Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(10): 10550–10580.
DOI: 10.3934/mbe.2022493
Received: 21 May 2022
Revised: 30 June 2022
Accepted: 04 July 2022
Published: 25 July 2022

Research article

Ensemble learning-based IDS for sensors telemetry data in IoT networks

Naila Naz[1], Muazzam A Khan[1], Suliman A. Alsuhibany[2,*], Muhammad Diyan[3], Zhiyuan Tan[4], Muhammad Almas Khan[1] and Jawad Ahmad[4]

[1] Department of Computer Science, Quaid-i-Azam University, Islamabad, Pakistan
[2] Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[3] School of Physics and Astronomy, University of Glasgow, United Kingdom
[4] School of Computing, Edinburgh Napier University, United Kingdom

* **Correspondence:** Email: salsuhibany@qu.edu.sa.

Abstract: The Internet of Things (IoT) is a paradigm that connects a range of physical smart devices to provide ubiquitous services to individuals and automate their daily tasks. IoT devices collect data from the surrounding environment and communicate with other devices using different communication protocols such as CoAP, MQTT, DDS, etc. Study shows that these protocols are vulnerable to attack and prove a significant threat to IoT telemetry data. Within a network, IoT devices are interdependent, and the behaviour of one device depends on the data coming from another device. An intruder exploits vulnerabilities of a device's interdependent feature and can alter the telemetry data to indirectly control the behaviour of other dependent devices in a network. Therefore, securing IoT devices have become a significant concern in IoT networks. The research community often proposes intrusion Detection Systems (IDS) using different techniques. One of the most adopted techniques is machine learning (ML) based intrusion detection. This study suggests a stacking-based ensemble model makes IoT devices more intelligent for detecting unusual behaviour in IoT networks. The TON-IoT (2020) dataset is used to assess the effectiveness of the proposed model. The proposed model achieves significant improvements in accuracy and other evaluation measures in binary and multi-class classification scenarios for most of the sensors compared to traditional ML algorithms and other ensemble techniques.

Keywords: ensemble learning; intrusion detection; IoT; sensors security; ToN-IoT; bagging

1. Introduction

Internet of Things (IoT) is influencing our lifestyle from the way we act to the way we behave. It is a collaborative network with connected smart IoT devices. These devices are equipped with sensors,

actuators, storage, computational, and communication capabilities to gather sensitive telemetry data from remote locations and share this data to receiving systems for analysis [1, 2].

In an IoT network, the devices are interconnected and communicate with each other through some protocols to provide a service. Thus, the behaviour or decision of one device depends on the data (telemetry) coming from another device, and that implicit relationship is device interdependence.

Today, IoT applications are surprisingly increasing in all fields to automate the traditional systems in the industry, retail, home and building, health, agriculture, etc. IoT smart objects (sensors and actuators) provide intelligence to our standard devices by blending computing and network capabilities [3,4]. Due to which the smart appliances can be operated automatically, e.g., a smart fridge can monitor the temperature condition and set it according to a threshold value and the state (open/close) of the smart door based on the detection of movement [5].

As per [6–8] currently, the devices connected to the internet are estimated to be more than 25 billion and up to 50 billion connected devices expected till 2022 [9]. With the exponential (dramatic) growth of IoT applications, IoT devices have become a smart object for attackers in attaining their goal of intrusion. For example, in 2015 Ukrainian government reported the outage of power service due to cyber security attacks in which approximately 225,000 customers were affected. An attacker penetrated the monitoring system of IoT devices due to a poor security mechanism which resulted in a power blackout [10]. The requirements for IoT applications vary as per industry. Thus, multi-layered network security system also takes into account the security challenges across each layer [11].

Various available security solutions such as encryption, firewall, and intrusion detection cannot be directly installed on IoT applications. Because IoT devices have their unique features such as interdependence, constraint (energy [12–15] , storage etc.,), diversity, intimacy, etc. [16]. These features also affect the security and privacy [17] of IoT networks. For example, in an automated smart light system, an intelligent device (sensor) senses the light level and compares it with the threshold value. If the light level is below than threshold, it tries to balance the light level and automatically turns on the smart bulb. The smart bulb activation depends on the light level capturing device data in this scenario. Similarly, another example (illustrated in Figure 1) of an intelligent room consists of multiple IoT devices, i.e., a smart plug, smart window, thermometer, and air-conditioner (AC). When the thermometer detects that the temperature inside the room has risen over a threshold and, simultaneously, the smart plug detects that the AC is turned off, the window will automatically open to stabilize the room's temperature. Here, if an intruder wants access to the room, he does not need to attack the smart widow directly. He can create a physical security breach by gaining access to a smart plug and using it to switch off AC that raises the temperature in a room and cause to open the window automatically [18].

Such independence among IoT devices in IoT applications opens a path for an intruder to indirectly control the targeted device by altering the data coming from another device. Therefore, to secure the sensitive telemetry data of IoT devices and ensure the secure interdependency among IoT devices, a system such as an Intrusion detection system (IDS) that specifically meets IoT applications' requirements is the need of the day.

1.1. Intrusion detection system

IDS is a software or a monitoring device that keeps track of the network to secure and flags the administrator if any suspicious activity has been detected. It can detect the malicious attacks data and normal data which might not be identified with traditional security mechanisms, i.e., firewall [19].

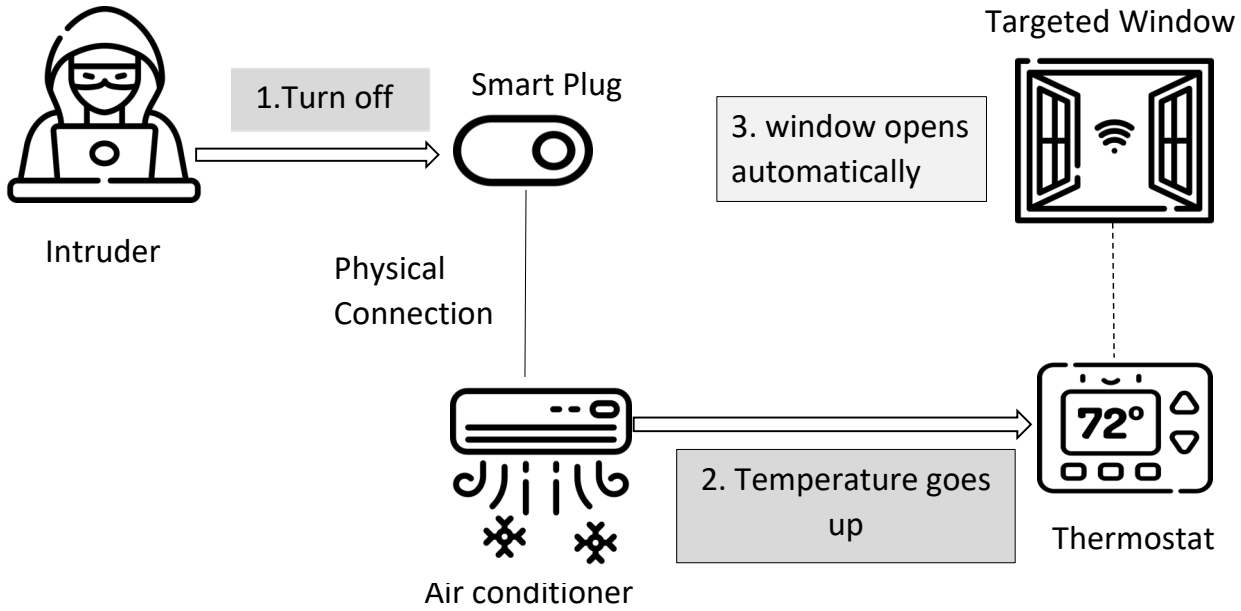

Figure 1. Intrusion due to device interdependence.

In literature, [19, 20] multiple dimensions, such as the source of information, identification method, detection method, and learning approach, are used to classify IDSs. Figure 2 illustrate the classification of IDS.

1.2. Information source based IDS

This type of IDS is mainly divided into Host-based (HIDS) and network-based (NIDS) IDS. HIDS are usually installed on localhost (i.e., IoT device or system) to secure it from malicious activities. It monitors the host's file system, system calls, network events and informs the system when the intruder performs any suspicious activity. HIDS is installed on a single computer or device and is only capable of detecting attacks on the system on which it is installed [21]. Whereas NIDS keeps an eye on the data related to a network, such as IP addresses, network packets, protocols [22], traffic volume, etc., to detect the intrusion [23]. To achieve real-time intrusion detection, HIDS and NIDS are also used.

1.3. Detection based IDS

In general, Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS) are primary intrusion detection techniques.

1.3.1. Signature based IDS (SIDS)

SIDS (misuse detection) is a database based mechanism for storing known attack signatures. The signatures are the patterns of attacks extracted from network packets' features. SIDS monitors the packets of network traffic and compares them with the previously stored signatures in the database. If the pattern matches with any existing database signature, then SIDS generates an alarm. It only detects the known attacks and is unable to identify the zero-day (unknown) attacks [24]. Thus, the

system's security requires the continuous identification of new signatures and updating the database with expert inception. The performance of a SIDS methodology is contingent upon the rules deployed, and modifications to the rules may affect the IDS's performance. The most common example of SIDS is the Snort tool [25].

1.3.2. Anomaly based IDS

An AIDS is referred to as behaviour-based detection and deals with unknown attacks. It observes the network and system behaviour and establishes a baseline from typical behavioural patterns using a variety of ways. Any considerable variation from the baseline of an inspected pattern is labelled as an anomaly [26] .

1.4. IDS placement schemes

In an IoT network, multiple devices communicate and share data through a medium like a router. There are a few techniques for deploying intelligent IDS to secure these data-sharing nodes.

1.4.1. Distributed IDS placement

Intrusion detection systems are deployed in every physical resource-constrained object or node in distributed placement. This aspect of resource-constrained devices must be considered when developing IDS. [27] address this issue and proposed lightweight IDS. The nodes in a distributed deployment may also be responsible for keeping an eye on their neighbours. Moreover, distributed IDS comprise several IDS dispersed around a vast IoT ecosystem and linked to one another or a central server.

