International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 2
ISSN: 2088-5334

IDS Based on Machine Learning in Reaction to IoT Attacks: Review and Empirical Evaluation

Abdelouahed Bamou [a,*], Moulay Driss El Ouadghiri [a], Badraddine Aghoutane [a], Loukmane Maada [a]

[a] *IA Laboratory Computer Science Department, Science Faculty, University Moulay Ismail, Zitoune, Meknes, 50070, Morocco*
Corresponding author: [*]ab.bamou@edu.umi.ac.ma

***Abstract*—Recently, connected objects have been the subject of cyber-attacks at an alarming rate. These devices connected to a vast volume data stream have insufficient resources and are not manually configured. Typically, attacks target the usability and exploitation of these vulnerabilities. These attacks make the mission of traditional intrusion detection (IDS) systems more challenging to limit intrusion threats. Machine learning (ML) can solve this problem, mainly since the Internet of Things (IoT) can collect and transfer massive amounts of data. This data is the essence of ML, enabling it to build security and privacy models which can predict or classify malicious nodes and network traffic in the IoT. This article looks at the more common forms of cyberattacks, which could lead to an IoT system failure, as well as a countermeasure capable of limiting their damage. First, we present a general review of IDS and these evaluation measures as a solution to limit these attacks. After reviewing the ML domain and these often-used algorithms, on which the IDS can be based to accomplish its mission, we examine the different datasets researchers use to form their IDS. Finally, we look at a practical example of using Python to evaluate ML methods on a current dataset (TON IoT). The research is based on previous research on the topic. The results enable us to choose the appropriate algorithms for the IDS to achieve the best binary and multi-classification based on the evaluation parameters.**

***Keywords*— IoT security; IDS; evaluation metrics; machine learning algorithms; attacks and threats in IoT; datasets for IDS in IoT; classification.**

Manuscript received 9 Apr. 2022; revised 7 Nov. 2022; accepted 27 Dec. 2022. Date of publication 30 Apr. 2023.

I. INTRODUCTION

IoT systems can now be found in businesses worldwide, as well as in a variety of other aspects of our daily life. IoT devices will continue to grow exponentially in response to global market demand. It is defined as a collection of objects with electronic components, software, sensors, and actuators linked over the Internet to assemble and transfer data using various communication protocols, resulting in a network of intelligent machines [1].

IoT devices can be found in a variety of settings, including smart cities, smart homes, medical equipment, and smart cars. IoT technologies generate massive amounts of data. This information is crucial to artificial intelligence systems, and it can assist everyone with a variety of tasks. However, the growth of IoT applications has brought in a slew of issues. IoT security is one of them that cannot be neglected. Thus, each layer of the IoT architecture can represent several vulnerabilities that can be exploited to launch a vast panoply of attacks. For example, an attacker can collect data that is classified as confidential from any important organization [2]. Since the devices are put in different locations, there are environmental risks such as wind, snow, rain, or accidental damage. In addition, saved data can be hijacked through a physical attack, unauthorized access (identity theft, ...), connectivity (poor quality of service, ...), and data exchange (shut-down due to network flooding, ...). Because of the scarcity of resources, communications are easy to trace and intercept (ciphertext attack, Man In The Middle (MITM)). Protocol attacks target resource consumption and network topology using malicious nodes. Furthermore, the application layer is prone to bugs, particularly insecure login identifications. Viruses, worms, Trojans, and cloud applications may be infected [3].

ML is a powerful tool for prospecting records to discover connected objects' usual and distinctive behaviors when communicating with each other or the outside. Indeed, the assembled data of each IoT object can be processed to identify the normal pattern of these interactions and the unusual behavior of these smart devices at an early stage. Furthermore, ML processes can predict future attacks by

training from existing examples, as these are, in most cases, evolutions of preceding attacks [4]. The success of ML practices in fraud detection, text sorting, and image understanding, where ML algorithms can be used to predict or classify the subjects under study, has prompted security academics to use these procedures for IDSs to develop IoT security [5]. These algorithms are provided data from training datasets with benign and malicious instances. By picking a recent and specialized IoT dataset to perform multi-dimensional classification tests, our goal in this research is to conduct a theoretical and practical study of ML-based IDS to adopt the best ML algorithms to recognize attacks on IoT networks.

The remainder of this work is outlined. We start with a detailed examination of the most common IoT attacks. Then we move on to a general examination of IDS and their evaluation metrics before moving on to the ML domain, which includes the most commonly used algorithms and the various datasets used in the literature. Finally, a practical example of IDS based on ML is presented.

A. Attacks and Threats in IoT Systems

IoT and Industrial IoT (IIoT) security threats can be studied in several ways [6], [7]. Surface attack: Classifies attacks according to IoT architecture layers: Physical layer, networks layer, Internet layer, transport layer, and application layer. Effects of the attack: The effects of the attack concern the security and confidentiality of the data; they are classified as identification, authorization, accessibility, confidentiality, and integrity.

Types of attacks: Physical attacks, in which the attacker has physical access to the device to manage or even disrupt the IoT service, and cyber-attacks, in which the attacker seeks to affect the exchange of information between connected objects, are split into passive and active attacks: Passive attacks are characterized by spying in the correspondence flows or the system, in other words, the authenticity and confidentiality of communications are compromised, the attacker can capture data from the instrument, and owners track and locate connected objects—for example, Traffic sniffing and Port scanning. While in Active threats, the attack is not limited to listening to communications, but it also has the power of modifying and even cutting through several techniques of disturbance and alteration, like DOS or DDOS attacks if they come from multiple resources. Masquerade attacks, Message replay [8].

