Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 3, Issue 3, 2025

MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM FOR IOT NETWORKS UTILIZING LIGHTWEIGHT PROTOCOLS MQTT AND COAP

Ayesha Eman[1], Shakeel Ahmed Laghari[2], Nadeem Ahmed*[3], Fayaz Hassan[4], Bahzad Qadir Memon[5]

[1,2, *3,4,5]*Deparment of Telecommunication Engineering, Mehran university of Engineering and Technology, Jamshoro, 71000 PK*

DOI: https://doi.org/10.5281/zenodo.15007653

Keywords
Internet of Things, DoS (Denial of Service), Intrusion Detection System (IDS), Anomaly Based IDS, Performance Evaluation.

Article History
Received on 01 January 2025
Accepted on 01 February 2025
Published on 11 March 2025

Copyright @Author
Corresponding Author: *

Abstract
With the massive growth of IoT devices, the attack surfaces have intensified, making cybersecurity a critical component for protecting organizational boundaries. Advancements in cybersecurity have increasingly integrated machine learning techniques to strengthen the identification and prevention of sophisticated threats. Intrusion Detection Systems (IDS) play a pivotal role in network security, IDS is employed in networks to raise critical flags during network management, particularly for malicious traffic identification and the detection of attacks, which remain a significant challenge. Traditional models often fall short due to their dependence on static rules and signature-based methods. This paper presents a cutting-edge IDS framework that leverages a combination of machine learning algorithms like Random Forest, Gradient Boosting, Naive Bayes, XGBoost, and Decision Trees to enhance both anomaly detection and threat classification. Tested on real network traffic, the system demonstrated an impressive accuracy rate of 99.46%. Furthermore, most existing research predominantly relies on datasets like NSL-KDD, KDD-CUP99, or CICIDS, which do not accurately reflect the traffic and attack patterns associated with lightweight communication protocols such as MQTT and CoAP, commonly used in IoT networks. To address this limitation, we utilized a more realistic dataset, the NET-SEC (Network Security) dataset, designed specifically for smaller network environments. This IDS offers a highly efficient and scalable solution, effectively addressing the limitations of conventional systems while providing robust, adaptive protection against evolving security threats.

INTRODUCTION

The rapid expansion of the Internet of Things (IoT) has fundamentally transformed industries such as healthcare, smart cities, industrial automation, and consumer electronics. By 2024, the number of connected IoT devices is projected to surpass 50 billion, generating an unprecedented volume of data and significantly increasing the demand for real-time communication . However, this massive growth has also exposed IoT networks to an escalating range of cyber threats. Unlike traditional network devices, IoT devices often operate with severe limitations in terms of computational power, memory, and battery life, making them particularly vulnerable to cyberattacks. As the complexity and sophistication of threats like Distributed Denial of Service (DDoS) attacks, malware, and botnets evolve, traditional security measures, such as firewalls, have proven inadequate in securing IoT environments.

Their IDS serves as a defensive wall against these growing threats. These systems actively observe network activity to indicate unauthorized access or anomalous behavior. However, common/static signature IDS models have problems coping with the dynamic nature of IoT environments. These systems cannot detect many newly emerging zero-day attacks, as they primarily rely on static predefined signatures to detect such attacks, which do not capture the new evolving threats. Further, there are diverse attack vectors with limited resources available in IoT devices, which make it difficult to deploy computationally intensive IDS solutions in real IoT ecosystems. Therefore, there is an ever-increasing demand for low-power adaptive IDS systems that keep high detection ratios without exhausting the limited resources of IoT. These improvements may offer the possibility of developing contemporary intelligent intrusion detection systems by using machine learning. Such ML-based IDS models can learn from past samples of network traffic for assessing known and unknown attack patterns. However, despite this potential, several hindrances face such dweller shape up models of ML-based IDS. Many models proposed in the literature are too complex to be deployed on resource-constrained IoT devices, making them impractical for real-world use. Moreover, the datasets typically used to train these models such as NSL-KDD, DARPA, and CICIDS fail to capture the unique traffic patterns and protocol-specific attack vectors found in IoT environments. As a result, these models often lack the robustness required for effective intrusion detection in modern IoT networks, which highlights a significant gap in current IDS research.To address these limitations, this paper makes the following key contributions:

- Development of the NET-SEC Dataset: We introduce the NET-SEC (Network Security) dataset, specifically tailored for IoT environments. While existing datasets like NSL-KDD and CICIDS have contributed to IDS research, they fail to accurately represent the traffic and attack patterns specific to lightweight communication protocols such as MQTT and CoAP, which are prevalent in IoT networks. NET-SEC addresses this gap by incorporating attacks that target these protocols, providing a more realistic benchmark for IoT security research.
- Comprehensive Evaluation of ML Classifiers: We evaluate several machines learning classifiers, including Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks, using the NET-SEC dataset. Our focus is not only on detection accuracy but also on computational efficiency, ensuring that the models are practical for deployment on resource-constrained IoT devices. This evaluation bridges the gap between academic research and practical deployment, highlighting models that can balance performance with efficiency.
- Real-World Validation: Beyond controlled experiments, we test the performance of our optimized IDS models in real-world network traffic scenarios, demonstrating their applicability in live IoT environments. This real-world validation ensures that our findings extend beyond academic benchmarks, providing actionable insights for practical implementation in IoT networks.

I. RELATED WORK

The Internet of Things (IoT) has transformed industries across the globe, bringing advancements in fields like healthcare, smart cities, and industrial automation. For example, in health care, the use of IoT devices such as pacemakers, insulin pumps, and wearable medical monitors has led to a complete rethinking of patient experience by real-time monitoring capabilities[2]. In smart homes and cities, IoT-enabled automation enhances efficiency and makes urban dwellings livable. Meanwhile, IIOT (Industrial Internet of Things) is involved in predictive maintenance, real-time monitoring, and even system automation. However, as the number of IoT devices increases, the risk involved in their deployment also increases. [3] . Many IoT devices operate with limited processing power, memory, and battery life, making them attractive targets for cyberattacks. Vulnerabilities in IoT devices have been leveraged in numerous high-visibility attacks such as botnets, malware infections, and Distributed Denial-Of-Service (DDoS) attacks [4] .Such breaches in security can have far-reaching consequences, especially in high-stakes scenarios like healthcare and industrial automation, where failure of a device or unauthorized access could entail massive financial losses or, in some extreme cases, loss of life. Mohamed et al [5] suggested that with the increasing interconnectivity of devices, the fast pace at which cyber threats are also evolving makes it almost imperative to have strong security mechanisms in place within IoT ecosystems.