1.4.2. Centralized IDS placement

The intrusion detection system is deployed in a central device, such as a router or dedicated host, in the centralized IDS location. It is also known as NIDS. The network router/dedicated host receives all the data collected by IoT devices and sends it to the internet. Resultantly [28], an IDS were installed in a router or dedicated host that can examine all traffic flowing among the nodes and the Internet.

1.4.3. Hybrid IDS placement

HIDS placement blends centralized and distributed placement techniques to take the edge of their advantages while avoiding their disadvantages.

1.5. Validation strategies of IDS

Validation ensures that the proposed or developed model acts accurately within the scope and determines objectives. There are many validation strategies of IDS. [29] defined the primary strategy followed by the researcher in literature for validating the proposed techniques are:

- Simulation: includes the methods for simulating some IoT scenarios using software like MATLAB, IoTIFY, etc.
- Empirical: Empirical validation is how a model's accuracy can be verified through systematic experimental data collected from an operational setting or context.

- Hypothetical: having an uncertain relationship to genuine phenomena and level of reality
- Theoretical: use of formal or precise theoretical reasons to validate results

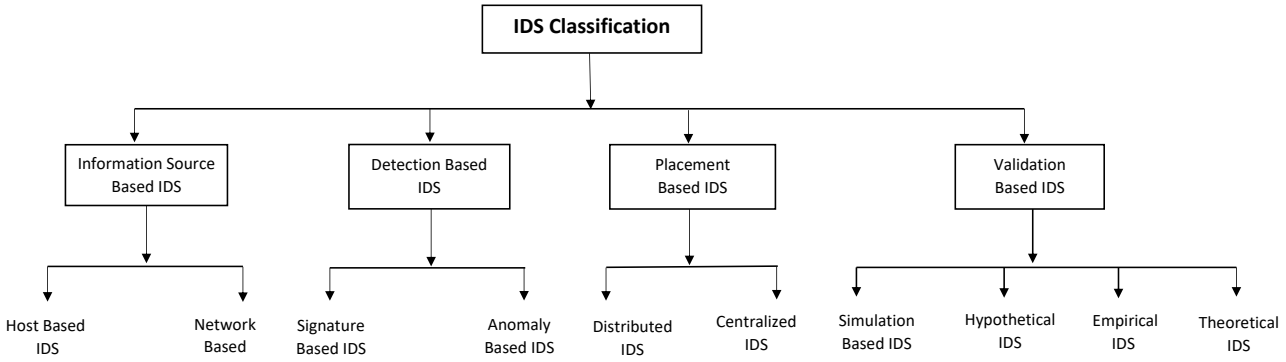

Figure 2. Categories of IDS.

The dependence feature among IoT devices allows intruders to alter the data into network by compromising another dependent IoT device. Without sufficient security and intelligence in smart objects, attackers manipulate the telemetry data with the help of MItM assault [30] and code injection assaults. The intruder attempts to send messages between two dependent devices (nodes) in a network and control the behaviour of other devices.

Therefore, The features (interdependence, constrained, etc.) of IoT devices (sensors) and new diverse attacks on telemetry data motivate us to upgrade IDS which meets the specific requirements of IoT devices and detect tempered telemetry data during communication between objects. It is found from the literature that ML has shown efficiency in different application areas, including intrusion detection systems for IoT [31–34]. Other solutions such as deep learning (DL) and artificial neural network (ANN) are also helpful, but they need more computation power and resources than ML. Deep learning solutions also requires large amount of data for training. In a survey [35] authors also highlighted the security and privacy issues of DL in recent years with respect to two types of attacks. Some IoT devices, i.e., sensors, are resource-constrained and require a lightweight solution based on computation power and resources [36]. Therefore, ML classifiers have been selected for designing an IDS for telemetry data security. But a single ML classifier's detection accuracy is not always good. For example, single LR does not perform well for non-linear data, and classification done by DT tends to majority class which causes overfitting. To improve detection rates while lowering training and generalization errors, complex strategies should be used, which drive us to apply ensemble methods to detect attacks in telemetry data. Consequently, a stacking-based ensemble model has been proposed based on ML classifiers to improve the detection rate in this study. In the proposed model, Linear Discriminant Analysis (LDA), Naive Bayes (NB), Random Forest (RF), and Linear SVC have been ensembles as a base model, and Logistic Regression (LR) is selected as a meta-model. This model is deployed on the Ton_IoT telemetry dataset for binary and multiclass classification. The model achieves significant improvement in accuracy. The proposed model handle the variance and biasness issues of single classifiers. The selected meta model have low variance which solves overfitting problem. The proposed model detect the attack and also identify the type of attack.

1.6. Contribution of study

The novel contributions of this study are:

- A stacking based ensemble intrusion detection (ID) model has been proposed for telemetry data protection
- The proposed model has been evaluated for both binary and multiclass classification scenarios on single IoT device data and merged IoT devices data.
- Cross comparison of homogeneous and heterogeneous ensemble model has been performed for telemetry data

1.7. Structure of paper

The remainder of the paper is organized as follows: Section II continues the study of existing ML and EL approaches for IDS. In contrast, Section III provides an overview of various supervised ML techniques. Section-IV deals with our research methodology's flow, followed by a detailed overview of the dataset and experimental setting. Whereas section-V discusses the experimental outcomes. Section VI concludes the study with a few concluding points.

2. Literature review

This section focuses on the prevailing IDS techniques based on ML and Ensemble Learning (EL) used by the research community to identify intrusions across various IoT applications. In the digital era of the smart environment, various application fields, including industrial processes, public safety, energy consumption, home automation, environmental monitoring, healthcare, etc., could benefit significantly from IoT systems [37]. Different sensors are deployed to control or execute the operations using multiple techniques [38, 39] in a smart environment. Smart things become more effective when IoT systems and smart environments are integrated. However, IoT systems are subject to several security vulnerabilities because of interconnectivity. The attack against any smart environment such as household appliances could harm the end-user (family). [40] describe that any IoT system's security problems can be divided into four categories: authentication and physical threats, confidentiality hazards, data integrity/ aggregation [41] issues, and privacy concerns. Most of the studies [22, 42, 43] provide efficient and secure data aggregation solutions for wireless sensor networks.

A smart home contains many smart appliances, such as door locks, power switches, light bulbs, smoke alarms, etc. A variety of network-based attacks can target these devices. The study in [44] describes how IoT devices are susceptible to attacks with an example. The author performed a test with three standard smart home devices: nest smoke-alarm, Phillips hue light-bulb, and the Belkin WeMo power switch. The author observed the network activities of these IoT devices and illustrated the ease with which security and privacy can be compromised. One of the results of this investigation was finding flaws in the request-response message passing between the bridge and the Philips hue light bulb app. The attacker can easily capture the user name and bridge IP address. They used a primary python language to deliver HTTP-PUT requests and gain complete control of the bridge. As a result, the author provided a solution for this problem by restricting access at the network level, i.e., the cloud service provider offers security as an overlay service that does not affect connected home devices. As a result, the proposed security solution, namely home network access control rules, is a feasible way

for ensuring privacy and security [45] in smart homes.

Another fascinating discussion about the smart home environment was done by [46]. He addressed privacy challenges and ramifications with connected devices. The article covers the data collected by smart home users. This discussion led to the conclusion that malware management was a significant research gap.

A detection model based on deep learning techniques was proposed by [47]. The experiments were conducted by collecting data from the gas pipeline system's Remote Telemetry Unit (RTU). The vital information was then extracted from this data, valid for the experiments. The proposed model outperformed other detection algorithms regarding detection rate, precision, and false-positive rate. Its results for the identification of zero-day attacks were good. For accurately designing and evaluating IoT/IIoT defense systems, [48] generate a new telemetry-based dataset by using heterogeneous devices. Diverse cyber-attacks were launched in a dataset. The performance of attack classification over IoT telemetry data was investigated using a variety of ML and deep learning algorithms. As a result, CART mainly obtains high accuracy.

A study based on supervised IDS for a smart home (consisting of 8 IoT devices) IoT network is conducted by [49]. The suggested IDS architecture comprises three levels to identify prevalent network-based cyber-attacks. The proposed IDS detects malicious activity and determines its type. Three tests were run with nine different classifiers over different layers. The result shows that the J48 classifier had the best performance in terms of f-measure in all three experiments, with 98%, 96.2%, and 90%. The system's flaw was that it required the integration of all three layers to identify an intrusion. The entire system will be affected in case of failure in any layer.

[50] used an SVM-based classifier including three kernel functions to create a lightweight IDS. The author solely considered the forms of assault that will impact the traffic intensity in that study. The proposed IDS only evaluates a packet arrival rate attribute according to the given results. The experiments were performed by using MATLAB version 2018b. Several functions such as linear, polynomial, and radial-basis evaluated the classifier. Because there were fewer inputs rather than significant inputs, SVM processing time and complexity were lowered. Traditional evaluation measures, i.e., accuracy, TPR, FPR, and FDR, were selected to evaluate the model. The inability to recognize intrusions without a corresponding influence on traffic intensity was a weakness of this proposed technique.

In another ML approach, [51] worked on KDD Cup 99 dataset for classification. The 22 types of attack were categorized into four classes in the dataset, namely, DDoS, Prob, U2R, R2L. They were developed Sequential Minimal Optimization (SMO), NB, Bayes Net (bN), Multilayer Perception (MLP), RF algorithms for the dataset. The results of the algorithms were compared with one another in terms of false rate, precision, recall, f-measure, and accuracy rate value. The experiments were performed on weka software for the classification of data.

Another comprehensive study conducted by [52] investigated 14 different ML algorithms that were applied for IDS in diverse situations. The experiments were conducted on the KDD99 Cup dataset using the MLA approach. This suggested model showed that algorithms such as RF, ANN, and decision tree provide superior development than others for identifying attacks. It also reveals that the area of application and the algorithm influenced the FPR, detection rate, and accuracy.

Furthermore, A comparative evaluation of commonly used ML algorithms such that LR, MNB, G-NB, B-NB, KNN, DT, AdaBoost, RF, MLP, and GB was tested on the UNSW-NB15 dataset in [53]. The study reveals that The RF classifier exceeds the other classifiers in terms of accuracy, positively

predicted value, and f-measure, with 87, 98, and 84%, respectively.