The following section presents the most common cyber-attacks in the IoT:

1) DOS and DDoS attacks: DoS (Denial of Service) attacks are used to prohibit legal users from retrieving the service. It makes use of vulnerabilities in IoT systems with limited resources (bandwidth, CPU, energy, etc.) to flood them with a huge range of possible destructive data streams. This attack requires an Internet link and a solitary system to attack the object [9]. DDoS (Distributed DoS) attacks are similar to DoS attacks, with the exception that they are carried out by botnets comprising many systems in different places. These are networks of devices connected to the Internet and controlled by the attacker [10]. Due to the vast number of connected objects dispersed geographically, and their low levels of security, DDoS attacks are the most common in the IoT [11]. They can be carried out in several ways [3][10]:

- HTTP flood, TCP SYN, and UDP or ICMP flood attacks: are based on a huge packet flooding the connection that the victim does not support to compromise its bandwidth and prevent legitimate users from accessing the servers [4][12].
- Teardrop Attack: This attack exploits the fragmentation principle of the IP protocol. The victim does not reassemble packets using the incompatible packet offset values they contain. This attack causes the systems to be planted [13].
- Ping of death: This type of attack pings the victim with IP packets whose size is greater than the maximum allowed by IP packets so that the victim cannot reassemble them [14].
- Smurf Attack: This attack is based on the use of broadcast servers to paralyze a network. The concept is to send a maximum flow of ICMP ECHO (ping) packets to the broadcast addresses. Each ping contains the victim's spoofed address [15].

2) Keylogging: are recorders of activities performed by a user. The goal is to not only track current work but also to retrieve sensitive credentials. The operating mode of the keyloggers is as follows: They are installed remotely via a Trojan horse in the connected object, then information theft malware allows the missions to be completed. They are, therefore, useful in cases of eavesdropping [16].

3) Probing Attacks: Also called reconnaissance or Scanning attacks. This attack combines the operating system (OS) scans and port scanning services. The scanner allows an attacker to observe all of a machine's open ports as well as the operating system that has been utilized; the result of this attack gives important information about the attacked system, for example, is it a connected object or not? It makes it possible to test various known flaws on OS and these services, to find vulnerabilities linked to this system [17].

4) Attacks on IoT protocols: IoT systems use lightweight protocols to deal with their resource limitations. These protocols are subject to attacks that exploit their internal structures to impact the communication channel as well as the communicated data. They can be classified into three groups [18].

Communication protocol-based attacks [19]: they try to exploit changes that occur during transitory phases between nodes during a communication. They include:

- Confidentiality attacks: This protocol is vulnerable to spoofing, eavesdropping, and MITM attacks due to the lack of data encryption.
- Authentication attacks: any node, including the malicious ones, can join the network and gain legitimate access because this protocol does not have an authentication system.
- Replay attack: If an intruder retrieves the packets, he can retransmit them as regular traffic but with changed content, as if a valid sender sent them.
- Sniffing: when an attacker manages to capture all the packets circulating in the network (user identity and their passwords….).
- Flooding attacks and the pre-shared key.

Attacks related to network protocols: they can happen during the connection phase. We find attacks such as [20]:

- Wormhole: An attacker can create a low latency private link between two malicious network nodes to pass sensitive information, thus causing degradation of traffic and network resources and a disorder of its topology.
- Sinkhole: In this case, the attacker draws all traffic to the compromised node, posing as the shortest route.
- Blackhole: Similar to a sinkhole, however, this time, the compromised network point quietly makes the traffic disappear without notifying the source.
- Selective Forward: Similar to the previous two, except that the attacker drops some particular packets from network traffic and forward all the others.

5) Data-related attacks [21]*:* affect the security of transmitted data, for example :

- Data exposure*:* Occurs when it is possible to inadvertently expose sensitive data (different from a data breach), due to insufficient protection, lack of encryption, or software vulnerabilities.
- Data Exfiltration: These attacks usually infiltrate private networks and extract confidential data for sharing with unauthorized third parties or transfer them to unsecured systems. This kind of attack can lead to the loss of sensitive data like credit card authentication [16].
- Data Corruption occurs when an attacker manages to make the transmitted data unreadable, resulting in a DoS attack.
- Data injection: The attacker can inject unwanted messages while communicating between two devices.

B. Intrusion Detection System

IDS are security mechanisms that aims to enhance the security of IoT systems. They are used to monitor nodes and network traffic in these systems. Intrusion is an undesirable or malevolent activity that is destructive to IoT systems. IDS can be in hardware or software form; it analyses the traffic and classifies the packets into legitimate users or intruders. It can detect known attacks, learned by their signatures, and unknown attacks by identifying abnormal network activities. The goal is to warn users of various attacks threatening their IoT networks, whether internal or external. Internal when the attacker is part of this network, while external when he is outside of it [22].

To build an IDS, it is necessary to develop three modules: the monitoring module, which monitors system resources and network traffic; the analysis and detection module, which corresponds to the core of the IDS and its role is to find attacks based on a; finally, the alarm module informs users of the presence of attacks [23]. The detection module is based on the following detection algorithms [22]: Anomaly, signature, specification, and hybrid approaches.

1) Signature-based detection: IDS recognizes intruders based on attack signatures stored in the system database. This strategy is incredibly effective and rapid when it comes to identifying known attacks, but he finds it challenging to identify unusual attacks.

2) Anomaly-based detection: This technique compares normal recorded behavior with abnormal network activities to this model. Here in this approach, the use of ML algorithms finds its place to allow the IDS to determine the consistent traffic activity pattern. Anomaly-based IDS successfully prevents unknown attacks, but it suffers from a precision issue as it often happens to place a legitimate data stream as malicious (false positive).

3) Specification detection: this is similar to Anomaly detection*,* and therefore, in this case, the entered specifications are generated manually to assess typical behavior and deviations in an IoT system. This method reduces the high rate of false alarms. But it requires experts and rigorous work for each platform or environment. Inappropriate specifications result in a rate of true negatives and false positives and, therefore, a limited efficiency of the IDS in its mission.

4) Hybrid detection: This form of IDS mixes signatures and anomalies techniques, resulting in increased accuracy; however, running the two modules in parallel needs additional computational resources. In reality, the hybrid model is the most frequent and practical because the attacks are discovered by the anomaly's method, and then they are taken care of by the signature-based method.

The anomalies method for IDS is most fit for ML algorithms because it is a classification problem between normal or abnormal states rather than multi-classification within different types of attacks. However, things are not as simple as they appear, as the ML algorithms can produce enough false positives, rendering IDS useless. It is, therefore, important to identify the models that produce the highest accuracy [24].