A. EVALUATION AND ROLE OF INTRUSION DETECTION SYSYTEM IN IoT SECURITY

Intrusion Detection Systems have been monitoring networks and detecting malicious activities for quite some time. Initially, these systems used signature-based mechanisms to match incoming traffic to known attack patterns stored in a database. While signature-based methods are effective against known threats, they cannot detect new or unknown attacks. As IoT devices proliferated, traditional IDS methods became inadequate to protect the heterogeneous and decentralized networks. For this reason, researchers have shifted increasingly toward anomaly-based IDS, which monitor actual network behavior in detecting deviations from normal patterns. This is useful for the detection of new attack vectors, including some zero-day exploits. The growing proliferation of IoT communication protocols such as MQTT and CoAP requires further adaptation of IDS systems to address the traffic peculiarities posed by IoT. Studies performed recently, such as in the work by, Mohamed et al [5] and Khan et al [6] indicate these evolving protocols will require more sophisticated IDS capable of attacking detection and performance handling under resource-constrained IoT devices.

B. MACHINE LEARNING IN INTRUSION DETECTION SYSTEM DEVELOPMENT FOR IoT NETWORKS

The involvement of machine learning (ML) techniques for detection in IDS has effectively enhanced the performance of identifying both known and unknown attacks [7] .Traditional rule-based and signature-based IDS methods rely on sets of predefined attack patterns and, therefore, are too rigid to face the fast-evolving nature of IoT cybersecurity threats. In contrast, machine-learning-based IDS can be trained on large datasets of network traffic to learn patterns of normal and malicious behavior; thus, they can detect novel attack types by spotting deviations from learned patterns. Machine learning algorithms such as Random Forest, Support Vector Machine (SVM), and Decision Trees are doing well in the classification of IoT network traffic and in identifying known attacks [8] . Higher-end techniques, such as unsupervised learning algorithms, are being employed to detect anomalies in network traffic. Some examples are A/k-means and Autoencoders, which are widely used in the identification of unusual patterns that may characterize cyberattacks. Al-Hawawreh et al [9] have shown that autoencoders are, in fact, the most promising anomaly detection solution for IoT networks as they allow precise detection with minimal processing requirements on computationally constrained devices.

C. DATASETS FOR IoT IDS RESEARCH

The datasets used to train the models have a crucial effect on machine learning-IDS performance. Most of the commonly used datasets for intrusion detection were built first and foremost for traditional IT environments; little consideration was given to traffic patterns or attack vectors apparent in IoT networks. [10] . Thus, there is an evident need for specific datasets in consideration of the uniqueness of IoT environments.

Some datasets used in IoT IDS studies include:

- **NSL-KDD:** Despite solving the redundancy and imbalance problems of the original KDD'99 dataset, this dataset does not consider any modern attack types or lightweight IoT protocols which operate on current IoT networks[11].

- **CICIDS2017:** This dataset captures a number of modern attack types such as DDoS and botnet infections; yet it is not an IoT-compliant dataset and does not capture the lightweight communication protocols used by IoT devices[12].

- **UNSW-NB15:** UNSW-NB15 is another dataset with realistic network traffic but is mainly geared towards those not IoT specific settings.[13].

- **TON_IoT 2020**: More IoT-centric research than others, the TON_IoT 2020 dataset focuses on the industrial IOT contexts. However, it is geared toward large systems and does not represent the characteristics of smaller, resource-constrained devices generally found in an IOT setting. [14], [15] . Although these datasets are available, their shortcomings illustrate the necessity of creating custom datasets tailored to represent IoT traffic and attacks.

D. CONTRIBUTION OF THIS STUDY: CUSTOM IoT DATASET

To address the gaps in current datasets, this study introduces a custom dataset designed specifically for IoT networks. The creation of the datasets involved modeling realistic IoT environments and the insertion of actual IoT devices such as Raspberry Pi and Arduino. The dataset captures data in traffic from IoT-specific protocols such as MQTT and CoAP, used for IoT deployments. It describes a range of novel IoT attack scenarios including DDoS attacks, botnet infections, and MQTT injection attacks.

The dataset is generated using Wireshark for network analysis, while network emulation frameworks emulate attack scenarios. [16] . This guarantees data that are reasonably exhaustive and within the bounds of relevant security problems in IoT today. With such a wide-ranging diversity of attack types and communication protocols involved, this dataset serves to be a blessing on many occasions to researchers seeking to improve IDS in IoT networks.

E. KEY CHALLENGES IN MACHINE LEARNING BASED INTRUSION DTECETION SYSTEM

Despite advancements in machine learning and dataset development, several challenges remain in designing effective IDS for IoT environments:

I. Managing False Positives and False Negatives

Managing the balance between false positives and false negatives remains a significant challenge in IoT IDS. Although unsupervised models are effective for identifying unknown threats, they tend to produce a high volume of false positives. These models require further optimization to enhance detection accuracy and reduce the workload for system administrators[17].

II. Resource Constraints in IoT Devices

IoT devices, such as Raspberry Pi, are constrained by limited processing power and memory. Developing lightweight IDS models is crucial to operate efficiently within these limitations. Our custom dataset supports this by facilitating the evaluation of resource-efficient machine learning algorithms designed for IoT environments [3].

F.FUTURE DIRETIONS IN IoT SECURITY RESEARCH

Several areas of research hold promise for improving IoT security:

I. **Federated Learning:** Federated learning allows IoT devices to collaboratively train machine learning models without sharing raw data, improving scalability and privacy [18].

II. **Edge Computing:** By offloading complex computations to edge servers, edge computing can reduce the processing burden on IoT devices and enable real-time intrusion detection.