Ensemble approaches have gained considerable attention in recent years in intrusion detection. In a study, [54] an XGBoost ensemble model was proposed to examine the botnet attacks against three protocols DNS, HTTP, and MQTT. An author claims that its proposed model is built on tree boosting ML methods, which smooth out the "bias-variance" trade-off. Precision, accuracy, recall, F-1 measure, and support evaluated the proposed IDS on the KDDCup99 dataset. The proposed approach attained a precision of 99.95%. However, the KDDCup99 dataset was obtained from traditional networks, and it did not include sensor telemetry data. Therefore, it was unable to serve as an adequate IoT benchmark dataset.

Three standard ensemble techniques bagging, boosting, and stacking, were explored in study [55] for a NIDS. The selected ML classifiers were NB, ANN, J48, and REPTree, and each classifier was employed as a based model in the bagging and boosting technique. While in the stacking model, REPTree was used as the base learner and an ANN as meta learner. The stacking model achieved the maximum accuracy of 87.92% than bagging and boosting. All tests were conducted on the UNSW-NB15 data set using the Weka Data Mining tool. The study results depicted that J48 and REPTree outperformed the NB and ANN by achieving the highest accuracy rates.

Another study [56] was presented an Ada Boost ensemble IDS for mitigating botnet attacks against the DNS, HTTP, and MQTT protocols used in IoT networks. An Ada boost model was implemented via DT, NB, and ANN. A new statistical flow feature technique was presented to generate new features from the protocols and used to evaluate them in malicious activity detection. The evaluation was conducted on UNSW-NB15 and NIMS botnet datasets using ensemble techniques in accuracy, DR, and FPR. The new method outperformed three current methods in DR and FPR when compared to SVM, Markov chain (MC), and Bayesian network (BN).

An AIDS model for IIoT networks with two phases was presented in [57]. In the first phase, SVM and NB classifiers were an ensemble, and Kfc-validation was used to build the train and test datasets. The second step used the ANN and RF results as input for model classification. The best classifiers were determined based on the results. The models were validated using three publicly available data sets: WUSTL IIOT-2018, N BaIoT, and Bot-IoT. The selected evaluation metrics Precision, recall, and accuracy attains up to 98% performance on all datasets.

Author [58] introduced an architecture ELNIDS to detect routing attacks against IPv6 Routing Protocol based on the Signature technique. Four ensemble models, boosted Tree, bagged Tree, subspace discriminant tree, and RUSBoosted tree, were implemented for this study. The dataset RPL-NIDDS17 was developed by using the NetSim tool. The expanded dataset contained traces of routing attacks on the RPL protocol. The simulation was carried on Matlab 2017b. Ensemble of Boosted trees achieved the highest accuracy of 94.5% then others based on provided results.

To determine the effectiveness of ensemble learning for NIDS [59] constructed both homogeneous and heterogeneous ensemble techniques. Rf was selected inhomogeneous method, and NB, KNN, RIPPER, DT classifiers were ensemble for the heterogeneous ensemble model. The binary and multi-class classification was performed on the UNSW_NB15 dataset. Reported results depicted that LSTM achieves accuracy up to 80% for binary type and 72% for multi-classification.In contrast, homogeneous model RF conducted high accuracy 98% for binary and 87.4% for multi-classification then LSTM. The results heterogeneous model was comparably same as the homogeneous model.

Authors in [60] proposed an ensemble model based on a Bayesian network and RF as base classifier

along with vote and RandomCommitte as meta classifiers. The proposed model was evacuated on the KDDcup99 dataset with 10-fold cross-validation. The model was evaluated regarding the accuracy and ROC curves as evaluation metrics. Their proposed model achieved better results (0.99%) in terms of AUC than a single Bayesian network and random tree for all attacks(Probe, DOS, U2R, R2L) classification.

To develop AIDS, a three-tier paradigm was proposed in [61]. The tries include feature selection, modeling of the classifier, and validation. The first feature selection tier uses a hybrid technique based on three evolutionary search algorithms to select important features. The model's rotation tree and banging were trained for classification in the second tier. In the last stage, the validation was performed with 10fcv. The study was conducted on datasets UNSW-NB15 and NSL-KDD datasets. Precision, accuracy, FPR, and recall were selected for evaluation. The model achieves accuracy 85.8%, sensitivity 86.8%, and detection rate 88.0% for the NSL-KDD dataset.

To increase the efficiency of AIDS in IoT networks [62] employed an RF classifier. Parameter tuning was performed with different sizes for ensemble trees in their work. The proposed methodology evaluated FAR, and accuracy for publicly available datasets named UNSW-NB15, GPRS, and NSL-KDD. The authors use statistical tests such as the Friedman and Nemenyi tests. The results indicate that the suggested model works better on the NSL-KDD dataset than on the GPRS dataset. It is demonstrated that the proposed strategy does not produce satisfactory results when dealing with sensor-related data. An author [48] tested several ML algorithms on proposed ToN-IoT dataset. Random forest is selected to test the performance of EL for telemetry data of IoT devices.

The Table 1 represents the comparison of literature based on Ensemble Learning techniques.

2.1. Limitations:

The following limitations have been extracted from the literature:

- Old datasets were used concerning telemetry data.
- The selected datasets do not have specific characteristics of IoT and IIoT applications
- Most of the study focused on network-based security and not focused on sensors telemetry data security
- The evaluation of ensemble techniques to secure the sensors telemetry data is not available in the literature

3. Supervised models for IDS

From the literature, it is found that researchers have used a variety of machine learning algorithms to detect intrusions. It is difficult and risky to share network data with the research community since it may contain sensitive communications or personal information. Due to that, the absence of training data makes it challenging to use ML for intrusion detection. This section presents the primary mathematics behind some classical machine learning models used as members of our proposed ensemble-based model. When applying multiple models to different problems, a comprehensive understanding of what happens under the hood is often unneeded. Understanding the fundamentals of each algorithm is beneficial when choosing a model and adjusting parameters to improve model performance.

Table 1. Summery of literature review.

Reference No	Year	Ensemble Technique		Models	Dataset	Classification		IoT Telemetry data of diverse Sensors	Classification of Diverse new attacks	Limitation
	H1	H2				Binary	Multi			
[49]	2019	No	No	Naïve bayes, J48, Simple Logistic , zero R, SVM	Self-testbed	Yes	Yes	No	No	– Requires integration of 3 layers for detection – Failure in one layer effect the entire system
[50]	2019	No	No	SVM with 3 kernel function	CICIDS-2017	Yes	No	No	No	– Only network packets are used for detection – unable to recognize intrusions without a corresponding influence on traffic intensity
[54]	2021	Yes	No	XGBoost	KDDCup99	Yes	No	No	No	Model is evaluated for non-sensors data.
[55]	2017	Yes	Yes	Naive Bayes, ANN J48, REPTree	UNSW-NB15	Yes	No	No	No	Low performance of single classifiers in terms of accuracy
[63]	2019	No	Yes	Bagged RF voting with Nave Bayes KNN, RIPPER DT classifiers, LSTM Bayesian tree	UNSW-NB15	Yes	Yes	No	Yes	– Don't classify some new attacks – The low detection rate of DL for multi-classification
[60]	2016	No	Yes	Random forest Voting	KDDCup99	No	Yes	No	No	Low accuracy for U2R attack detection
[61]	2019	Yes	No	rotation forest bagging	NSL-KDD UNSW-NB15	No	Yes	No	No	Low detection rate for R2L and U2R
[62]	2017	Yes	No	Random forest	UNSW-NB15 GPRS NSL-KDD	Yes	No	No	No	Model performance on GPRS (Sensors-related data) is not satisfactory
[48]	2020	Yes	No	ML Models, Random forest LSTM Stacking with SVM	ToN-IoT	Yes	No©	Yes	Yes	Single ML classifier performance is low for sensors data
This Study	2022	Yes	Yes	Naïve Bayes Logistic Regression Random forest, LDA Bagging with DT	ToN-IoT	Yes	Yes	Yes	Yes	—

H1= Homogeneous Ensemble Technique, H2= Heterogeneous Ensemble Technique, No©= Multi classification is not available for single sensors

3.1. ML classifiers for IDS

3.1.1. Logistic Regression (LR)

LR is a classification algorithm that deals with the discrete set of classes. LR estimates a probability (chance that a test sample belongs to a particular class) value of the test sample that may be mapped to two or more discrete types using the logistic sigmoid function [64]. Sigmoid function mapped absolute values into another value between 0 and 1. During classification, the model predicts that the instances belong to a negative class if the estimated probability is below 50%; otherwise, illustrations belong to a positive class [65]. The mathematical representation is shown as:

$$g(z) = \frac{(1)}{(1 + e^{-z})} \tag{1}$$

where z is the function's input, e is the natural log's base, and g(z) is the output, a probability estimated between 0 and 1. The cost function for logistic regression is the average log loss overall training cases. The cost function is written as follows:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i))} log(h(z(\theta)^{(i)}) + (1 - y^{(i)}) log(1 - h(z(\theta)^{(i)})) \tag{2}$$

where m is the set of possible training samples, y^i is the actual label for the i^{th} training sample, and $h(z(\theta)^{(i)})$ is the model's prediction for the i^{th} training example.
LR performance is better for the simple datasets with linearly separable classes. The LR underperforms to increasingly complicated datasets, but regularization methods can avoid this limitation.

3.1.2. Linear support vector machine

SVM stands for Support Vector Machine and is a supervised learning technology that deals with classification and regression problems. However, in ML, it is usually turned to account for classification problems. The number of features determines the number of dimension plans in SVM, and the value of an attribute is defined as the value of a set of coordinates. In n-dimensional space, the data elements are marked as a point. Then classification is accomplished by locating the optimal hyperplane (best decision boundary) that distinctly classifies the data points. Linear SVM classifies the linearly separable data points by using a straight line [66]. Multiple hyperplanes can be created via kernel such as a linear, polynomial, Gaussian Radial Basis Function (RBF), etc. The mathematical representation of a single hyperplane is:

$$w.x + b = 0 \tag{3}$$

The equation is driven from two-dimensional vectors, where w is represented as the weight vector, x is the input vector, and b is the bias. The hyperplane is used to predict by using a hypothesis function that is defined as:

$$h(xi) = \begin{cases} 1 & \text{if } wx + b \geq 0 \\ -1 & \text{if } wx + b < 0 \end{cases} \tag{4}$$

The data elements above or on the hyperplane are classified as positive class +1, and the data points below the hyperplane are classified as negative class -1.