C. IDS for IoT systems-based ML

ML is a powerful method of mining data collected from every part of the IoT system, which can be studied to determine normal patterns and early-stage malicious behaviors. Researchers have demonstrated the success of ML models for IDS, and they can cope with the ever-growing IoT network and numerous zero-day attacks. Moreover, ML methods could intelligently predict new unknown attacks from existing examples [25].

In this part, we'll go through the various ML techniques utilized in IDS for IoT and the many Datasets used in the IDS's learning phase.

1) ML algorithms used in IDS: ML may be divided into two forms depending on the kinds of training data: supervised, where the input parameters have needed outputs, and unsupervised, where the data is unlabeled, with each category having many IDS models based on ML (Fig. 1).

Naive Bayesian (NB), Decision Trees (DT), Support Vector Machines (SVM), k-Nearest Neighbor (KNN), Random Forest (RF), and Ensemble learning (EL) are examples of supervised techniques.

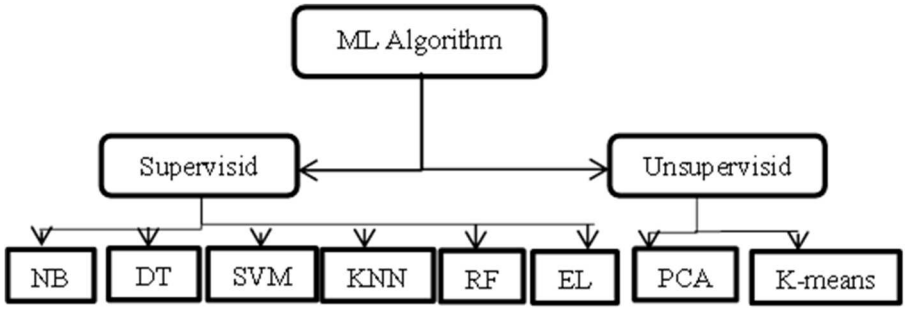

Fig. 1 Machine Learning Algorithms

k-Means clustering, and Principal component analysis (PCA) are examples of unsupervised methods. The article [22] explains the intricacies of supervised algorithms and some recent work in this field.

2) Datasets for IDS in IoT: Data is the most important component of machine learning, and a good Dataset is a basis for building a successful Machine Learning model. The data in this dataset should be large enough to reflect the problem we are looking to solve, as well as high-quality and clean of errors [26].

In this item, we will get diverse Datasets used in the literature to develop an IDS-based ML. KDD Cup99: is among the earliest datasets for IDS created in 1998 and then upgraded in 1999 by DARPA, which used a benchmarking environment for realistic and comprehensive IDS. This dataset has the advantage of initiating research in the field, but the attack records they contain are outdated and not specific to the IoT environment. It has 41 features and 411 033 372 benign records in all plus the entirety of malicious records 4,176,086, Bhati et al. [17] used this dataset.

NSL-KDD: After removing redundant records from the KDD cup99 dataset in 2009, it was formed with 125,973 records, 22 attacks for all training data and 22,544 records, and 17 attacks for test data. This dataset includes 43 features, and the attacks are organized in the 4th class but are not modern and specific to the IoT environment [27].

ISCX-2012: This dataset was created in 2012, according to the analysis of HTTP, IMAP, POP3, SMTP, FTP, and SSH protocols in real and labeled network traffic, contains various attack scenarios. The traffic recorded on the network lasted seven days. This dataset has five different kinds of threats, eight different features, 2,381,532 safe records, and 68,792 bad records. This dataset fails to include modern network protocol traffic and is not special to IoT [28].

UNSWNB15: Created by Moustafa & Slay in 2015 [29], this dataset is a network intrusion packet capture (pcap), CSV file, pre-tagged as an attack or normal. It holds 2,540,044 records in four CSV files: UNSW-NB151.csv, UNSW-NB152.csv, UNSW-NB153.csv, and UNSW-NB154.csv. The training set contains 175,341 records, while the test set contains 82,332 records, separated into 2,218,761 normal records and 321,283 abnormal records. It covers 49 features and 9 attacks: DoS, Generic, Reconnaissance, Analysis, Fuzzers, Worms, Backdoors, Shellcode, and Exploits, but this dataset is not particular to IoT [30].

CICIDS2017: The NetFlowMeter network traffic flow analyzer recorded this dataset. The tool gathers 80 network traffic parameters using tagged streams based on HTTP, FTP, HTTPS, SSH, as well as email protocols. There are 2,273,097 normal records and 557,646 abnormal records in PCAP format. The attacks implemented include, among others: DDoS, Portscan, XXS, SQL Injection…. It varies from previous datasets in that it contains large-scale modern attacks based on real users and complex functionality [31], but this dataset includes missing and redundant data records, in addition to an imbalance issue that leads to low precision of the model, studied. Finally, it is not destined for IoT [32].

N-BaIoT: was produced to address the shortage of botnet datasets used for IoT. Traffic collection is done from 9 authentically commercial IoT devices after injecting two types of attacks into these devices: Mirai and bashlite. It was created in 2018 and included 115 features, 17,936 benign and 831,298 malicious records. This dataset is asymmetrical with malicious records, sufficiently larger than benign ones, and therefore requires pre-processing. It also lacks data logs from operating systems and telemetry data from IoT sensors, both of which are essential in determining the capacity of the IoT security systems under investigation to do their task [33].

Bot-IoT: was created by the UNSW Canberra Cyber Attack Center in 2019 in an environment that incorporates different types of botnet attacks and normal virtual IoT traffic. It provides full packet capture of data with appropriate labels in original file formats (.pcap) extension and (CSV) files. The dataset includes 45 features, 9,543 benign records, 73,360,900 malicious records, and kinds of attacks like DoS, DDoS, Keylogging, data exfiltration, Service Scan, and OS scan [34]. This dataset has the advantage of integrating different IoT systems with various data characteristics and several botnet and malware attacks, but it does not have the Audit files of the operating systems or the hacking vectors against the IoT systems [5].

IoT-23: It is indeed a dataset of IoT device network traffic. Stratosphere Laboratory recently produced it for the advantage of Avast Software Prague. It is a large, real, and labeled dataset offered to researchers, combining benign IoT traffic and data from IoT malware infections. The following is a breakdown of the data collected between 2018 and 2019: 3 recordings of realistic connected object network traffic and 20 recordings of malware running on IoT nodes (pcap files). Data gathering is based on HTTP, IRC, Telnet, DHCP, DNS, and SSL protocols; despite that, we notice the lack of some current protocols like HTTPS. This dataset is organized as follows: 23 characteristics, 11 attack types captured, 30,858,735 benign records, and 294,449,255 malicious records [35].