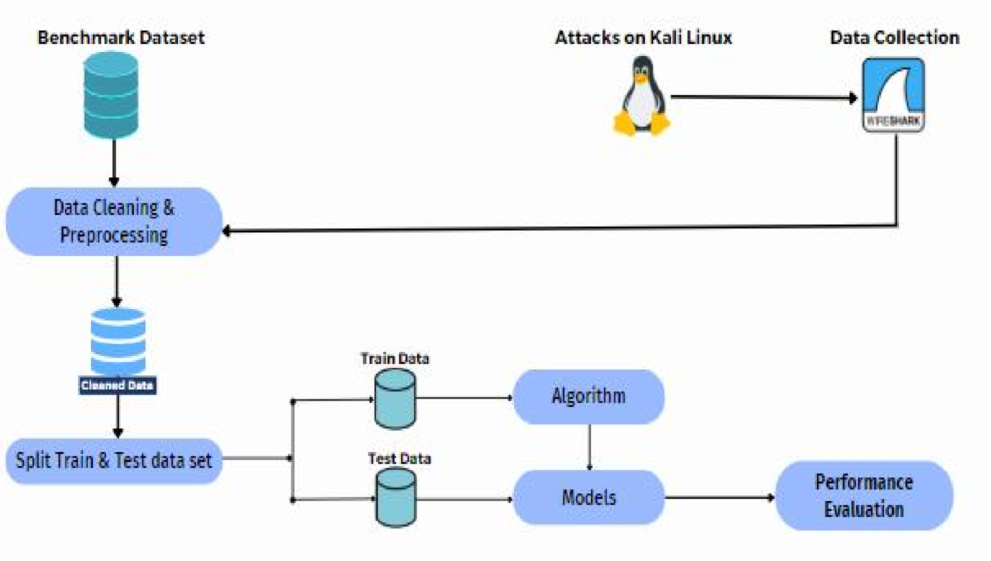
III. **Explainable AI (XAI):** Incorporating explainable AI techniques into IDS models can increase the transparency and interpretability of decision-making processes, which is critical for trust among administrators and end-users [19].

IV. **Dynamic Dataset Updates:** As IoT environments evolve and attack strategies become more sophisticated, it is essential to develop datasets that are dynamic and reflect changing traffic patterns and emerging attack types[20].

High-quality datasets while ensuring that the systems are designed to take care of the peculiarities of IoT environments. In the future, the focus of research is going to be directed toward handling false positives, resource limitations, and relevance in datasets, while at the same time exploring emerging techniques like federated learning and explainable AI that will be enhancing integrity and credibility for IoT IDS systems.

II. METHODOLOGY

FIGURE 1: Process flow diagram of the methodology. The diagram illustrates the sequential steps involved in the proposed approach, detailing data preprocessing, feature extraction, model training, and evaluation.



In this section, we detail the methodology employed for our study, encompassing the processes of data preprocessing, extraction, and analysis. Each phase was carefully designed to ensure the reliability, accuracy, and robustness of the results. The process flow diagram illustrating this methodology is presented in Error! Reference source not found.

A.SETUP AND CONFIGURATION

The experimental arrangements were made to mimic real-world network traffic while evaluating the performances of different IDS models. This section describes the system configuration used for simulation of network traffic and prosecution of potential threats.The system had the following components:

- Primary Machine, a performance server, modulated for running machine learning models and processing network traffic data.
- Secondary Machine for simulating network traffic, simulation runs, and monitoring IDS performance during attacks and normal operations.

For simulating the network traffic, the necessary condition for the field to happen took form as a controlled testing environment consisting of Wireshark, Tcpdump, and Tcpreplay for packet capturing and traffic generation describes instantaneously their use was such that behavior of each other was a draw for the investigation. IDS tools such as Snort were employed to monitor traffic in real-time and identify any potential cyber threats. The machine configuration is outlined in **TABLE** 1.

TABLE 1: CONFIGURATION OF SYSTEM COMPONENTS

Components	Details
Machine Specifications	12GB RAM, Windows 10 Pro
Programming Language	Python 3.6.1
Machine Learning Library	Scikit-learn 0.18
Network Simulation OS	Ubuntu 14.04 (64-bit)
Network Protocol Analyzer	Wireshark
Traffic Generation Tool	Tcpreplay
Network Security Monitoring	Snort
Packet Capturing Tool	Tcpdump
Dataset Preprocessing Tools	NfDump, NfSen

B. DATASETS

Five datasets were used to evaluate the performance of the IDS models: KDDCUP1999 [21] , MSKD[22] ,DARPA1999[23], CICIDS2017[24], and CICIOT [25] . These datasets were chosen for their relative completeness and ability to represent various network environments, types of attacks, and traffic patterns. A brief description of each dataset follows:

I. KDDCUP1999 Dataset

This dataset ranked among the most popular benchmarks for assessing IDS. It has 100,000 instances of network traffic with a balance of normal and abnormal occurrences. There are the following types of attacks in this dataset: denial of service (DoS), remote-to-local (R2L), user-to-root (U2R), and probing. KDDCUP1999 has been subjected to various criticisms for being obsolete; however, it remains one of the few paths for early-stage evaluation of IDS models.

II. NSLKDD dataset

The other aspect of this dataset is a refinement of the different categories of attacks in the KDDCUP1999 database, eliminating duplicates in its data and making it more appropriate to IDS applications. It is way better than KDDCUP1999 with a very much smaller chance of error, thus simulating network traffic with higher accuracy.

III. DARPA1999 Dataset

Under controlled exercises in cyber security, the DARPA1999 dataset simulates different attack types, both insider and outsider. It is a very famous dataset for IDS studies because it can emulate real attack patterns within the structured environment. In relation to DARPA1999, the work assesses the IDS models against historically as well as still relevant attack types, which would facilitate a thorough evaluation of the systems' performance. The inclusion of this dataset also helps bridge the gap between traditional security challenges and modern-day threats, making it an essential part of the evaluation framework.

IV. CICIDS2017 Dataset

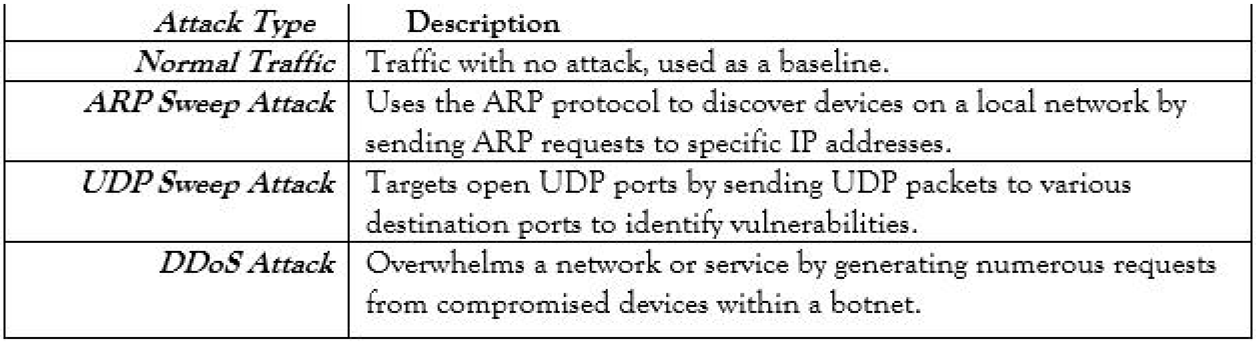
The CICIDS2017 dataset is the product of the Canadian Institute for Cybersecurity (CIC) and includes the evolution of modern enterprise networks. It also accommodates recent attacks such as Distributed Denial of Service (DDoS) attacks, ransomware, or botmasters. This dataset gives realistic traffic data across the networks, which would thereby be ideal in testing IDS models involved in the newest cyber threats [26] . It is thus realistic for surveying such kinds of enterprise network environments that the study may then try to assess their effectiveness based on the IDS models in tackling very complex and evolving attacks. The pivotal role held in this research will ensure that IDS solutions remain fresh and able to address challenges entrenched in today's security concerns.

