method 2, rare classes are removed by excluding classes with fewer than one thousand samples, which reduces severe class imbalance and supports a more stable comparison.

1) A. *Overall comparison using evaluation metrics:* The overall performance of all models is compared using accuracy, precision, recall, and F score. Using multiple metrics is important because accuracy alone can be misleading in intrusion detection, especially when class distributions are imbalanced. Precision reflects the false alarm tendency by measuring how many predicted attacks are truly attacks, while recall reflects the missed detection tendency by measuring how many true attacks are successfully captured. The F score summarizes the balance between precision and recall and is useful when both missed attacks and false alarms matter.

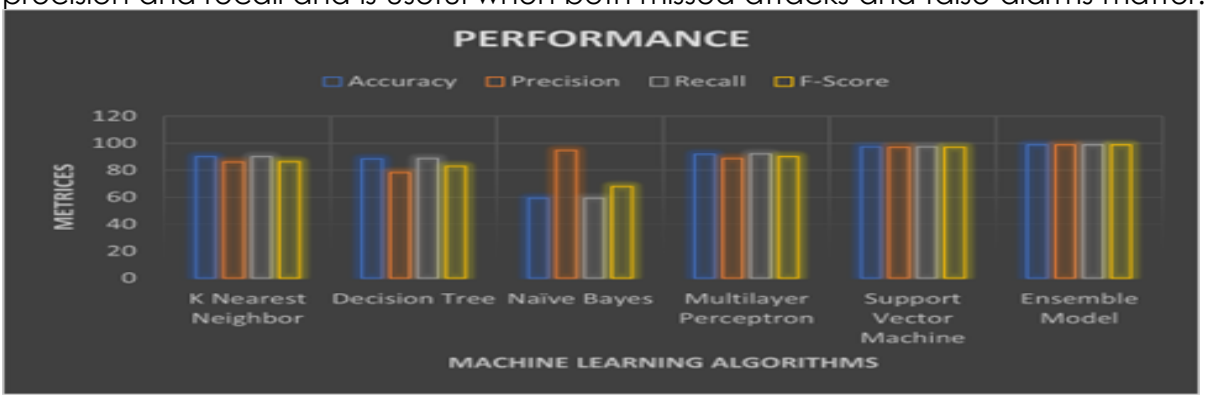

Figure 2.
Performance Comparison of ML algorithms under Method 1

To present the comparison clearly, the results are summarized using group bar charts for the two evaluation settings. Figure 2 shows the performance of K nearest neighbor, decision tree, naive Bayes, multilayer perceptron, support vector machine, and the ensemble model under Method 1, where all classes are retained for multi class evaluation. Figure 3 shows the corresponding performance under Method 2, where rare classes are removed to reduce severe imbalance and improve training stability. Together, these figures provide a compact view of how model performance changes across evaluation settings and highlight the consistent advantage of the voting based ensemble over single classifiers.

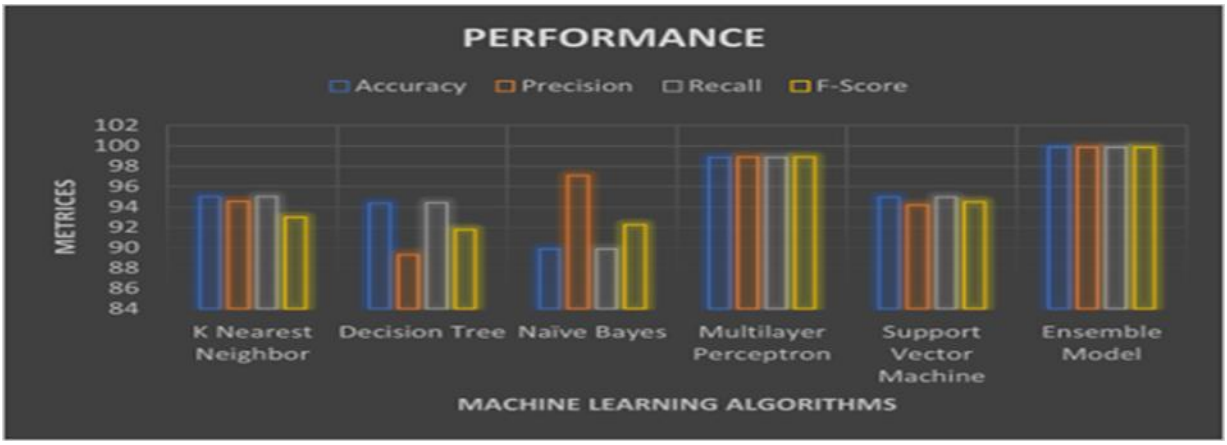

Figure 3.
Performance Comparison of ML algorithms under Method 2

Overall, the ensemble model provides the strongest performance because it aggregates predictions from multiple learners and reduces the impact of errors produced by any single classifier. Among individual models, performance differs because each algorithm learns decision boundaries in a different way and responds differently to class imbalance and class overlap. Method 2 typically produces higher and more stable scores because extremely rare classes are removed, which reduces unstable learning caused by very small class sizes.