TON_IoT: It's a modern dataset developed from heterogeneous data sources of seven distinct IoT devices,

including a refrigerator, a weather sensor, a Modbus, a thermostat, a GPS, a garage door, and a motion light at UNSW Canberra. Its objective is to assist researchers with a set of data that may be used to verify the efficacy of cybersecurity systems that use machine learning, such as intrusion detection systems and privacy applications. The data comes from IoT and Industrial IoT (IIoT) service Telemetry and their characteristics, as well as logs from Operating systems (Windows 7 & 10, Ubuntu 14 & 18 TLS) plus Network IoT logs; (thus the name TON). That was created from a realistic model of a medium-sized network. This data was also labeled as regular or attack, and a category feature (marking attack subclasses for multi-classification problems). The records in this dataset are stored in seven different CSV file formats. It has 46 features and nine different types of cyberattacks: DoS, DDoS, ransomware, backdoor, XSS: Cross-site Scripting, password cracking, scanning, injection, and MITM. [33].

D. Performance Evaluation Metrics

Machine learning for IDS requires a set of measures to evaluate the system's performance. An extensive set of measurements has been employed in several investigations. This study makes use of the accuracy, precision, recall, F1 score, and ROC-AUC Score [36]:

Accuracy: Although it is inefficient in some cases, accuracy is the most commonly employed of these measurements. It's the ratio of real intrusion detections to the overall number of predicted intrusions. It has the following formula:

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

TN: True-negative. FP: False-positive

Precision is defined as the capacity to distinguish between intrusions and regular behavior. It's the proportion of intrusions that are successfully categorized to the overall number of entries. That is, how many of the findings projected as positive by an IDS are truly positive? It's computed by this equation [37]:

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

Recall: it represents the number of correctly classified predictions produced from all positive instances in the dataset. This formula is used to figure it out:

$$Re\,c\,all=\frac{TP}{TP+FN} \quad (3)$$

F1-score: To calculate the performance of an IDS, the F1 score achieves a compromise between accuracy and recall. It gives the entire number of positive class predictions produced from the dataset's positive instances. It may be calculated using the following formula:

$$F1-Score=2\times\frac{Pr\,ecision\times Re\,call}{Pr\,ecision+Re\,call} \quad (4)$$

ROC-AUC Score: It reflects the efficiency function of the receptor, and it depicts the ratio between the rate of true positives (sensitivity) and the rate of false positives (specificity), and the anti-specificity (1 minus specificity). It is a measure of the performance of a binary classifier, and it illustrates how efficiently a model distinguishes negative and positive target classes [38].

E. Related Works

To prepare for this work, we reviewed the papers presented in Table 1 according to their references, datasets, algorithms, and evaluation metrics. The goal is to take advantage of the different techniques used in our study subject and to compare our results with those of the works of literature.

II. Materials and Methods

After presenting the theoretical part of our IDS based ML, we will move on to its implementation. For de materials, the suggested system performance analysis was built in Python 3.7, and the tests were run on a PC with an Intel Core i5-5300 CPU and 8 GB of RAM. For the methodology (Fig 2), we first select the dataset, then perform its pre-processing, split the data into 80% for training and the other for testing, and finally, use the ML algorithms for binary and multi-classification.

TABLE I
ARTICLES USED IN THE LITERATURE REVIEW

Article	Dataset	Algorithms	Evaluation metrics
A. Alsaedi et al. [39]	Network _Ton_IoT	RF, NB, SVM, LR, KNN, LDA, CART, LSTM.	Accuracy, Recall, Precision, F-measure.
A. Churcher et al. [40]	BoT_IoT	ANN, Logistic _Regression, RF, SVM, DT, NB, KNN.	Accuracy, F1- Score, Recall, Log Loss, ROC, AUC, Precision.
A. Khraisa et al. [41]	BoT_IoT	C4.5, NB, RF, MLP, SVM, CART, KNN.	Accuracy
A. Alhowaide et al. [38]	NSL-KDD, UNSWNB15, BoT_IoT,	El	Accuracy, ROC-AUC F-Score, Variance Efficiency Score.
P. Kumar et al. [42]	Network Ton-IoT	DT, NB, RF, El.	Confusion matrix, precision, F1-score, Accuracy. ROC curve, False alarm rate, Detection rate.
Tim M. Booij et al. [43]	Network _Ton-IoT	GBM, RF, MLP.	Accuracy, Mean Square Error, Gini measures, F1 score, AUC.
Nour Moustafa [33]	Network _Ton-IoT	GBM, RF, NB, Deep Neural Network	ROC_curve, Confusion matrix.