V. CICIOT Dataset

As IoT networks become increasingly prevalent, the CICIOT dataset is critical for evaluating IDS models tailored to Internet of Things (IoT) environments. It includes data from various IoT devices under attack scenarios, such as Botnet attacks and Distributed IoT Attacks. This dataset ensures the IDS models are tested for security in the context of IoT networks.

Each dataset contributes uniquely to the study, collectively providing a diverse and comprehensive testing ground for IDS models across a range of network conditions and threat types. This approach ensures that the IDS models are rigorously evaluated for their adaptability,accuracy, and effectiveness in both legacy and modern network environments.

TABLE 2: SUMMARY OF TRAFIC TYPES AND ATTACK DESCRIPTIONS



Attack Type	Description
Normal Traffic	Traffic with no attack, used as a baseline.
ARP Sweep Attack	Uses the ARP protocol to discover devices on a local network by sending ARP requests to specific IP addresses.
UDP Sweep Attack	Targets open UDP ports by sending UDP packets to various destination ports to identify vulnerabilities.
DDoS Attack	Overwhelms a network or service by generating numerous requests from compromised devices within a botnet.

C. DATA EXTRACTION AND DATA FORMATION

The custom NET-SEC dataset was created by extracting and refining information from the Benchmark dataset (CICIoT). Both normal and malicious traffic were deliberately generated under realistic network conditions, using a Wireshark on Ubuntu machine with IDS tools monitoring and detecting any potential threats. Wireshark for captured and analyzed data sources, while IDS tools detect any suspicious activity as happened. Then thoroughly filtered and cleaned all this data in terms of accuracy and relevance. This process eventually resulted in proper documentation creation of NET-SEC dataset.it is a very good and reliable basis to judge and optimize IDS algorithm.

I NET-SEC Dataset Development

The NET-SEC dataset was developed specifically for this research by extracting data from the CICIOT dataset. The extracted data was categorized into normal traffic and anomalous traffic to create a balanced dataset suitable for training and testing machine learning-based IDS models. To ensure the dataset's accuracy, all irrelevant data was removed, and the remaining traffic instances were preprocessed to include only necessary features such as:

- Source IP Address
- Destination IP Address
- Protocol
- Payload Size
- Timestamp
- Traffic Direction (incoming or outgoing)

The NET-SEC dataset includes a variety of cyberattack patterns, such as ARP Sweep, UDP Sweep, and DDoS attacks, making it an essential resource for evaluating IDS models in dynamic network environments.

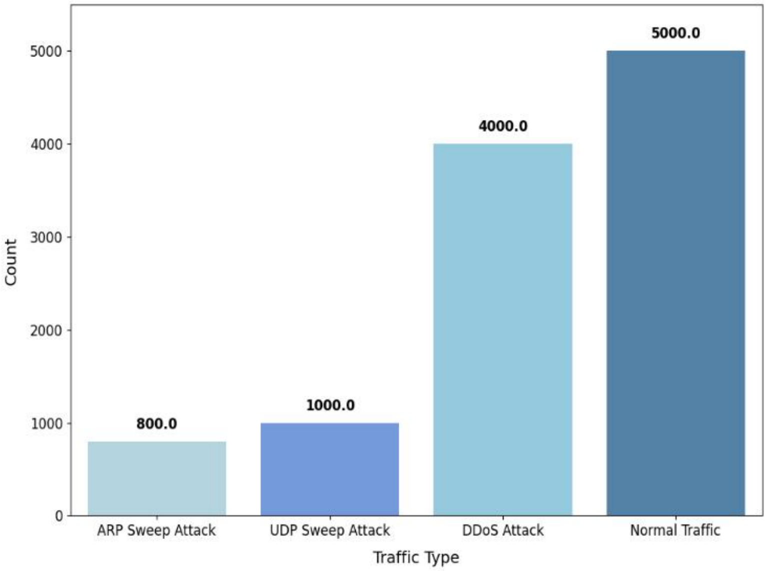
Datasets Development and Labeling

To develop an effective dataset for Intrusion Detection Systems, we have visited and collected network traffic data that shows normal activities as well as anomalous activities. Examples of threats include denial of service, DDoS attacks, port scanning, to mention a few. The core information recorded in the dataset includes the source and destination IP addresses, protocols, data payloads, and type of network activities. The next step in making the dataset usable in training for IDS models is to label each data entry as normal or anomalous. The labeling will help the IDS model to learn the normal network behavior from the deviation thereof or security threat within a network. Accurate labeling allows the model to learn to recognize when to act in response to multiple types of cyber-attacks.

a) ARP Sweep Attack: An ARP sweep attack uses the Address Resolution Protocol (ARP) to discover devices on a local network by sending ARP requests to specific IP addresses, identifying active devices. This technique is often used for network mapping and security testing.

b) **UDP Sweep Attack:** UDP sweep attack targets open User Datagram Protocol (UDP) ports, sending UDP packets to various destination ports to identify vulnerabilities. This method helps locate accessible ports, typically used by services such as DNS and DHCP.

FIGURE 2: The count plot for the three types of attacks, the visualization represents the frequency of occurrences for each attack type, highlighting their distribution within the dataset.



DDos Attack: A Distributed Denial of Service (DDoS) attack overwhelms a network, service, or website with a high volume of traffic. By generating numerous requests from compromised devices within a botnet, the attack aims to incapacitate the target and prevent legitimate access. In our simulation, 4,000 DDoS requests were used to test its disruptive potential. Legitimate access. In our simulation, 4,000 DDoS requests were used to test its disruptive potential.