2) Confusion matrix analysis: To complement the metric level comparison, confusion matrices are used to examine class wise prediction behavior and to identify common misclassification patterns. Unlike summary metrics, a confusion matrix shows how predictions are distributed across classes and whether errors are concentrated between closely related attack types. This analysis is important for intrusion detection because misclassifying an attack record as normal traffic is more critical than confusing one attack type with another, since it represents a missed detection.

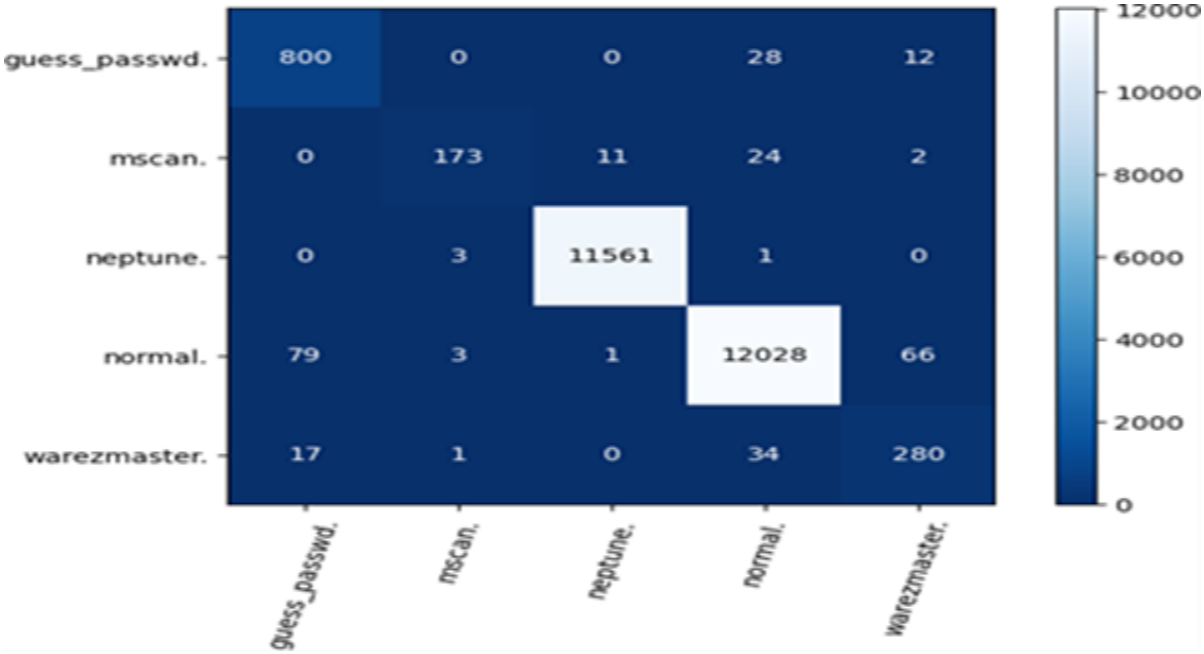

Figure 4.
Confusion matrix of the multilayer perceptron showing class wise detection behavior.

Figures 4 and 5 present confusion matrices for the multi-layer perceptron and the voting based ensemble, respectively. The multilayer perceptron is selected as a strong single classifier baseline, while the ensemble represents the proposed combined approach. In Figure 4, most samples fall along the diagonal, indicating that the multilayer perceptron correctly classifies a large portion of the records, although some confusion remains for specific classes. In Figure 5, the ensemble produces a clearer diagonal structure and fewer off diagonal errors, which indicates improved consistency across classes and reduced misclassification compared with relying on a single model. These results support the conclusion that combining complementary learners through voting improves class wise reliability and strengthens intrusion detection performance. *Summary of Findings:* The comparative evaluation indicates that the voting-based ensemble provides the most reliable intrusion detection performance across the considered evaluation settings. The overall metric results show that combining multiple classifiers leads to consistently strong accuracy, precision, recall, and F score, which supports the use of ensemble learning for traffic classification tasks where both missed detections and false alarms matter. The results

also show that individual classifiers can perform well under certain conditions, yet their limitations become clearer when the dataset contains class imbalance or when different attack types share similar traffic patterns. In these cases, a single model may produce higher false alarms or miss particular attack instances by predicting them as normal traffic. The ensemble reduces this risk because it aggregates decisions from models that learn different patterns from the same data. As a result, errors made by one classifier are often corrected by others, leading to more stable class wise behavior.