3.1.3. Naive bayes

The Naive Bayes algorithm is a supervised classification technique based on the Bayes theorem. It does not work as a rule-based classification. It works on probability theory for classification. So NB is a simplified Bayesian probability model that assumes that all attributes' class values are independent. It indicates that the probability calculated for one feature does not affect the other [67]. The three main types of NB are Gaussian, Multinomial and Bernoulli. NB combines prior probability and conditional probabilities into a single formula, which is mathematically expressed as follows:

$$P(L|M) = \frac{P(M|L).P(L)}{P(M)} \tag{5}$$

where P(L) and P(M) are the prior probabilities of the targeted class and independent variable, respectively. However, P(L/M) is the posterior probability of targeted class L for independent variable M. The probability of P(M/L) is calculated using the following formula:

$$P(M|L) = P(m_1|L) * P(m_2|L) * P(m_3|L)..... * P(m_n|L) \tag{6}$$

P(M/L) represents the probability calculated for each instance m for targeted class L. Whereas, conditional probability is mathematically expressed as:

$$P(M) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{7}$$

where μ and σ represents the mean and variance of a value for a given class. The values of i and j can be 1,2,3...n. The Naive Bayes technique is a popular one that is frequently used by the research community for classification, and feature reduction [68, 69]. Assumption of all mutually independent attributes and dependence on all attributes being categorical are the limitations of Naive Bayes.

3.1.4. Random Forest (RF)

RF or random decision forests is an EL method, in which a considerable number of de-correlated trees are built and then averaged [70]. RF generates the forest of decision tree from randomly split dataset into samples. An individual decision tree is created for each attribute depending upon an independent random sample [71]. To classify test data, predictions from each tree are obtained, and finally, the class is assigned to test data by majority voting or averaging technique [48].

3.1.5. Linear Discriminant Analysis (LDA)

At the stage of data pre-processing, LDA is a well-known linear algorithm that is widely used as a dimensionality reduction [72] and pattern classification method. However, in this study, LDA is used as a classification approach to developing an IDS. LDA has several advantages, i.e., easy to apply, efficient, and lower computation cost. Therefore, it makes LDA a good choice for creating an IDS [73]. The main steps of LDA are listed as follows:

- Estimate d-dimensional mean vectors for the distinct classes from the dataset. The mathematical representation of the mean vector of class metrics is:

$$\mu_{Cls_i} = [m_{a_1}, m_{a_2}, m_{a_3}, ..., m_{a_n}] \tag{8}$$

  where m_a_i represents the mean of the i^{th} attribute of the class matrix.

- Compute covariance matrix for multivariate features from the training data from the equation as:

$$Cls_i^{cov} = \frac{Cls_i^{mc^T} * Cls_i^{mc}}{n_i} \tag{9}$$

where Cls^{mc^T} represent the mean corrected class matrix. The number of rows are represented by n. whereas i can be 1 to n.

- Based on the Bayes theorem and the probability of each class, the probability of output class (attacked/normal) is estimated for a given observation.
- Make final prediction by using discriminate function. The discriminant function expresses as:

$$f_k(x) = x * \frac{\mu_k}{\sum} - \frac{\mu_k^2}{2 * \sum} + log(P_k) \tag{10}$$

where $fk^{(x)}$ is the discriminate function for class k given observation x, co-variance matrix is as sum sign. μ sign shows the mean and estimated probability is denoted as P.

3.1.6. Decision Tree (DT)

DT is one of the most widely used for classification and intrusion detection. DT consists of three basic components, namely decision node (identifies test attribute), branch (possible choice based on the test attribute value), and a leaf node (the class that the instance is a member of) [74]. The data set is learned and modeled first, and then a tree is formed in the DT algorithm. When test data is given to DT, it will be classified based on the prior dataset's classification procedure. A test is performed for classification using the test attribute value and a decision procedure (denoted by root node). The class (normal, attack) is assigned to the test data when the leaf node is reached. DT better perform for huge datasets [26]. DT has the advantages of better detection performance, generalization accuracy, etc.

3.2. Ensemble Learning (EL)

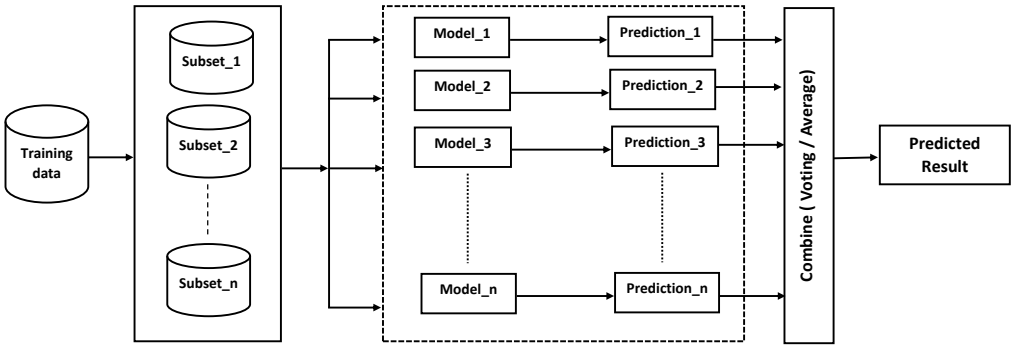

Figure 3. Ensemble learning technique.

Each ML algorithm has various constraints, like a low bias and a slight variance. EL address the limitations of standalone machine learning techniques. The concept of the EL technique was introduced in 1979 by [75] to improve the performance of independent ML algorithms. An ML paradigm

merges a diverse range of models (weak learners) to produce one optimal predictive model that gives more accurate results than a single model. The modular structure of EL avoids the overfitting issues associated with high variance. The algorithms combined in the EL technique should be selected based upon their computational cost to achieve better performance.

The visual representation of EL is given in Figure 3. In the ensemble learning process, firstly, the dataset is split into training and testing data. Then training data divides into multiple subsets using various techniques (i.e., with-replacement, without replacement). The subsets are given to selected models to train them in the next step. Then the test data(unseen) is provided as input to these trained models for the prediction. Finally, the predictions made by each model (base learner) are combined with a majority vote or averaging methodology [76]. Mathematically it is expressed by the equation:

$$E_{n,m} = \frac{1}{N} \sum_{n=1}^{N} P^n n = 1, 2, 3.....N \tag{11}$$

where $E_{n,m}$ is the output of the ensemble model of the N machine learning classifier. P is the prediction of each classifier.

3.2.1. Types of EL

EL is categorized into two main types: Homogeneous EL and heterogeneous EL. In the homogeneous EL approach, the same kind of base classifiers (weak learners) is used to train on a different subset of data. The result of each classifier is aggregated to improve the precision. The homogeneous EL uses the same feature selection method for all training data. This type of EL is suitable for large datasets. Bagging and boosting are the most common type of homogeneous EL. Whereas, in heterogeneous EL, different base classifiers are used to train on the same data. This technique works well for small datasets. The feature selection technique for each base learner is different for the same data. An example of heterogeneous EL is stacking.

3.2.2. Bagging

The term “bagging” is an acronym for ”bootstrap aggregation.” It is a parallel ensemble process that reduces the variance by averaging multiple trees and increasing the prediction accuracy [77]. Random forest is considered the advanced version of bagging. The following lines summarize how to bootstrap aggregation works:

- The random sample of data are selected from the training dataset with replacement technique (which means the individual instance can be chosen multiple times)
- In the next step, the base models are trained individually in parallel.
- The final output is obtained by combining the predictions of each base model with an appropriate technique (i.e., majority vote, averaging, weighted average) based on the type of task(classification, regression)

Table 2. Features in each file.

File Name	Features
IoT_Fridge	Ts, date, time, fridge_temperature , temp_condition, label, type
IoT_Garage	Ts, date, time, door_state, s_phone signal, label, type
IoT_GPS_Tracker	Ts, date, time, latitude, longitude, label, type
IoT_modebus	Ts, date, time, FC1_read_input_register, FC2_read_discrete value, FC3_read_holding_register, FC1_read_coil, label, type
IoT_light_motion	Ts, date, time, motion_status, light_status, label, type
IoT_thermostate	Ts, date, time, current_temp, Thermostat_status, label, type
IoT_weather	Ts, date, time, Temperature, Pressure, Humidity, label, type

Table 3. Statistics of TON_IoT dataset.

File No	Total Record	Normal Instances	Attack Instances							Total Attack	% of attack data	% of Normal data
			scanning	DDOS	Ransomware	Backdoor	Injection	XSS	Password			
1	59944	35000	Nil	5000	2902	5000	5000	2042	5000	24944	41	58
2	59587	35000	529	5000	2902	5000	5000	1156	5000	24587	41	58
3	58960	35000	550	5000	2833	5000	5000	577	5000	23960	40	59
4	51106	35000	529	Nil	Nil	5000	5000	577	5000	16106	31	68
5	59488	35000	1775	5000	2264	5000	5000	449	5000	24488	41	58
6	52774	35000	61	Nil	2264	5000	5000	449	5000	17774	33	66
7	59260	35000	529	5000	2865	5000	5000	866	5000	24260	41	59
8	401119	245000	3973	25000	16030	35000	35000	6116	35000	156119	38	61

1.IoT-Fridge File 2. IoT_Garage 3. IoT_GPS_Tracker 4. IoT_Modbus 5. IoT_Light_Motion 6. IoT_Thermostat 7. IoT_weather 8. Integrated_File

4. Proposed methodology

The workflow of our methodology is illustrated at Figure 4.

4.1. Dataset: TON_IoT

The available datasets used for intrusion detection have some limitations. These datasets are outdated in telemetry data and do not have specific IoT and IIoT applications characteristics. The frequently used datasets (KDD-CUP99, UNSWNB (2015), CIC-IDS (2017), BOT IOT (2018), and CIC-IDS (2017)) in the IDS domain for IoT are compared based on several parameters. Study [48] shows that these datasets did not have various sensors following with diverse new attacks, which could result in a lake of telemetry data. Resultantly, the TON_IoT dataset was chosen for the evaluation of EL approaches. The most recent effort to create an IDS dataset was made by UNSW (Australia) in 2020,

and developed telemetry data name as TON_IoT. It is a publicly available dataset [78]. The name TON shows that this dataset was collected from different Telemetry data of IoT applications, Operating system logs, and Network traffics of IoT networks. The testbed based on realistic representation of medium-scale networks has been set up to collect data from cyber range and IoT labs. The testbed was built using three layers (edge, fog, and cloud) where multiple IoT applications and network elements interacted with each other. The dataset comprises seven CSV files that have sensors telemetry data. The features of each file are presented in Table 4.