Normal traffic, simulated with 5,000 instances, operated smoothly without any issues. The ARP Sweep and UDP Sweep Attacks effectively met their goals, showing high efficiency in detecting active devices and vulnerabilities. These observations illustrate the different effects of each attack type on network performance and the effectiveness of each method, as depicted in FIGURE 2. The dataset was labeled for each entry as normal or anomalous. Labeling helps the IDS model learn the normal network behavior and identify deviations indicating a security threat. Below are the labels used in this study:

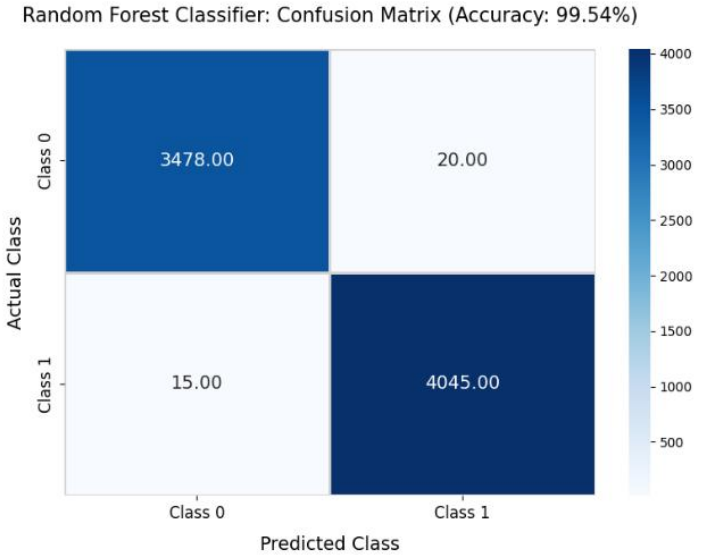
II Data Classification

Across the dataset, the entire parameter of machine-learning classifiers was utilized to analyze models that could accurately detect malicious traffic. The classifiers under this discourse are Decision Trees, Random Forest, Support Vector Machine, and Artificial Neural Network. Their choice was based on their well-established capabilities in the detection of intrusions and ability to handle different types of network traffic data. Each classifier was trained and evaluated as a systematic manner to detect the patterns associated with normal traffic and attack traffic by making good use of their various strong points in data analysis and predictions. Some of these classifiers were enhanced in their performance by feature selection performed via the Random Forest Classifier. This included feature selection that could preserve dimensionality and provide enhanced efficiency of the model in detecting attacks. The feature is that the least useful attributes help in classifying normal and malicious protection. The dataset was separated into two subsets: 70% set aside to train the models and 30% to test them. Through this approach, balanced validation of the model greatly helps the classifiers generalize unseen data well. The activated features for classification were Flow Duration, Packet Size, Protocol Type, and Flow Count, which gave the character of network traffic and differences to the classifiers in the detection of anomalies and threats with higher accuracy.

IV Classifier Training

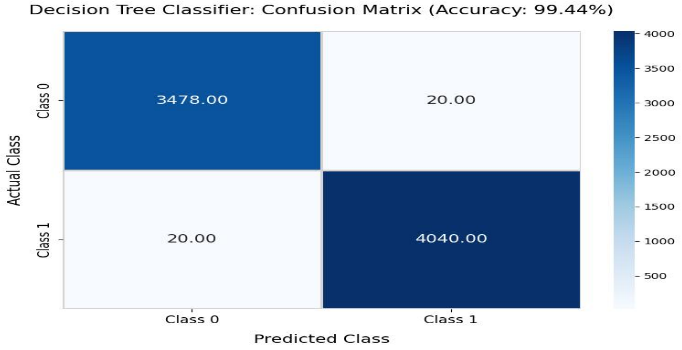
The Random Forest classier improves the prediction accuracy by combining the outputs of multiple decision trees. A random subset of the data and features is used to train each tree. The Random Forest uses the majority voting mechanism to accrue impressive results achieving an accuracy of 98.72%. This means that about 98.72% of predictions were correct regarding this model. In addition, the versatile Decision Tree Classifier builds a tree-like model using the other data attributes to make various decisions. It can also be used for both classification and regression, which demonstrates its paramount importance in supervised learning.

FIGURE 3: Accuracy of Random Forest classifier. The figure illustrates the classification performance of the model, showcasing its effectiveness in distinguishing between different network traffic patterns.



III Evaluation Metrics and Validation Method

FIGURE 4: Accuracy Score of decision tree classifier. The figure represents the model's performance in classifying network traffic, highlighting its ability to detect anomalies and cyber threats.



In the effort to enhance the performance of classifiers like Random Forest (RF), Naive Bayes (NB), Gradient Boosting (GB), Boost (XGB), and Decision Trees (DT), carefully selection of parameter settings is pivotal. Employing a random search procedure, we explore various datasets to identify the optimal input parameters. The evaluation process begins by dividing the dataset into training and testing subsets. This study specifically utilizes key metrics

such as accuracy, precision, F1 score, and recall score to evaluate the classifier's performance, as mathematically represented in equations (1-4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (3)$$

$$F1\ Score = \frac{2.\ Precision.Recall}{Precision + Recall} \qquad (4)$$

In terms of evaluation for classifiers, these metrics are: True Positive (TP), which defines the attack that is correctly predicted; True Negative (TN), indicating a correct prediction of the normal instances; False Positive (FP), which refers to the attack that is incorrectly classified as normal; and False Negative (FN), indicating a normal activity that is wrongly classified as an attack. Within this research context, accuracy that measures correctly to total cases was one of the primary metrics.. This performance shows how the Random Forest classifier judges attack and normal occurrences accurately. Because of the tree maps attached that illustrate, they also affirm the fact that the accuracy in the Random Forest model is superior regarding robustness for this application. By evaluating both False Positive (FP), which means an attack wrongly classified as normal, and False Negative (FN), indicating a normal activity that is wrongly classified as an attack. Such were the metrics for evaluating classifiers. Accuracy, which measures correctly classified cases to the total cases, was among the much-used metrics in the research.

The two classifiers of Decision Tree and Random Forest really demonstrated high accuracy, as pictured clearly above in **FIGURE 4** and **FIGURE 3** Whereas the Decision Tree had clear individual predictions, the Random Forest classifier aggregated many decision trees that made their own decisions with an impressive grand total of 99.5%. This performance clearly indicates that the Random Forest classifier judges attack and normal occurrences very accurately. Tree maps attached also illustrate and validate the point that Random Forest model accuracy is superior in terms of robustness as per the requirements of this application.