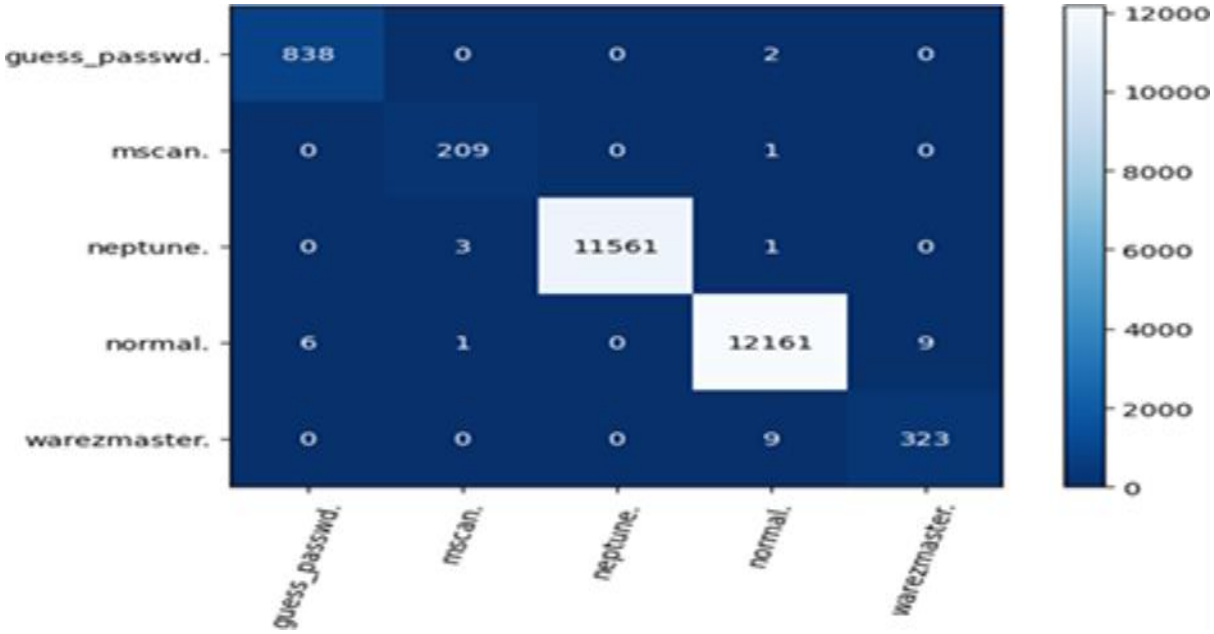

Figure 5.
Confusion matrix of the voting based ensemble showing improved class wise consistency.

3) Across both evaluation methods, the ensemble maintains the most consistent diagonal dominance in the confusion matrix analysis and the strongest overall performance trends in the metric comparison graphs. This consistency suggests that the proposed ensemble approach is a practical choice for IoT intrusion detection systems that must operate under heterogeneous traffic behavior and changing attack characteristics.

DISCUSSION

The results show that the voting based ensemble achieves the best overall performance and provides more consistent detection across classes than any single classifier. Under the full label evaluation, the ensemble reaches 98.87 percent accuracy, outperforming the best individual models. When rare classes are removed in the filtered evaluation, performance improves further and the ensemble reaches 99.87 percent accuracy, mainly because the learning task becomes more stable when extremely small classes are excluded. Differences among individual classifiers follow expected behavior. Naive Bayes performs weakest because its independence assumption often does not match real traffic patterns, while SVM and MLP perform strongly because they can learn better separating boundaries for complex data. Decision tree and KNN provide moderate performance, but they can be affected by overfitting and high dimensional distance effects. Confusion matrix analysis supports these findings by showing that the ensemble produces fewer off diagonal errors and a clearer diagonal structure, which indicates improved class wise reliability. This is

important for intrusion detection because predicting attacks as normal traffic represents missed detections. Overall, the ensemble approach is a practical choice for IoT intrusion detection when robust performance across diverse traffic behaviors is required.