Type and label features are common in each file. The Label feature in each file comprises 0 (indicate normal instance) and 1 (indicate the attack instance) values. The label feature performs binary classification, whereas the type feature contains categorical values (indicating the attack type) used for multiclass classification. The dataset contains nine diverse attacks named scanning, password cracking, data injection, XSS, backdoor, and ransomware. Table 3 represents the detail description of dataset. The seven files are combined in one CSV file that holds 401119 instances. A total of 245000 instances were normal, and 156119 were attacked instances. The dataset is well balanced, with normal records accounting for 61% of the total and attacked records accounting for 38%.

Table 4. Features in each file.

File Name	Features
IoT_Fridge	Ts, date, time, fridge_temperature , temp_condition, label, type
IoT_Garage	Ts, date, time, door_state, s_phone signal, label, type
IoT_GPS_Tracker	Ts, date, time, latitude, longitude, label, type
IoT_modebus	Ts, date, time, FC1_read_input_register, FC2_read_discrete value, FC3_read_holding_register, FC1_read_coil, label, type
IoT_light_motion	Ts, date, time, motion_status, light_status, label, type
IoT_thermostate	Ts, date, time, current_temp, Thermostat_status, label, type
IoT_weather	Ts, date, time, Temperature, Pressure, Humidity, label, type

4.2. Pre-processing and Training of dataset

To evaluate the model on the Ton_IoT dataset, it is necessary to pre-process it before using it. Some frequently used methods (cleaning, encoding, and scaling) have been employed to perform pre-processing on ToN_IoT. The detail in each method is given below:

4.2.1. Method for feature selection

To select the essential features from the pole of features in the dataset is the most vital step. The dataset chosen contains seven files which comprise up to six parts. The integrated file (combined seven files) holds 22 features vector in total. The chi-Square feature selection technique has been employed to select the top 19 features. Time, timestamp, and date features were removed due to overfitting problems at training time [48].

4.2.2. Method for data cleaning

This step involves handling missing and null values in the dataset. For Ton_IoT, the median values are calculated for each feature. Afterward, all the missing/ null values have been filled with the estimated median value. The reason behind the selection of median is that it is less vulnerable to outlier mistakes as compared to mean imputation [79].

4.2.3. Method for data encoding

ML deals with numeric data; therefore, the features in categorical form should be converted into numeric values. The conversion of absolute values into numeric values is data encoding. There were six features in the selected dataset that contained categorical values. Temperature-condition, light-status, phone-signals, door-state, thermostat-status, and type. The robust method "Label Encoding" has been used to perform encoding. Label encoding transforms categorical values into numeric in ascending order. The categorical features with values on/off, true/false, high/low, open/close were converted into nominal values 1/0. The "type" feature containing nine different attack names was converted into numerical values 1 to 9.

4.2.4. Method for normalization

The dataset contains multiple features which have a different range of values. The values of some features were in binary (1/0), and some were in hundreds or thousands, which leads to inaccurate results. So, there was a need to rescale all values in a range. To do this, the Min-Max scalar method has been employed on concerned features before modeling it. Min-Max Scaler method is one of the most widely used scaling algorithms. In this method, the minimum of all data (data in one column of the dataset) is subtracted from each value and divided by the minimum and maximum value difference. The equation of Min-Max scaler is shown as:

$$y = \frac{(x - min)}{(max - min)} \tag{12}$$

4.2.5. Training of model

At the stage of training, the dataset has been divided into two parts (training data and testing data) with a ratio of 80: 20. The ratio (80: 20) for splitting the dataset is more recommended in literature [79, 80] due to the fair chance of selection of data in both training and testing sets. Therefore, 80% of data has been used to train the given model, and 20% of data has been used to test the model using the sklearn library's train-test-split method.

4.2.6. Evaluation metrics

To compare the performance of EL methods, the quantitatively used evaluation matrices: precision (PRE), recall (REC), f1- score (FS), and accuracy (ACC) were selected. The detail of each metric is represented in Table 5.

4.3. Proposed stacking-based ensemble model

Stacking is a type of heterogeneous EL paradigm used for classification problems. It is also known as stacked generalization [81]. In the proposed stacking ensemble model, two-level estimators were used to solve the classification of intrusion or normal activities. In the first level, all base learners were trained on the subsets of the dataset for the prediction of output. Four classifiers Naive Bayes (NB) [55, 63, 82] , Linear Discriminant Analysis (LDA) [48, 73] Random Forest (RF) [70, 71] and Linear SVC [66] were selected as base learners. In the next level, Logistic Regression (LR) [64] was selected as a meta-model that takes predictions of all the base learners as input and produces a final prediction that is more accurate than homogeneous EL models. LR is mostly used for binary class classification problems and, with the default setting, cannot be used for multi-class directly. Therefore, the well-known heuristic method one-vs-rest (OvR) has been used in LR in the "multi-class" parameter to perform multiclass classification. OvR scheme divides the multi-class data into several binary class problems, and then the model is trained on each binary class problem. The probability of each has been calculated. The reason for choosing RF as a classifier is that it performs well on telemetry data [48]. Studies [55] and [82] also claims that DT and NB has better performance in IDS respectively. The model is presented to improve the standalone performance of NB, LDA, and Linear SVM with ensemble techniques for telemetry data.

model.pdf model.pdf

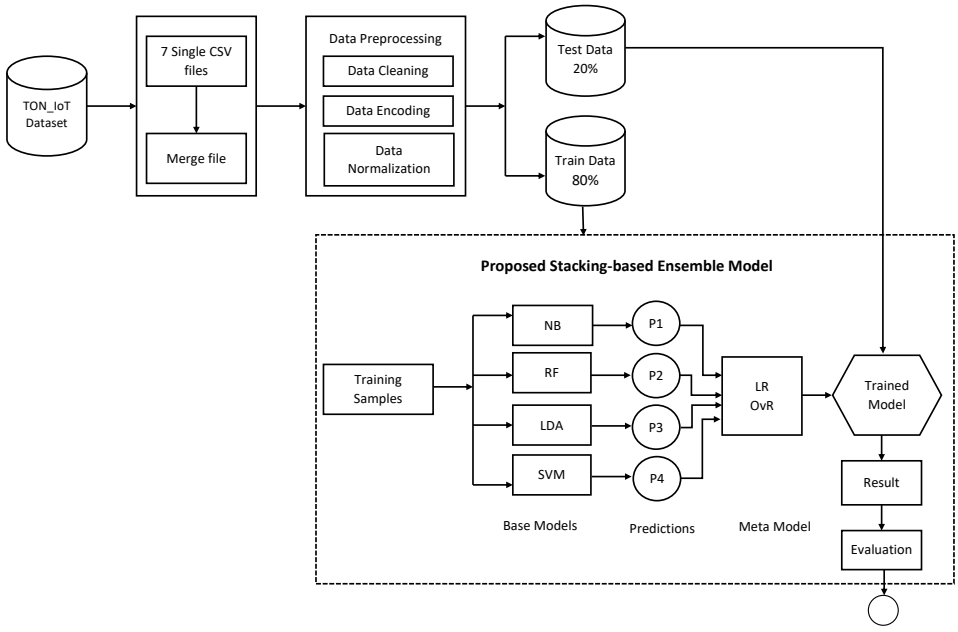

Figure 4. Workflow of proposed methodology.

Table 5. Description of evaluation matrices.

Sr No	Evaluation Matric	Symbolic Representation / Formula	Description
1	True Positive	TP	Number of actual Attacked data observations that are correctly classified as attack
2	True Negative	TN	Number of actual Normal data that are correctly classified as normal
3	False Positive	FP	Number of actual normal instances that are incorrectly predicted as attack
4	False Negative	FN	Number of actual attack observation that are incorrectly predicted as normal
5	Precision	R = TP / TP+FP	Fraction of correctly detected attacked observations and overall detected attack
6	Recall	R = TP / TP+FN	Fraction of correctly detected attacked observation and total number of attack observations in test dataset .
7	Accuracy	Acc= TP+TN / TP+TN+FP+FN	Fraction of sum of correctly detected normal and attacked observation and sum of all correctly and incorrectly normal and attacked observations in dataset.
8	F-Score	F1= 2* (Recall* Precision)/Recall+ Precision	Harmonic Mean of Precision and Recall

5. Results and evaluation

To perform the experiments, the hardware specification of the system includes an Intel (R) Core (TM) i5 CPU running at 3.20 GHz, 16.0 GB of RAM, and Windows 10 Pro. In software specification Jupyter Notebook (Anaconda3) has been used to implement the proposed model in the programming language Python version 3.7.3. Several libraries, such as Pandas and sklearn, etc., have been used to pre-process the model.

The findings and comparison of our suggested model with classic ML models such as LDA, RF, NB, LR, Linear SVC, and LSTM [48] are presented in this section. We have applied K-fold (4,5,7) cross-validation on the given model. The proposed model achieves better results with K = 5. Thus, all the results were obtained using the average value of accuracy, precision, recall, and F-measure. The proposed model is evaluated against each IoT sensor using binary and multi-attack classification. To test the performance of a model for heterogeneous telemetry data (data from different sensors), all the seven dataset files are integrated into one CSV file. The proposed model outperforms binary and multiclass classification.

5.1. Binary classification

5.1.1. Performance of binary classification on per-device(sensor) data files

The binary classification was applied to the dataset using the state of art models RF, LR, NB, LDA, SVM, and the proposed stacking-based-ensemble model (Stacking). Table 6 shows the comparison of the results of binary classification of the proposed model and state-of-art method. The proposed model outperforms most individual IoT sensors data_file compared to traditional ML algorithms. The experimental results depict that a single ML algorithm does not provide more accurate results for all IoT sensor data_files. The performance of the proposed model in terms of accuracy is 100% for fridge sensor and Garage door file compared to linear classifiers such as LR, LDA, NB, SVM with 57, 77, 50, and 81% respectively. The 100% result is that both files contain discrete values, which are easier

Table 6. Binary classification on individual IoT data_files.