III. RESULTS AND ANALYSIS

This section presents the detailed results of our machine learning models trained and tested on two datasets: the Benchmark dataset and the custom-built Net-Sec dataset. We evaluate each classifier's performance using key metrics such as accuracy, precision, recall, and F1 score. Additionally, visual representations in the form of charts are included to provide a clear comparison between training and testing phases for each metric.

TABLE 3 PERFORMANCE METRICS FOR BENCHMARK DATASET

Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Logistic Regression	Training: 0.7920	Training: 0.79208	Training: 0.7920	Training: 0.7920
	Testing: 0.7785	Testing: 0.8880	Testing: 0.7785	Testing: 0.8004
KNN	Training: 0.9314	Training: 0.9314	Training: 0.9314	Training: 0.9314
	Testing: 0.9030	Testing: 0.9247	Testing: 0.9030	Testing: 0.9040
Decision Tree	Training: 0.9996	Training: 0.9996	Training: 0.9996	Training: 0.9996
	Testing: 0.9869	Testing: 0.9872	Testing: 0.9869	Testing: 0.9870
Random Forest	Training: 0.9995	Training: 0.9875	Training: 0.9872	Training: 0.9995
	Testing: 0.9872	Testing: 0.9995	Testing: 0.9872	Testing: 0.9080
XGBoost	Training: 0.9055	Training: 0.9055	Training: 0.9055	Training: 0.9055
	Testing: 0.8928	Testing: 0.9216	Testing: 0.8928	Testing: 0.8945

Naïve Bayes	Training: 0.7598	Training: 0.7598	Training: 0.7598	Training: 0.7598
	Testing: 0.7474	Testing: 0.9170	Testing: 0.7474	Testing: 0.7988

A. MODEL PERFORMANCE FOR BENCHMARK DATASET

The Benchmark dataset, refined from the original CICIoT [25] dataset, was employed to evaluate various machine learning models. The performance metrics for different classifiers are summarized in **TABLE 3 Error! Reference source not found.**, and they provide a quantitative assessment of each model's ability to identify network anomalies.

B. EVALUATION OF CLASSIFIER PERFORMANCE ON THE BENCHMARK DATASET

The classifiers show good performance regarding different metrics, where such metrics as precision and recall go beyond 90% for most of the models. These two metrics are quite important as far as anomaly detection is concerned since they minimize both false positives and false negatives. Decision Tree and Random Forest models are also worth mentioning, on the one hand, they have shown perfect accuracy during training, and on the other hand, they have shown a slight decrease in their accuracy during testing.

C. MODEL PERFORMANCE ON NET-SEC DATASET

The Net-Sec dataset, designed to simulate realistic network threats such as ARP, Sweep, and DDoS attacks, was used to further test the classifiers. Performance metrics for this dataset are presented in TABLE 4.

TABLE 4: PERFORMANCE METRICS FOR NET-SEC DATASET.

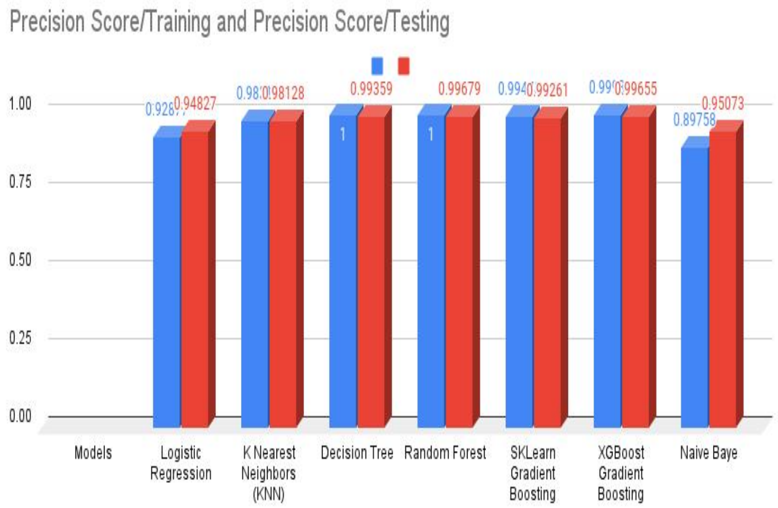
Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Logistic Regression	Training: 0.9287	Training: 0.9287	Training: 0.9287	Training: 0.9287
	Testing: 0.9231	Testing: 0.9482	Testing: 0.9121	Testing: 0.9298
KNN	Training: 0.9831	Training: 0.9831	Training: 0.9831	Training: 0.9831
	Testing: 0.9813	Testing: 0.9812	Testing: 0.9839	Testing: 0.9084
Decision Tree	Training: 1.0000	Training: 1.0000	Training: 1.0000	Training: 1.0000
	Testing: 0.9948	Testing: 0.9935	Testing: 0.9967	Testing: 0.9084
Random Forest	Training: 1.0000	Training: 1.0000	Training: 1.0000	Training: 1.0000
	Testing: 0.9964	Testing: 0.9967	Testing: 0.9965	Testing: 0.9084
XGBoost	Training: 0.9993	Training: 0.9993	Training: 0.9993	Training: 0.9993
	Testing: 0.9962	Testing: 0.9965	Testing: 0.9965	Testing: 0.9965
Naïve Bayes	Training: 0.8975	Training: 0.8975	Training: 0.8975	Training: 0.8975
	Testing: 0.8970	Testing: 0.9507	Testing: 0.8697	Testing: 0.9084

The results from the above table show that Decision Tree and Random Forest again achieve high performance, with accuracy scores nearing 99%. Despite the addition of more complex network threats in the Net-Sec dataset, these models exhibit robust performance across all metrics. Even in challenging conditions, XGBoost and K-Nearest Neighbors demonstrate their effectiveness in detecting network anomalies**.**