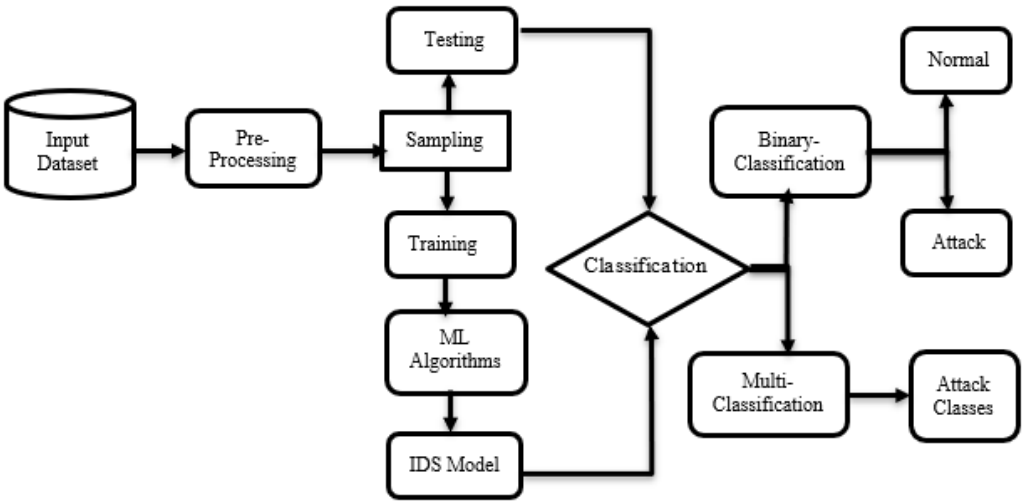

Fig. 2 Diagram of the proposed IDS

A. Input Dataset

After studying different datasets in section I.C.2, we chose Ton-IoT dataset because it is a specific dataset for IoT. Recently, it contains two features for Attack: 'label,' categorized as normal or attack, as well as 'type,' denoting subclasses of threats for multi-class classification tasks, and diverse attack scenarios; in addition, it holds Telemetry data of IoT and Distinct dataset one per IoT device. Because of the heterogeneity (OS logs, pcap files, sensor data, and Bro logs) and relevance of the different sorts of attacks attempted on more varied IoT devices, Tim M. Booij et al.[43] proved that the ToN_IoT dataset is the best available collection for IoT network IDSs. Further, they found a good balance in the distribution of the functionalities between the test and training sets which is important for the performance of ML applications.

1) Dataset Description: We only used part of the four parts of the TON_IoT dataset, i.e., the Train_Test_IoT dataset. The Features Description is followed in [39]. For example, Table 2 gives these the IoT Fridge:

TABLE II
IOT FRIDGE ACTIVITY

ID	Feature	Description
1	Date	Date of recording data.
2	Time	Time of recording data.
3	fridge_ temperature	Temperature measurement of the connected object.
4	temp_ condition	Temperature settings of a connected refrigerator, expressed by low or high according to a given threshold.
5	Label	Attack or normal where 1 specifies attacks and 0 specifies normal.
6	Type	Attack classes, like DDoS, ransomware, injection, password backdoor, normal…

The attacks used in this dataset are Dos and DDOS, Scanning attacks, Injection attacks, XSS, Ransomware attacks, Backdoor attacks, Password and MITM attacks distributed as shown in Fig. 3. The Train_Test IoT records dataset comprises 396,119 malicious and benign data records [39], with 62% being normal (as shown in Fig. 3) and the rest being for various attacks.

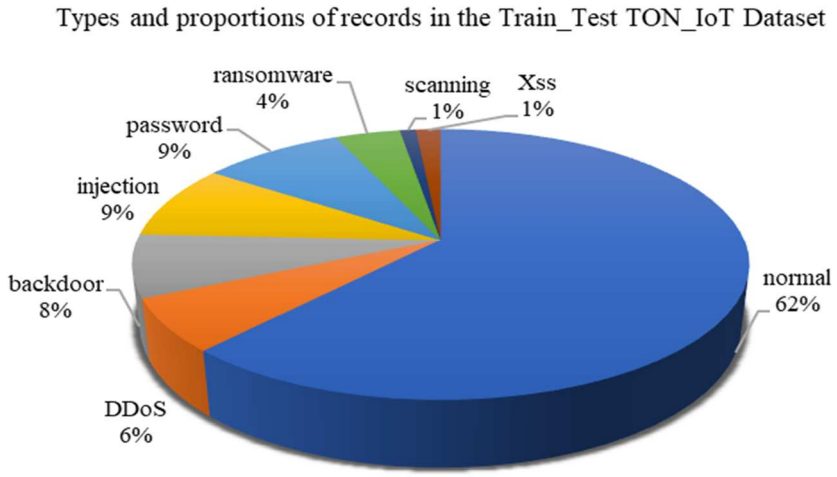

Fig. 3 Training and testing record statistics in Train_Test IoT records.

B. Pre-Processing:

In this part, we started with data cleaning, such as filling in the null values. Then comes the normalization step for numeric features and coding for categorical data because the types of features in the TON_IoT dataset are varied (categorical and numeric at a different scale), the normalization is performed by the min-max method given by equation 5:

$$Xnorm = \frac{X\text{-}Xmin}{Xmax\text{-}Xmin} \in [0,1] \quad (5)$$

Categorical data are coded with the integer of their index to transform them into numerical values (1,2,3, …). The Garage Door dataset has states open/close in their records, and the Motion_Light dataset has on/off in their records. In addition, all datasets have the features Label for the normal/attack state and Type for the attack classes.

C. ML Algorithms for the classification:

The TON_IoT dataset offers two output variables: For binary classification, it uses "label," while for multi-class classification, it uses "type". To achieve this classification, we use in this study the following classical algorithms: SVM, NB, KNN, DT, RF; and then we will combine these methods to improve the results of machine learning in Ensemble Learning (EL).

Several EL methods are most popular:

- Booster comprises the most common algorithms: AdaBoost and Gradient Boosting.
- Vote consists of creating autonomous algorithms from the training data set. Then a voting classifier encapsulates the model by combining the predictions of these already-created models.
- Stacking: after training each classifier based on the full training set, the classifiers are combined via a meta-classifier. The latter is adjusted according to the outputs of the meta-characteristics of the individual models.

The EL model improves performance a little more. To find it we tested Gradient Boosting, Voting Classifier, Bagging Classifier, and Stacking Classifier; based on the score of the first three algorithms studied and taking RF as a meta-classifier afterward, we took the best result from these models.

III. Result and Discussion

A. Binary Attack Classification

Table 3 summarizes the results of the binary attack classification of attacks based on the metrics: Accuracy, Precision, Recall, and F1-score. The AUC_ROC Curve metric is presented below:

All the models used for IoT Fridge (Fig. 4), Garage_Door, Motion_Light, and Thermostat datasets do not exceed the baseline, and the other metrics (Accuracy, recall, precision…) are low. This results in the inability of these models to perform binary classification tasks, and this may be due to a weak correlation between the input and output features or to a lack of features to predict the output in these datasets.

We exclude the SVM and NB models for the IoT_ Modbus dataset (Fig. 5), which seem useless for binary classification tasks (AUC = 0.5). All other models are far from the baseline. The models' outcomes are excellent in light of the testing methods used. The Ensemble algorithm, which has an AUC=0.95, is the best, followed by RF and DT.