File Name	Reference	Model	Evaluation Measure			
			ACC	PRE	REC	FS
IoT_Fridge	**[48]**	LR	0.57	0.34	0.58	0.43
		LDA	0.77	0.79	0.77	0.77
		RF	0.97	0.97	0.97	0.97
		NB	0.50	0.53	0.51	0.51
		SVM	0.81	0.86	0.82	0.80
		LSTM	1.00	1.00	1.00	1.00
	Proposed Model	**stacking**	**1.00**	**100**	**1.00**	**1.00**
IoT_GPS_Tracker	**[48]**	LR	0.86	0.88	0.86	0.87
		LDA	0.86	0.88	0.86	0.87
		RF	0.85	0.85	0.85	0.85
		NB	0.84	0.86	0.85	0.86
		SVM	0.86	0.88	0.87	0.87
		LSTM	0.87	0.89	0.88	0.88
	Proposed Model	**stacking**	**0.95**	**0.96**	**0.96**	**0.96**
IoT_Modbus	**[48]**	LR	0.67	0.46	0.68	0.55
		LDA	0.67	0.46	0.68	0.55
		RF	0.97	0.98	0.98	0.98
		NB	0.67	0.46	0.68	0.55
		SVM	0.67	0.46	0.68	0.55
		LSTM	0.68	0.47	0.68	0.55
	Proposed Model	**stacking**	**0.96**	**0.97**	**0.97**	**0.97**
IoT _Light_Motion	**[48]**	LR	0.58	0.34	0.59	0.43
		LDA	0.58	0.34	0.59	0.43
		RF	0.58	0.34	0.59	0.43
		NB	0.58	0.34	0.59	0.43
		SVM	0.58	0.34	0.59	0.43
		LSTM	0.59	0.35	0.59	0.44
	Proposed Model	**stacking**	**0.59**	**0.35**	**0.59**	**0.44**
IoT_Weather	**[48]**	LR	0.58	0.60	0.59	0.53
		LDA	0.60	0.59	0.60	0.53
		RF	0.84	0.84	0.84	0.84
		NB	0.69	0.72	0.69	0.67
		SVM	0.63	0.68	0.63	0.55
		LSTM	0.82	0.82	0.81	0.80
	Proposed Model	**stacking**	**0.97**	**0.98**	**0.98**	**0.98**

to work with. The model outperforms both GPS_tracker and Mod_bus sensor data file, then ML and LSTM (a DL model) with 95 and 96%, respectively. The model gave an accuracy of 97% for weather sensors data. In contrast, The achieved accuracy and other evaluation metrics (precision, recall, and F1-score) from stacking EL are comparably same as the ML models for Garage door and light motion file, presented in Table 7 due to the heterogeneity of data sources in Ton-IoT dataset.

Table 7. Binary classification on individual IoT data_files.

File Name	Reference	Model	Evaluation measure			
			ACC	PRE	REC	FS
IoT_Garage_Door	**[48]**	LR	1.00	1.00	1.00	1.00
		LDA	1.00	1.00	1.00	1.00
		RF	1.00	1.00	1.00	1.00
		NB	1.00	1.00	1.00	1.00
		SVM	1.00	1.00	1.00	1.00
		LSTM	1.00	1.00	1.00	1.00
	Proposed Model	**stacking**	**1.00**	**1.00**	**1.00**	**1.00**
IoT_Thermostat	**[48]**	LR	0.66	0.44	0.66	0.53
		LDA	0.66	0.44	0.66	0.53
		RF	0.66	0.59	0.66	0.53
		NB	0.66	0.44	0.66	0.53
		SVM	0.66	0.44	0.66	0.53
		LSTM	0.66	0.45	0.67	0.54
	Proposed Model	**stacking**	**0.66**	**0.44**	**0.67**	**0.53**

5.1.2. Performance of binary classification for merged IoT telemetry data

The data of each sensor file is automatically combined into a single CSV file (known as merge IoT data) by implementing a python script. The three features, date, time, and timestamp, were removed from the feature vector due to the over-fitting problem. From the results depicted in Table 8, it is evident that model achieves the 86% accuracy as compare to linear classifiers such as LR (61%), LDA (68%), NB (62%), SVM (61%), LSTM (81%). The proposed model has better accuracy, precision, recall, and F1-score results than LSTM in the merge file. The success of our model is due to its ensemble nature which combines the performance of individual classifiers and boosts up the results.

5.1.3. Cross comparison of proposed model and homogeneous ensemble model(bagging)

Bagging is homogeneous EL that uses the same base model for the classification. Different base models such as Decision Tree, KNN, SVC, and Logistic Regression are used in bagging to conduct the experiments. They were finally bagging with Decision Tree as base estimator was selected for comparison because it achieves better results than others. The proposed model, which is heterogeneous in type, is compared with bagging (with DT). Comparison of both homogeneous and proposed heterogeneous techniques for binary classification for each file and merged file are presented in Table 9. The results depicted that the proposed heterogeneous ensemble model outperforms another bagging in

Table 8. Binary classification on merged IoT telemetry data.

File name	Reference	Model	Evaluation measure			
			ACC	PRE	REC	FS
Combine file	[48]	LR	0.61	0.37	0.61	0.46
		LDA	0.68	0.74	0.68	0.62
		RF	0.85	0.87	0.85	0.85
		NB	0.62	0.63	0.62	0.51
		SVM	0.61	0.37	0.61	0.46
		LSTM	0.81	0.83	0.81	0.80
	Proposed model	**stacking**	**0.86**	**0.87**	**0.85**	**0.86**

most cases for Binary classification. However, in the case of Merged data_file, the accuracy and other evaluation parameters are comparable to the bagging model in binary type.

Table 9. Binary classification comparison of homogeneous and proposed heterogeneous model.

Data files	Bagging with decision tree					Proposed model		
	Mode	ACC	PRE	REC	FS	ACC	PRE	REC	FS
IoT_Fridge	Single	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
IoT_Garage_Door		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
IoT_GPS_Tracker		0.92	0.92	0.92	0.92	0.95	0.96	0.96	0.96
IoT_Modbus		0.95	0.96	0.96	0.96	0.96	0.97	0.97	0.97
IoT_Light_Motion		0.58	0.35	0.59	0.44	0.59	0.35	0.59	0.44
IoT_Thermostat		0.59	0.55	0.58	0.56	0.66	0.44	0.67	0.53
IoT_Weather		0.89	0.89	0.89	0.89	0.97	0.98	0.98	0.98
Integrated_Ton_IoT	Combine	0.82	0.88	0.87	0.86	0.86	0.87	0.85	0.86

Table 10. Proposed stacking model results of multi classification on per-device CSV file.

File Name	ACC	PRE	REC	FS
IoT_Fridge	0.66	0.65	0.67	0.65
IoT_Garage_Door	0.66	0.62	0.67	0.62
IoT_GPS_Tracker	0.91	0.91	0.91	0.91
IoT_Modbus.csv	0.98	0.99	0.99	0.99
IoT_Light_Motion	0.66	0.60	0.66	0.61
IoT_Thermostat	0.76	0.75	0.76	0.75
IoT_Weather	0.99	0.99	0.99	0.99

Table 11. Multi classification on merged IoT data_files.

File name	Reference	Model	Evaluation measure			
			ACC	PRE	REC	FS
Combine file	**[48]**	LR	0.61	0.38	0.62	0.47
		LDA	0.62	0.46	0.63	0.51
		RF	0.71	0.69	0.72	0.67
		NB	0.54	0.59	0.51	0.52
		SVM	0.60	0.37	0.61	0.46
		LSTM	0.68	0.64	0.68	0.63
	Proposed model	**stacking**	**0.77**	**0.80**	**0.75**	**0.74**

5.2. Multiclass-classification

5.2.1. Performance of multi classification on per-device (sensor) IoT telemetry data file

The proposed model is applied for multi-classification with all 19 features of the dataset. The time, timestamp, and date are excluded due to the mentioned reason above. The TON_IoT dataset contains a feature name "type." The type feature includes seven different IoT attacks names, which helps identify the attack type. The evaluation of the proposed model for the multi_classification problem is presented in this section. The proposed model has LR as a meta-model mainly used for binary classification problems. LR with default setting cannot be used for multi_class directly. Therefore, the well-known heuristic method one-vs-rest (OvR) has been used in LR in multi_class parameter. OvR scheme divides the multi-class data into several binary class problems, and then the model is trained on each binary class problem. The probability of each has been calculated. Finally, the class with the highest probability has been selected as the final prediction [79]. The results for multi-classification for individual IoT sensor data_files are shown in Table 10. The results depict that the proposed model achieves good results for mod_bus weather_file with all metrics up to 98 and 99%, respectively. The model shows similar results for fridge, light_motion, and garage_Door sensors. The accuracy score for these files is 66%. The proposed model outperforms all those data_files that have heterogeneous sensors data compared to the data_files that have binary data in case of multi_class classification.

5.2.2. Performance of multi_classification for merged IoT telemetry data

The summary of multi_classification for merge file is presented in Table 11. The results show that single ML classifiers do not perform well for multi_classification. RF has a good score of 71% accuracy whereas other classifiers (i.e., LR, LDA, NB, SVM) performance is not more excellent than 62%. In contrast, our proposed model outperforms with 77% accuracy, 80% precision, and 75% recall.

5.2.3. Cross comparison of the proposed model and homogeneous ensemble model(bagging in multi_classification)

In multi_classification, both the proposed model and bagging (DT) achieve almost similar results presented in Table 12 for individual files as well as for merge files in terms of accuracy(bagging = 76% and stacking = 77%).

Table 12. Multi classification comparison of homogeneous and proposed heterogeneous model.