D. EVALUATION OF CLASSIFIER PERFORMANCE ON THE NET-SEC DATASET

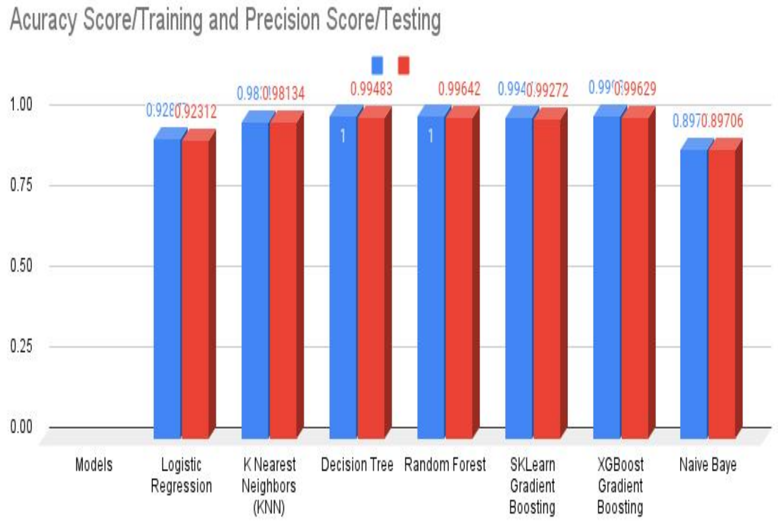
The classifiers tested on the Net-Sec corpus demonstrated consistent performance up to their high levels, where accuracy, precision, recall, and F1 scores are generally above the 90 percent mark for most of the top models. One is Naïve Bayes, which is still moderately accurate but does very well with a few types of attacks. The model should then be useful in simpler detection tasks. The high recall rates indicate that the models could be able to identify true positives, which is a major requirement in network security, where an attack missing could have huge repercussions.

FIGURE 5: Accuracy score of classifiers with respect to Netsec dataset. The figure illustrates the classifier's performance in detecting network anomalies and cyber threats by distinguishing between normal and attack traffic.



To further illustrate the models' performance, the following charts provide visual comparisons between training and testing scores for accuracy, precision, recall and F1 scores.

FIGURE 6: Precision score of classifiers with respect to the NET-SEC dataset. The figure illustrates the precision performance of different classifiers, indicating their ability to correctly identify attack instances while minimizing false positives.



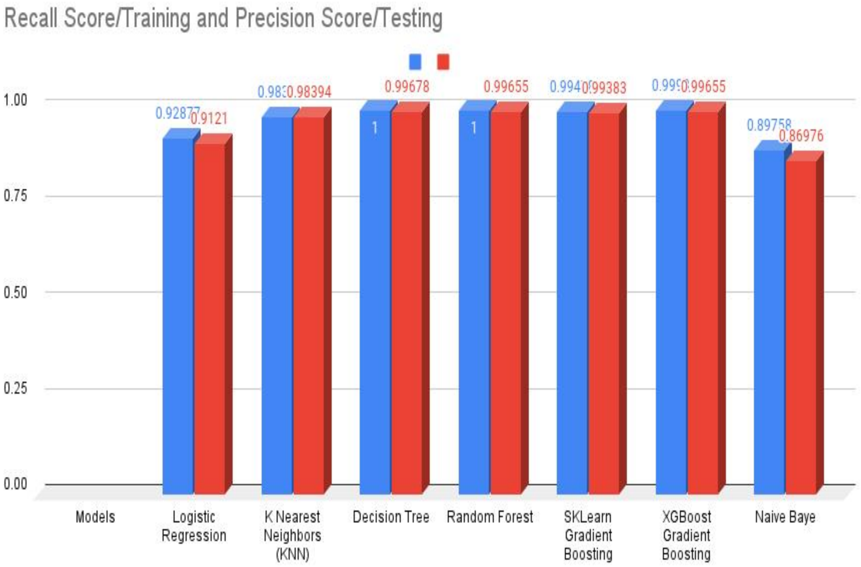
a) Accuracy Chart

The chart shows the training and test accuracy of each model with respect to the Nets-sec dataset revealed. Clearly, the best models of the group possess a high level of accuracy at both times, followed by minute drops when tested.

b) Precision Chart

The precision report brings out the classification of a case being positive by models into the limelight. Random Forest and XGBoost both present nearly complete coverage. Nevertheless, the other classifiers prove reliable in additional ways, representing good classification abilities.

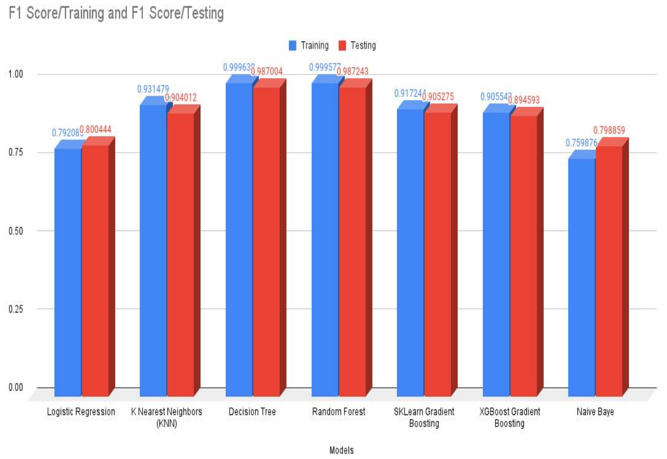
FIGURE 7: Recall score of classifiers with respect to the NET-SEC dataset. The figure highlights the classifiers' ability to correctly identify attack instances, emphasizing their effectiveness in minimizing false negatives.



c) Recall Chart

The recall chart presents how effective the models are in identifying true positive instances. This metric becomes especially important in identifying security threats. The graph of recall indicates how well models could identify the true positive instances. Measure becomes more important while identifying security threats. The recall result shows an overall strong performance with high sensitivity.

FIGURE 8: F1 score of classifiers with respect to the NET-SEC dataset. The figure represents the balance between precision and recall, providing a comprehensive measure of the classifiers' effectiveness in detecting network threats.



d) **Accuracy Chart:** The F1 score balances precision and recall, offering a more comprehensive evaluation of the models. The chart reveals that Decision Tree and Random Forest achieved the highest F1 scores, reflecting their overall superior performance in identifying and classifying network attacks.

Analysis: The Benchmark and Net-Sec results reflect the successful application of sophisticated machine learning models such as Random Forest, XGBoost, and Decision Tree towards detecting difficult network security threats. These models all gave performance results that, In most cases, outscored performances from the simple algorithms, notably Naïve Bayes, particularly in scenarios involving complicated attack patterns. The high numbers indicated, particularly in terms of recall, may suggest the kind of worth they hold in a real-life scenario, whereby accurate identification of the network anomaly is required. However, such slight overfitting in the Decision Tree and Random Forest models suggests the need for further tuning for generalization purposes.