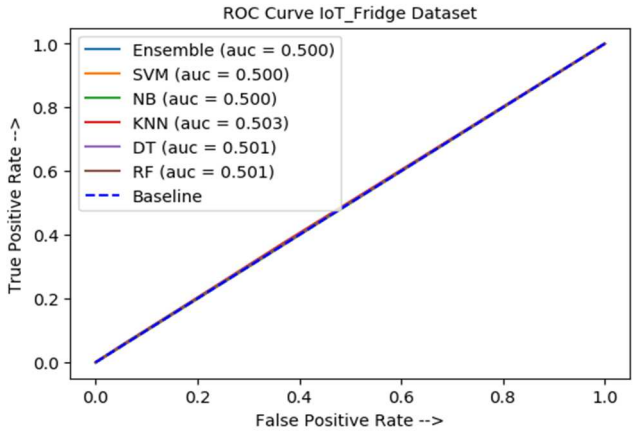

Fig. 4 AUC_ROC Curve of IoT_Fridge dataset

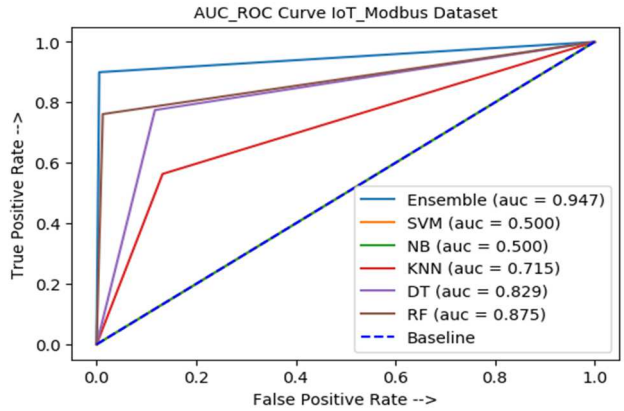

Fig. 5 AUC_ROC Curve of IoT_ Modbus dataset

All models are far from the baseline for the IoT_GPS_Tracker dataset (Fig. 6). In regard to the assessment measures used; the models' outcomes are excellent. The EL and KNN algorithms are the best, with an AUC score of 0.93, followed by RF and DT while SVM and NB have a low score.

Except for SVM (Fig. 7), which appears to be useless for binary classification tasks (AUC = 0.5), all models are distant from the baseline for the IoT_weather dataset. In terms of the evaluations employed, the models' performances are great. With an AUC value of 0.94, the EL algorithm is the best, followed by KNN and RF.

B. Multi-class attacks classification

The results of the multi-class attack classification are presented in Table 4. From this table, it can be seen that in most cases, according to the different metrics used, the EL algorithm outperforms other ML algorithms, followed by RF, DT, and KNN, which also achieve good performance results. However, in the case of the fridge, garage door, and motion lights datasets, the results are not sufficient for all inspected ML algorithms.

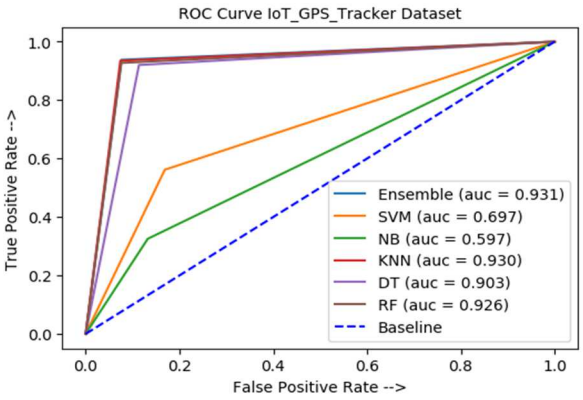

Fig. 6 ROC Curve of IoT_ GPS_Tracker dataset

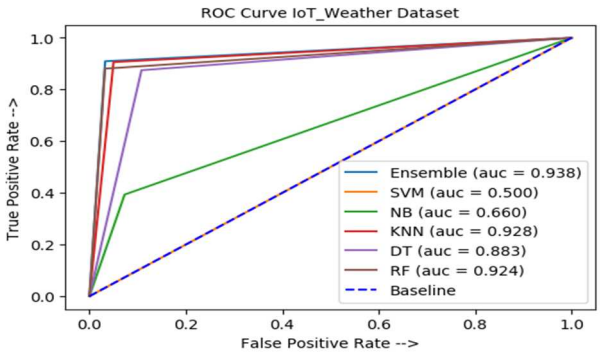

Fig. 7 ROC Curve of IoT_weather dataset

TABLE III

EVALUATION METRICS RESULTS OF THE TRADITIONAL ML ALGORITHMS IN BINARY-CLASS ATTACK CLASSIFICATION.

Datasets	Evaluation metrics	SVM	NB	KNN	DT	RF	EL
IoT Fridge activity	Accuracy	0,52	0.51	0,54	0.50	0.50	0.58
	Precision	0.51	0.51	0.51	0.51	0.51	0.52
	Recall	0.52	0.51	0.54	0.50	0.50	0.59
	F1-score	0.51	0.51	0.51	0.50	0.50	0.46
IoT GPS Tracker activity	Accuracy	0.72	0.65	0.93	0.90	0.93	0.93
	Precision	0.72	0.64	0.93	0.90	0.93	0.93
	Recall	0.72	0.65	0.93	0.90	0.93	0.93
	F1-score	0.72	0.62	0.93	0.90	0.93	0.93
IoT MotionLight activity	Accuracy	0.48	0.48	0.59	0.48	0.48	0.48
	Precision	0.52	0.52	0.76	0.52	0.52	0.52
	Recall	0.48	0.48	0.59	0.48	0.48	0.48
	F1-score	0.48	0.48	0.44	0.48	0.48	0.48
IoT Garage Door activity	Accuracy	0.59	0.59	0.43	0.59	0.59	0.59
	Precision	0.34	0.34	0.50	0.34	0.34	0.34
	Recall	0.59	0.59	0.43	0.59	0.59	0.59
	F1-score	0.43	0.43	0.33	0.43	0.43	0.43
IoT Modbus activity	Accuracy	0.51	0.50	0.82	0.82	0.93	0.97
	Precision	0.58	0.58	0.83	0.82	0.93	0.97
	Recall	0.51	0.50	0.82	0.82	0.93	0.97
	F1-score	0.52	0.51	0.82	0.82	0.92	0.97
IoT Thermo-stat activity	Accuracy	0.66	0.66	0.60	0.56	0.58	0.60
	Precision	0.44	0.44	0.55	0.56	0.56	0.55
	Recall	0.66	0.66	0.60	0.56	0.58	0.60
	F1-score	0.53	0.53	0.57	0.56	0.57	0.57
IoT Weather Activity	Accuracy	0.59	0.71	0.93	0.89	0.94	0.94
	Precision	0.39	0.73	0.93	0.89	0.94	0.94
	Recall	0.59	0.71	0.93	0.89	0.93	0.94
	F1-score	0.44	0.68	0.93	0.89	0.94	0.94

TABLE IV
OUTCOMES OF CLASSICAL ML ALGORITHMS IN MULTI-CLASS ATTACKS CLASSIFICATION.