Data files	Bagging with decision tree					Proposed stacking model			
	Mode	ACC	PRE	REC	FS	ACC	PRE	REC	FS
IoT_Fridge	Single	0.66	0.65	0.66	0.65	0.66	0.65	0.67	0.65
IoT_Garage_Door		0.67	0.62	0.67	0.62	0.66	0.62	0.67	0.62
IoT_GPS_Tracker		0.84	0.84	0.84	0.84	0.91	0.91	0.91	0.91
IoT_Modbus		0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99
IoT_Light_Motion		0.67	0.60	0.67	0.61	0.66	0.60	0.66	0.61
IoT_Thermostat		0.75	0.75	0.76	0.76	0.76	0.75	0.76	0.75
IoT_Weather		0.95	0.95	0.95	0.95	0.99	0.99	0.99	0.99
Integrated_Ton_IoT	Combine	0.76	0.87	0.79	0.79	0.77	0.86	0.79	0.79

5.3. Discussion

In an IoT network, devices are interdependent to make smart decisions. Attackers capture the data of IoT devices to temper it and alter the decision of other IoT objects within the network. This study examines the performance of ensemble learning for IDS designed for telemetry data. The analysis of experimental results shows that the proposed model gives optimum performance as compared to other single ML models (i.e., LR, LDA, RF, NB, SVM), a deep learning model LSTM, and heterogeneous model (bagging with DT). It is evident from the results that a single classifier does not produce accurate results for all sensors. The proposed model can detect both binary and multi-class attacks for single sensor data and for merge data of multiple sensors. To test the performance of the proposed model for heterogeneous data, the data is integrated into one file from all sensors. Moreover, we have applied Kfold (with K = 4, K = 5, K = 7) cross-validation on the given model. The proposed model achieves better results with K = 5. The RF in proposed model boost up the performance of LDA and SVM. Therefore, our model achieves significant increase in accuracy for both binary and multi-class attack detection.

6. Conclusions

This study investigates the performance of ensemble learning techniques for IDs of sensors telemetry data in IoT networks by using ML classifiers. A stacking-based ensemble model has been proposed for making the devices more intelligent. The proposed stacking ensemble integrated NB, LDA, Linear SVC, RF as base classifiers, and LR was selected as meta classifier. The experiments were evaluated on a recently publicly available ToN-IoT telemetry data set. The ToN-IoT dataset is an accurate representation of telemetry data. The experimental results reveal that the stacking ensemble technique achieves high accuracy for all sensors data compared to the stand-alone ML classifier and bagging ensemble model. For binary classification, the performance of the proposed model compared to the state-of-art classifiers in terms of accuracy was 100, 95, 97% for the fridge device, GPS_tracker device, weather sensors, respectively. However, the accuracy performance for Garage_door and Thermostat devices were comparably the same as the ensemble model. The proposed model achieved a significant increase of 86% in accuracy for merge file in binary classification and 77% in multi_classification.

In future work, a deep learning model will be used and extended to improve the proposed model's accuracy in identifying unusual and novel types of IoT intrusions in IoT devices.

Acknowledgement

The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, S. Guizani, Internet-of-things-based smart cities: Recent advances and challenges, *IEEE Commun. Mag.*, **55** (2017), 16–24. https://ieeexplore.ieee.org/document/8030479
2. H. Yang, J. Yuan, C. Li, G. Zhao, Z. Sun, Q. Yao, et al., BrainIoT: Brain-like productive services provisioning with federated learning in industrial IoT, *IEEE Int. Things J.*, **9** (2021), 2014–2024. https://ieeexplore.ieee.org/document/9454442
3. B. Bera, S. Saha, A. K. Das, A. V. Vasilakos, Designing blockchain-based access control protocol in iot-enabled smart-grid system, *IEEE Int. Things J.*, **8** (2020), 5744–5761. https://ieeexplore.ieee.org/document/9222155
4. M. Hatamian, M. A. Bardmily, M. Asadboland, M. Hatamian, H. Barati, Congestion-aware routing and fuzzy-based rate controller for wireless sensor, *Radioengineering*, **25** (2016), 114–123. https://www.radioeng.cz/fulltexts/2016/16_01_0114_0123.pdf
5. M. Nobakht, V. Sivaraman, R. Boreli, A host-based intrusion detection and mitigation framework for smart home iot using openflow, *IEEE*, (2016), 147–156. https://ieeexplore.ieee.org/document/7784565
6. H. Alkahtani, T. H. Aldhyani, M. Al-Yaari, Adaptive anomaly detection framework model objects in cyberspace, *Appl. Bionics Biomech.*, **2020** (2020). https://doi.org/10.1155/2020/6660489
7. T. H. Aldhyani, M. R. Joshi, Intelligent time series model to predict bandwidth utilization, *Int. J. Comput. Sci. Appl.*, **14** (2017).
8. M. Tang, M. Alazab, Y. Luo, Big data for cybersecurity: Vulnerability disclosure trends and dependencies, *IEEE Trans. Big Data,* **5** (2017), 317–329. https://ieeexplore.ieee.org/document/7968482
9. V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, E. A. Nasr, Robust attack detection approach for iiot using ensemble classifier, preprint, arXiv:2102.01515.
10. S. I. E-ISAC, Analysis of the cyber attack on the ukrainian power grid, 2016.

11. Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, D. Qiu, Security of the internet of things: perspectives and challenges, *Wireless Networks*, **20** (2014), 2481–2501. https://doi.org/10.1007/s11276-014-0761-7
12. A. N. Shahbaz, H. Barati, A. Barati, Multipath routing through the firefly algorithm and fuzzy logic in wireless sensor networks, *Peer-to-Peer Networking Appl.*, **14** (2021), 541–558. https://doi.org/10.1007/s12083-020-01004-2
13. A. Mosavifard, H. Barati, An energy-aware clustering and two-level routing method in wireless sensor networks, *Computing*, **102** (2020), 1653–1671. https://doi.org/10.1007/s00607-020-00817-6
14. N. N. Dezfuli, H. Barati, Distributed energy efficient algorithm for ensuring coverage of wireless sensor networks, *IET Commun.*, **13** (2019), 578–584. https://doi.org/10.1049/iet-com.2018.5329
15. N. N. Dezfouli, H. Barati, A distributed energy-efficient approach for hole repair in wireless sensor networks, *Wireless Networks*, **26** (2020), 1839–1855. https://doi.org/10.1007/s11276-018-1867-0
16. W. Zhou, Y. Jia, A. Peng, Y. Zhang, P. Liu, The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved, *IEEE Int. Things J.*, **6** (2019), 1606–1616. https://ieeexplore.ieee.org/document/8386824
17. J. Ni, K. Zhang, A. V. Vasilakos, Security and privacy for mobile edge caching: Challenges and solutions, *IEEE Wireless Commun.*, **28** (2020), 77–83. https://doi.org/10.48550/arXiv.2012.03165
18. W. Zhou, Y. Jia, A. Peng, Y. Zhang, P. Liu, The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved, *IEEE Int. Things J.*, **6** (2019), 1606–1616. https://ieeexplore.ieee.org/document/8386824
19. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, *Cybersecurity*, **2** (2019), 1–22. https://doi.org/10.1186/s42400-019-0038-7
20. N. H. Al-A'araji, S. O. Al-Mamory, A. H. Al-Shakarchi, Classification and clustering based ensemble techniques for intrusion detection systems: A survey, in *Journal of Physics: Conference Series*, **1818** (2021), 012106.
21. J. Arshad, M. Azad, M. Abdeltaif, K. Salah, An intrusion detection framework for energy constrained IoT devices, *Mech. Syst. Signal Process.*, **136** (2020), 106436. https://doi.org/10.1016/j.ymssp.2019.106436
22. S. S. Sharifi, H. Barati, A method for routing and data aggregating in cluster-based wireless sensor networks, *Int. J. Commun. Syst.*, **34** (2021), e4754. https://doi.org/10.1002/dac.4754
23. J. P. Amaral, L. M. Oliveira, J. J. Rodrigues, G. Han, L. Shu, Policy and network-based intrusion detection system for ipv6-enabled wireless sensor networks, *IEEE*, (2014), 1796–1801. https://ieeexplore.ieee.org/document/6883583
24. B. S. Bhati, G. Chugh, F. Al-Turjman, N. S. Bhati, An improved ensemble based intrusion detection technique using xgboost, *Trans. Emerging Telecommun. Technol.*, (2020), e4076. https://doi.org/10.1002/ett.4076

25. M. Roesch, Snort: Lightweight intrusion detection for networks, *Lisa*, **99** (1999), 229–238.
26. J. Singh, M. J. Nene, A survey on machine learning techniques for intrusion detection systems, *Int. J. Adv. Res. Comput. Commun. Eng.*, **2** (2013), 4349–4355.
27. T. H. Lee, C. H. Wen, L. H. Chang, H. S. Chiang, M. C. Hsieh, A lightweight intrusion detection scheme based on energy consumption analysis in 6lowpan, in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*, (2014), 1205–1213. https://doi.org/10.1007/978-94-007-7262-5_137
28. S. Raza, L. Wallgren, T. Voigt, Svelte: Real-time intrusion detection in the internet of things, *Ad Hoc Networks*, **11** (2013), 2661–2674. https://doi.org/10.1016/j.adhoc.2013.04.014
29. B. B. Zarpel˜ao, R. S. Miani, C. T. Kawakani, S. C. de Alvarenga, A survey of intrusion detection in internet of things, *J. Network Comput. Appl.*, **84** (2017), 25–37. https://doi.org/10.1016/j.jnca.2017.02.009
30. H. Wong, T. Luo, Man-in-the-middle attacks on mqtt-based iot using bert based adversarial message generation, *KDD'20*, 2020.
31. M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A. P. Sheth, Machine learning for internet of things data analysis: A survey, *Digital Commun. Networks*, **4** (2018), 161–175. https://doi.org/10.1016/j.dcan.2017.10.002
32. K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, V. H. C. de Albuquerque, Internet of things: A survey on machine learning-based intrusion detection approaches, *Comput. Networks*, **151** (2019), 147–157. https://doi.org/10.1016/j.comnet.2019.01.023
33. N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for iot security based on learning techniques, *IEEE Commun. Surv. Tutor.*, **21** (2019), 2671–2701. https://ieeexplore.ieee.org/document/8629941
34. M. Dibaei, X. Zheng, Y. Xia, X. Xu, A. Jolfaei, A. K. Bashir et al., Investigating the prospect of leveraging blockchain and machine learning to secure vehicular networks: A survey, *IEEE Trans. Intell. Transp. Syst.*, **23** (2021), 683–700. https://ieeexplore.ieee.org/document/9519843
35. X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying et al., Privacy and security issues in deep learning: A survey, *IEEE Access*, **9** (2020), 4566–4593. https://ieeexplore.ieee.org/document/9294026
36. M. Wazid, A. K. Das, V. Bhat, A. V. Vasilakos, Lam-ciot: Lightweight authentication mechanism in cloud-based iot environment, *J. Network Comput. Appl.*, **150** (2020), 102496. https://doi.org/10.1016/j.jnca.2019.102496
37. E. Borgia, The internet of things vision: Key features, applications and open issues, *Comput. Commun.*, **54** (2014), 1–31. https://doi.org/10.1016/j.comcom.2014.09.008
38. F. Papi, H. Barati, Hdrm: A hole detection and recovery method in wireless sensor network, *Int. J. Commun. Syst.*, **35** (2022), e5120. https://doi.org/10.1002/dac.5120
39. E. G. Dehkordi, H. Barati, Cluster based routing method using mobile sinks in wireless sensor network, *Int. J. Electron.*, (2022), 1–13. https://doi.org/10.1080/00207217.2021.2025451