CONCLUSION AND FUTURE WORK

This study investigated supervised machine learning approaches for intrusion detection in IoT oriented network traffic by evaluating five widely used classifiers and a voting based ensemble model. The experimental results show that the ensemble consistently provides the strongest performance, indicating that combining multiple learners produces more reliable detection than relying on a single classifier. The comparison also highlights that individual models differ in their strengths, yet their performance can be affected by class imbalance and similarity between attack patterns. Future work will extend this study in three directions. First, lightweight deep learning models can be explored to improve feature learning while keeping computational cost manageable. Second, the intrusion detection workflow can be adapted for real time operation and tested on edge or gateway devices to study latency, memory use, and energy requirements. Third, evaluation should be performed on additional and larger scale IoT security datasets and on more realistic traffic traces to confirm generalization and robustness under changing attack behavior and evolving network conditions[28] ,[29].

DECLARATIONS

Acknowledgement: We appreciate the generous support from all the contributor to the research and their different affiliations.
Funding: No funding body in the public, private, or nonprofit sectors provided a particular grant for this research.
Availability of data and material: In the approach, the data sources for the variables are stated.
Authors' contributions: Each author participated equally in the creation of this work.
Conflicts of Interest: The authors declare no conflict of interest.
Consent to Participate: Yes
Consent for publication and Ethical approval: Because this study does not include human or animal data, ethical approval is not required for publication. All authors have given their consent.

REFERENCES

A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," Future Generation Computer Systems, vol. 82, pp. 761–768, 2018.

A. Aggarwal and M. L. Das, "Rfid security in the context of internet of things," in *Proceedings of the 1st International Conference on Security of Internet of Things*, 2012, pp. 51–56.

A. Kumar and T. J. Lim, "Edima: Early detection of iot malware network activity using machine learning techniques," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019.

A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commu- nications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

A. Makkar *et al.*, "An efficient spam detection technique for iot devices using machine learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 903–912, 2021.

A. Ukil *et al.*, "Iot healthcare analytics: Anomaly detection and related algorithms," in *Proceedings of the IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 535–542.

ACM Computing Surveys, vol. 41, no. 3, p. 15, 2009.

E. M. Karanja, S. Masupe, and M. G. Jeffrey, "Analysis of internet of things malware using image texture features and machine learning techniques," *Internet of Things*, vol. 9, p. 100153, 2020.

F. Hussain *et al.*, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, 2020.

H. H. Pajouh *et al.*, "A two-layer dimension reduction and two-tier classification module-based intrusion detection system for iot traffic," *IEEE Transactions on Emerging Topics in Computing*, 2019.

H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved knn," International Journal of Computer Applications, vol. 173, no. 1, pp. 5–9, 2017.

Irum, A., Khan, M. A., Noor, A., & Shabir, B. (2020). DDoS detection and prevention in internet of things. . EasyChair, (2486), 1-7.

J. Rolda´n et al., "Integrating complex event processing and machine learning: An intelligent architecture for detecting iot security attacks," Expert Systems with Applications, vol. 149, p. 113251, 2020.

J. Theiler and D. M. Cai, "Resampling approach for anomaly detection in multispectral images," in *Proceedings of SPIE 5093, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX*, 2003, pp. 230–240.

L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "Iot security techniques based on machine learning: How do iot devices use ai to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41– 49, 2018.

M. Hasan, M. Z. Islam, M. M. A. Hashem, and A. S. M. Kayes, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019.

O. Vermesan and P. Friess, Eds., *Digitising the Industry: Internet of Things Connecting the Physical, Digital and Virtual Worlds*. River Publishers, 2016.

P. Sun *et al.*, "Modeling and clustering attacker activities in iot through machine learning techniques," *Information Sciences*, vol. 479, pp. 456– 471, 2019. pp. V5–484–V5–487.

R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," in 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018.

R.-H. Hwang *et al.*, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30 387– 30 399, 2020.

S. Dange and M. Chatterjee, "Iot botnet: The largest threat to the iot network," in *Data Communication and Networks*. Singapore: Springer, 2020, pp. 137–157.

S. J. Horng *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, no. 1, pp. 306–313, 2011.

UCI KDD Archive, "Kdd cup 1999 data," [Online], available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Accessed: 2024.

V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey,"

Y. Liu *et al.*, "Defending against advanced persistent threats with on-off attacks using trust joint matrix," *Future Generation Computer Systems*, vol. 84, pp. 1–13, 2018.

Y. Meidan *et al.*, "Detection of unauthorized iot devices using machine learning techniques," 2017, arXiv:1709.04647.