Datasets	Evaluation metrics	SVM	NB	KNN	DT	RF	EL
IoT Fridge activity	Accuracy	0.58	0.58	0.49	0.58	0.58	0.58
	Precision	0.43	0.43	0.37	0.43	0.43	0.34
	Recall	0.58	0.58	0.49	0.58	0.58	0.58
	F1-score,	0.34	0.34	0.42	0.34	0.34	0.43
	Roc_auc score	0.50	0.49	0.49	0.50	0.50	0.50
IoT GPS_ Tracker activity	Accuracy	0.64	0.61	0.89	0.85	0.87	0.89
	Precision	0.50	0.57	0.89	0.85	0.87	0.89
	Recall	0.64	0.61	0.89	0.85	0.87	0.89
	F1-score,	0.53	0.58	0.89	0.85	0.87	0.89
	Roc_auc score	0.75	0.82	0.95	0.84	0.93	0.96
IoT Motion Light activity	Accuracy	0.59	0.59	0.59	0.59	0.59	0.59
	Precision	0.35	0.35	0.35	0.35	0.35	0.35
	Recall	0.59	0.59	0.59	0.59	0.59	0.59
	F1-score,	0.44	0.44	0.44	0.44	0.44	0.44
	Roc_auc score	0.50	0.5	0.50	0.50	0.50	0.50
IoT Garage Door activity	Accuracy	0.59	0.59	0.59	0.59	0.59	0.59
	Precision	0.34	0.34	0.35	0.34	0.34	0.35
	Recall	0.59	0.59	0.59	0.59	0.59	0.59
	F1-score,	0.44	0.44	0.44	0.44	0.44	0.44
	Roc_auc score	0.50	0.50	0.50	0.50	0.50	0.50
IoT Modbus activity	Accuracy	0.68	0.68	0.72	0.81	0.93	0.96
	Precision	0.78	0.78	0.69	0.82	0.93	0.96
	Recall	0.68	0.68	0.72	0.81	0.93	0.96
	F1-score,	0.56	0.56	0.70	0.82	0.93	0.95
	Roc_auc_score	0.50	0.51	0.88	0.80	0.94	0.96
IoT Thermo-stat activity	Accuracy	0.66	0.66	0.60	0.48	0.48	0.66
	Precision	0.44	0.44	0.47	0.47	0.47	0.44
	Recall	0.66	0.66	0.60	0.48	0.48	0.66
	F1-score,	0.53	0.53	0.52	0.48	0.48	0.53
	Roc_auc_score	0.50	0.51	0.50	0.50	0.50	0.50
IoT Weather Activity	Accuracy	0.62	0.61	0.91	0.83	0.90	0.91
	Precision	0.64	0.52	0.91	0.83	0.90	0.91
	Recall	0.62	0.61	0.91	0.83	0.90	0.91
	F1-score,	0.51	0.53	0.91	0.83	0.90	0.91
	Roc_auc score	0.76	0.82	0.97	0.87	0.98	0.99

To summarize these results, we evaluate proposed models for two classification types using precision, recall, F1-score, precision, and Auc-Roc curves. IDS is expected to score higher on all of these metrics. In our case and according to tables 3 and 4, for the IoT Fridge, Garage_Door, Motion_Light, and Thermostat subsets of this dataset, the results of the metrics turned around 0.5 and the Auc_Roc curve shows that the models have trouble predicting the meaning of the classification. Therefore, these models need to be improved. For the IoT GPS_ Tracker, Modbus, and Weather subsets, the results found are high when testing on these datasets. The algorithms used in the suggested system are classified in order of performance in most of the cases as follows: EL, RF, KNN, DT, then SVM, and NB, which are also found in the literature.

IV. CONCLUSION

This work, which tries to be detailed and exhaustive, presents a theoretical and practical study of IDS-based ML in the IoT domain and proposes an evaluation of ML algorithms that may be used in the core of an IDS to deal with various cyberattacks in the IoT. Based on Accuracy, Precision, F1-Score, Recall, and AUC-Roc curve metrics, the performance of the models based on SVM, NB, KNN, DT, RF, and EL algorithms were compared on the TON_IoT dataset. The results suggest that EL, RF, and KNN algorithms perform much better, while NB and SVM are often the least appropriate. These results can guide researchers in selecting ML algorithms enhancing IDS' capabilities. In the future, we plan to continue studying IDS using deep learning techniques.

NOMENCLATURE

CART: Classification and Regression Trees.
LDA: Linear Discriminant Analysis.
MLP: Multi-layer perception.
LSTM: Long _Short-Term Memory.
C4.5: Chi-square automatic interaction detection
GBM: Gradient Boosting Machine

REFERENCES

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," IEEE Commun. Surv. Tutorials, vol. 23, no. 3, pp. 1622–1658, 2021.

[2] K. Rambabu and N. Venkatram, "Ensemble classification using traffic flow metrics to predict distributed denial of service scope in the Internet of Things (IoT) networks," Comput. Electr. Eng., vol. 96, no. PA, p. 107444, 2021.

[3] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," IEEE Commun. Surv. Tutorials, 2019.

[4] B. B. Gupta, P. Chaudhary, X. Chang, and N. Nedjah, "Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers," Comput. Electr. Eng., vol. 98, no. February, p. 107726, 2022.