CONCLUSION

This study focused on developing Intrusion Detection Systems (IDS) for IoT devices using machine learning techniques. A custom dataset was created to simulate a range of normal and attack scenarios, that will benefit the researchers in developing and examining different types of attacks. IDS models were implemented on a Raspberry Pi, demonstrating their viability on resource-constrained devices. Experiments are investigated using Benchmark and NetSec datasets, and their performance is evaluated based on various metrics. Various algorithms, including decision trees, random forests, support vector machines (SVM), and neural networks, were evaluated for their performance within IoT constraints. The observed accuracy, Decision Tree and Random Forest shows the superiority of proposed methods than the existing methods. The findings affirm the practicality of using IDS in IoT environments while highlighting the importance of optimizing models to balance accuracy with the management of false positives and negatives. Future research aims to find advanced anomaly detection methods, optimize models for edge computing, and investigate federated learning approaches in evolving cyberattacks. In future work, we aim to utilize deep learning algorithms to assess model performance across various datasets. This work paves the way for improving IDS to address the specific challenges of IoT networks.

REFERENCES

[1]M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie, and T. Shongwe, “Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey,” IEEE Access, vol. 11, pp. 145813–145852, 2023, doi: 10.1109/ACCESS.2023.3346299.

[2]J. Ashraf et al., “IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities,” Sustain Cities Soc, vol. 72, Sep. 2021, doi: 10.1016/j.scs.2021.103041.

[3]P. S.-2017 I. S. C. (IntelliSys) and undefined 2017, “ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things,” ieeexplore.ieee.orgP Shukla2017 Intelligent Systems Conference (IntelliSys), 2017•ieeexplore.ieee.org, Accessed: Dec. 10, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8324298/

[4]A. Sivanathan, D. Sherratt, H. H. Gharakheili, V. Sivaraman, and A. Vishwanath, “Low-cost flow-based security solutions for smart-home IoT devices,” ieeexplore.ieee.orgA Sivanathan, D Sherratt, HH Gharakheili, V Sivaraman, A Vishwanath2016 IEEE International Conference on Advanced Networks and, 2016•ieeexplore.ieee.org, Accessed: Dec. 10, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7947781/

[5]S. Mohamed and B. Abdelbasit, “Cybersecurity Attacks Detection For MQTT-IoT Networks Using Cybersecurity Attacks Detection For MQTT-IoT Networks Using Machine Learning Ensemble Techniques Machine Learning Ensemble Techniques”,

Accessed: Sep. 19, 2024. [Online]. Available: https://repository.rit.edu/theses.

[6]M. A. Khan et al., "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT," Sensors 2021, Vol. 21, Page 7016, vol. 21, no. 21, p. 7016, Oct. 2021, doi: 10.3390/S21217016.

[7]A. Thakkar and R. Lohiya, "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges," Archives of Computational Methods in Engineering, vol. 28, no. 4, pp. 3211–3243, Jun. 2021, doi: 10.1007/S11831-020-09496-0.

[8]N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," pp. 127–156, 2017, doi: 10.1007/978-3-319-59439-2_5.

[9]M. Al-Hawawreh, F. Den Hartog, and E. Sitnikova, "Targeted Ransomware: A New Cyber Threat to Edge System of Brownfield Industrial Internet of Things," IEEE Internet Things J, vol. 6, no. 4, pp. 7137–7151, Aug. 2019, doi: 10.1109/JIOT.2019.2914390.

[10]E. P. Nugroho, T. Djatna, I. S. Sitanggang, A. Buono, and I. Hermadi, "A review of intrusion detection system in IoT with machine learning approach: current and future research," ieeexplore.ieee.orgEP Nugroho, T Djatna, IS Sitanggang, A Buono, I Hermadi2020 6th international conference on science in information, 2020•ieeexplore.ieee.org, doi: 10.1109/ICSITech49800.2020.9392075.

[11]"EBSCOhost | 136146833 | A REVIEW ON KDD CUP99 AND NSL NSL-KDD DATASET." Accessed: Dec. 10, 2023. [

[12]"CICIDS2017 Benchmark (Network Intrusion Detection) | Papers With Code." Accessed: Sep. 19, 2024.

[13]N. Moustafa, J. S.-2015 military communications and, and undefined 2015, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," ieeexplore.ieee.orgN Moustafa, J Slay2015 military communications and information systems conference, 2015•ieeexplore.ieee.org, doi: 10.1109/MilCIS.2015.7348942.

[14]N. Moustafa, M. Keshk, E. Debie, and H. Janicke, "Federated TON_IoT windows datasets for evaluating AI-based security applications," Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020, pp. 848–855, Dec. 2020, doi: 10.1109/TRUSTCOM50675.2020.00114.

[15]N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustain Cities Soc, vol. 72, Sep. 2021, doi: 10.1016/j.scs.2021.102994.

[16]A. Churcher, R. Ullah, J. Ahmad, F. Masood, M. G.- Sensors, and undefined 2021, "An experimental analysis of attack classification using machine learning in IoT networks," mdpi.comA Churcher, R Ullah, J Ahmad, F Masood, M Gogate, F Alqahtani, B Nour, WJ BuchananSensors, 2021•mdpi.com, Accessed: Dec. 10, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/446?src=1257042

[17]E. Anthi, L. Williams, S. Słowi´nska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," ieeexplore.ieee.orgE Anthi, L Williams, M Słowińska, G Theodorakopoulos, P BurnapIEEE Internet of Things Journal, 2019•ieeexplore.ieee.org, Accessed: Dec. 10, 2023.

[18]M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Access, vol. 8, pp. 140699–140725, 2020, doi: 10.1109/ACCESS.2020.3013541.

[19]W. Samek, T. Wiegand, and K.-R. Müller, "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models," Aug. 2017, Accessed: Feb. 01, 2025. [Online]. Available: http://arxiv.org/abs/1708.08296

[20]L. Yang and A. Shami, "IoT data analytics in dynamic environments: From an automated machine learning perspective," Eng Appl Artif Intell, vol. 116, Nov. 2022, doi: 10.1016/J.ENGAPPAI.2022.105366.

[21]M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Dec. 2009, doi: 10.1109/CISDA.2009.5356528.

[22]N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Information Security Journal, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

[23]R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," Computer Networks, vol. 34, no. 4, pp. 579–595, Oct. 2000, doi: 10.1016/S1389-1286(00)00139-0.

[24]"A detailed analysis of cicids2017 dataset for designing... - Google Scholar." Accessed: Dec. 10, 2023.

[25]E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors 2023, Vol. 23, Page 5941, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/S23135941.

[26]R. Panigrahi, & S. B.-I. J. of E., and undefined 2018, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," researchgate.netR Panigrahi, S BorahInternational Journal of Engineering & Technology, 2018•researchgate.net, vol. 7, no. 3, pp. 479–482, 2018, Accessed: Dec. 10, 2023