[5] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Futur. Gener. Comput. Syst., vol. 100, pp. 779–796, 2019.

[6] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, "A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis," Sensors, vol. 20, no. 13, p. 3625, Jun. 2020.

[7] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," J. Netw. Comput. Appl., vol. 161, p. 102630, Jul. 2020.

[8] N. Zhang, R. Wu, S. Yuan, C. Yuan, and D. Chen, "RAV: Relay Aided Vectorized Secure Transmission in Physical Layer Security for Internet of Things Under Active Attacks," IEEE Internet Things J., vol. 6, no. 5, pp. 8496–8506, Oct. 2019.

[9] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the IoT," J. Inf. Telecommun., vol. 4, no. 4, pp. 482–503, Oct. 2020.

[10] N. Mishra and S. Pandya, "Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review," IEEE Access, vol. 9, pp. 59353–59377, 2021.

[11] A. Bamou, M. Khardioui, M. D. El Ouadghiri, and B. Aghoutane, "Implementing and Evaluating an Intrusion Detection System for Denial of Service Attacks in IoT Environments," in Lecture Notes in Networks and Systems, 2020.

[12] C.-L. Chen and J.-M. Chen, "Use of MARKOV Chain for Early Detecting DDoS Attacks," Int. J. Netw. Secur. Its Appl., vol. 13, no. 04, pp. 01–11, 2021.

[13] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: a survey," J. Supercomput., vol. 76, no. 7, pp. 5320–5363, Jul. 2020.

[14] A. Abdollahi and M. Fathi, "An Intrusion Detection System on Ping of Death Attacks in IoT Networks," Wirel. Pers. Commun., vol. 112, no. 4, pp. 2057–2070, Jun. 2020.

[15] R. Nath N and H. V Nath, "Critical analysis of the layered and systematic approaches for understanding IoT security threats and challenges," Comput. Electr. Eng., vol. 100, p. 107997, May 2022.

[16] P. Podder, M. R. H. Mondal, S. Bharati, and P. K. Paul, "Review on the Security Threats of Internet of Things," Int. J. Comput. Appl., vol. 176, no. 41, pp. 37–45, Jul. 2020.

[17] B. S. Bhati, C. S. Rai, B. Balamurugan, and F. Al-Turjman, "An intrusion detection scheme based on the ensemble of discriminant classifiers," Comput. Electr. Eng., vol. 86, 2020.

[18] Rachit, S. Bhatt, and P. R. Ragiri, "Security trends in Internet of Things: a survey," SN Appl. Sci., vol. 3, no. 1, pp. 1–14, 2021.

[19] N. Abosata, S. Al-Rubaye, G. Inalhan, and C. Emmanouilidis, "Internet of Things for System Integrity: A Comprehensive Survey on Security, Attacks and Countermeasures for Industrial Applications," Sensors, vol. 21, no. 11, p. 3654, May 2021.

[20] R. Yugha and S. Chithra, "A survey on technologies and security protocols: Reference for future generation IoT," J. Netw. Comput. Appl., vol. 169, no. September 2019, p. 102763, 2020.

[21] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of Internet of Things," Comput. Sci. Rev., vol. 38, p. 100312, 2020.

[22] A. Bamou, M. D. E. L. Ouadghiri, and B. Aghoutane, "Intrusion detection in the internet of things," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 1.5 Special Issue, pp. 1–7, 2020.

[23] M. Abdel-Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam, and O. M. Elkomy, "Federated Intrusion Detection in Blockchain-Based Smart Transportation Systems," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 3, pp. 2523–2537, Mar. 2022.

[24] A. Bamou, M. D. EL Ouadghiri, and B. Aghoutane, "Current Works on IDS Development Strategies for IoT," 2022, pp. 15–24.

[25] J. Roldán, J. Boubeta-Puig, J. Luis Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks," Expert Syst. Appl., vol. 149, p. 113251, Jul. 2020.

[26] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, 2019.

[27] S. Sapre, P. Ahmadi, and K. Islam, "A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets Through Various Machine Learning Algorithms," Dec. 2019.

[28] S. Dwivedi, M. Vardhan, S. Tripathi, and A. K. Shukla, "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection," Evol. Intell., vol. 13, no. 1, pp. 103–117, Mar. 2020.

[29] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc., 2015.

[30] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," Cluster Comput., vol. 23, no. 2, pp. 1397–1418, Jun. 2020.

[31] S. Thakur, A. Chakraborty, R. De, N. Kumar, and R. Sarkar, "Intrusion detection in cyber-physical systems using a generic and domain specific deep autoencoder model," Comput. Electr. Eng., vol. 91, no. February, p. 107044, 2021.

[32] Z. Pelletier and M. Abualkibash, "Evaluating the CIC IDS-2017 Dataset Using Machine Learning Methods and Creating Multiple Predictive Models in the Statistical Computing Language R," Int. res. j. adv. eng. sci., vol. 5, no. 2, pp. 187–191, 2020.

[33] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustain. Cities Soc., vol. 72, no. May, 2021.

[34] A. Mihoub, O. Ben Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," Comput. Electr. Eng., vol. 98, no. December 2021, p. 107716, 2022.

[35] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "Anomaly detection model based on gradient boosting and decision tree for IoT environments security," J. Reliab. Intell. Environ., Jul. 2022.

[36] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," Procedia Comput. Sci., vol. 171, pp. 1251–1260, 2020.

[37] M. Achir, A. Abdelli, L. Mokdad, and J. Benothman, "Service discovery and selection in IoT: A survey and a taxonomy," J. Netw. Comput. Appl., vol. 200, no. December 2021, p. 103331, 2022.

[38] A. Alhowaide, I. Alsmadi, and J. Tang, "Ensemble Detection Model for IoT IDS," Internet of Things, no. March, p. 100435, 2021.

[39] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130–165150, 2020.

[40] A. Churcher et al., "An experimental analysis of attack classification using machine learning in IoT networks," Sensors (Switzerland), vol. 21, no. 2, pp. 1–32, 2021.

[41] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," Electron., vol. 8, no. 11, 2019.

[42] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," Comput. Commun., vol. 166, no. December 2020, pp. 110–124, 2021.

[43] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets," IEEE Internet Things J., no. May, 2021.