40. X. Liu, M. Zhao, S. Li, F. Zhang, W. Trappe, A security framework for the internet of things in the future internet architecture, *Future Int.*, **9** (2017), 27. https://doi.org/10.3390/fi9030027

41. E. Yousefpoor, H. Barati, A. Barati, A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks, *Peer-to-Peer Networking Appl.*, **14** (2021), 1917–1942. https://link.springer.com/article/10.1007/s12083-021-01116-3

42. E. Hasheminejad, H. Barati, A reliable tree-based data aggregation method in wireless sensor networks, *Peer-to-Peer Networking Appl.*, **14** (2021), 873–887. https://doi.org/10.1007/s12083-020-01025-x

43. M. Naghibi, H. Barati, Shsda: secure hybrid structure data aggregation method in wireless sensor networks, *J. Ambient Intell. Human. Comput.*, **12** (2021), 10769–10788. https://doi.org/10.1007/s12652-020-02751-z

44. S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, R. Boreli, An experimental study of security and privacy risks with emerging household appliances, in *2014 IEEE Conference on Communications and Network Security*, (2014), 79–84. https://ieeexplore.ieee.org/document/6997469

45. Z. Hajipour, H. Barati, Eelrp: energy efficient layered routing protocol in wireless sensor networks, *Computing*, **103** (2021), 2789–2809. https://doi.org/10.1007/s00607-021-00996-w

46. A. Arabo, I. Brown, F. El-Moussa, Privacy in the age of mobility and smart devices in smart homes, in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, (2012), 819–826. https://ieeexplore.ieee.org/document/6406331

47. M. Al-Hawawreh, E. Sitnikova, F. den Hartog, An efficient intrusion detection model for edge system in brownfield industrial internet of things, in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, (2019), 83–87. https://doi.org/10.1145/3361758.3361762

48. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, Ton iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems, *IEEE Access*, **8** (2020), 165130–165150. https://ieeexplore.ieee.org/document/9189760

49. E. Anthi, L. Williams, M. Słowinska, G. Theodorakopoulos, P. Burnap, A supervised intrusion detection system for smart home iot devices, *IEEE Int. Things J.*, **6** (2019), 9042–9053. https://ieeexplore.ieee.org/document/8753563

50. S. U. Jan, S. Ahmed, V. Shakhov, I. Koo, Toward a lightweight intrusion detection system for the internet of things, *IEEE Access*, **7** (2019), 42450–42471. https://ieeexplore.ieee.org/document/8675917

51. F. Ertam, L. F. Kilincer, O. Yaman, Intrusion detection in computer networks via machine learning algorithms, in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, (2017), 1–4. https://ieeexplore.ieee.org/document/8090165

52. T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, M. A. Khan, Performance analysis of machine learning algorithms in intrusion detection system: A review, *Procedia Comput. Sci.*, **171** (2020), 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

53. O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, A. Althunibat, Machine learning classifiers for network intrusion detection system: comparative study, in *International Conference on Information Technology (ICIT)*, (2021), 440–445. https://ieeexplore.ieee.org/document/9491770

54. B. S. Bhati, G. Chugh, F. Al-Turjman, N. S. Bhati, An improved ensemble based intrusion detection technique using xgboost, *Trans. Emerging Telecommun. Technol.*, **32** (2021), e4076. https://doi.org/10.1002/ett.4076

55. M. Belouch, S. E. hadaj, Comparison of ensemble learning methods applied to network intrusion detection, in *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, (2017), 1–4. https://doi.org/10.1145/3018896.3065830

56. N. Moustafa, B. Turnbull, K. K. R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, *IEEE Int. Things J.*, **6**1 (2018), 4815–4830. https://ieeexplore.ieee.org/document/8470090

57. V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, E. A. Nasr, Robust attack detection approach for iiot using ensemble classifier, preprint, arXiv:2102.01515.

58. A. Verma, V. Ranga, Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things, in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, (2019), 1–6. https://ieeexplore.ieee.org/document/8777504

59. A. Elijah, A. Abdullah, N. Jhanjhi, M. Supramaniam, B. Abdullateef, Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study, *Int. J. Adv. Comput. Sci. Appl.*, **10** (2019), 520–528. 10.14569/IJACSA.2019.0100969

60. Y. Wang, Y. Shen, G. Zhang, Research on intrusion detection model using ensemble learning methods, in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, (2016), 422–425. https://ieeexplore.ieee.org/document/7883100

61. B. A. Tama, M. Comuzzi, K. H. Rhee, Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system, *IEEE Access*, **7** (2019), 94497–94507. https://ieeexplore.ieee.org/document/8759867

62. R. Primartha, B. A. Tama, Anomaly detection using random forest: A performance revisited, in *2017 International Conference on Data and Software Engineering (ICoDSE)*, (2017), 1–6. https://ieeexplore.ieee.org/document/8285847

63. R. Abdulhammed, M. Faezipour, A. Abuzneid, A. AbuMallouh, Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic, *Int. J. Adv. Comput. Sci. Appl.*, **10** (2019), 1–4. https://ieeexplore.ieee.org/document/8526292

64. C. Ioannou, V. Vassiliou, An intrusion detection system for constrained wsn and iot nodes based on binary logistic regression, in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems,* (2018), 259–263. https://doi.org/10.1145/3242102.3242145

65. P. Ghosh, R. Mitra, Proposed ga-bfss and logistic regression based intrusion detection system, in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)* (2015), 1–6. https://ieeexplore.ieee.org/document/7060117

66. S. U. Jan, S. Ahmed, V. Shakhov, I. Koo, Toward a lightweight intrusion detection system for the internet of things, *IEEE Access*, **7** (2019), 42450–42471. https://ieeexplore.ieee.org/document/8675917

67. B. Sharmila, R. Nagapadma, Intrusion detection system using naive bayes algorithm, in *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE.IEEE*, (2019), 1–4. https://ieeexplore.ieee.org/document/9019921

68. Z. Muda, W. Yassin, M. Sulaiman, N. Udzir, Intrusion detection based on k-means clustering and naıve bayes classification, in *2011 7th International Conference on Information Technology in Asia*, (2011), 1–6. https://ieeexplore.ieee.org/document/5999520

69. A. Halimaa, K. Sundarakantham, Machine learning based intrusion detection system, in *2019 3rd International conference on trends in electronics and informatics (ICOEI)*, (2019), 916–920. https://ieeexplore.ieee.org/document/8862784

70. N. Farnaaz, M. Jabbar, Random forest modeling for network intrusion detection system, *Procedia Comput. Sci.*, **89** (2016), 213–217. https://doi.org/10.1016/j.procs.2016.06.047

71. M. A. M. Hasan, M. Nasser, B. Pal, S. Ahmad, Support vector machine and random forest modeling for intrusion detection system (ids), *J. Intell. Learn. Syst. Appl.*, **2014** (2014). https://www.scirp.org/journal/paperinformation.aspx?paperid=42869

72. D. Zheng, Z. Hong, N. Wang, P. Chen, An improved lda-based elm classification for intrusion detection algorithm in iot application, *Sensors*, **20** (2020), 1706. https://doi.org/10.3390/s20061706

73. B. Subba, S. Biswas, S. Karmakar, Intrusion detection systems using linear discriminant analysis and logistic regression, in *2015 Annual IEEE India Conference (INDICON)*, (2015), 1–6. https://ieeexplore.ieee.org/document/7443533

74. L. Rutkowski, M. Jaworski, L. Pietruczuk, P. Duda, Decision trees for mining data streams based on the gaussian approximation, *IEEE Trans. Knowl. Data Eng.*, **26** (2013), 108–119. https://ieeexplore.ieee.org/document/6466324

75. B. V. Dasarathy, B. V. Sheela, A composite classifier system design: Concepts and methodology, *Proc. IEEE*, **67** (1979), 708–713. https://ieeexplore.ieee.org/document/1455590

76. A. Lal, B. Datta, Performance evaluation of homogeneous and heterogeneous ensemble models for groundwater salinity predictions: A regional-scale comparison study, *Water, Air, Soil Pollut.*, **231** (2020), 1–21. https://doi.org/10.1007/s11270-020-04693-w

77. A. Subasi, S. Algebsani, W. Alghamdi, E. Kremic, J. Almaasrani, N. Abdulaziz, Intrusion detection in smart healthcare using bagging ensemble classifier, in *International Conference on Medical and Biological Engineering*, (2021), 164–171. https://doi.org/10.1007/978-3-030-73909-6_18

78. U. Sydney, The ton-iot dataset, 2020. Available from: https://research.unsw.edu.au/projects/toniot-datasets.

79. A. Geron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019.

80. V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, E. A. Nasr, Robust attack detection approach for iiot using ensemble classifier, preprint, arXiv2102.01515.

81. S. Rajagopal, P. P. Kundapur, K. S. Hareesha, A stacking ensemble for network intrusion detection using heterogeneous datasets, *Secur. Commun. Networks*, **2020** (2020). https://doi.org/10.1155/2020/4586875

82. S. Fenanir, F. Semchedine, A. Baadache, A machine learning-based lightweight intrusion detection system for the internet of things, *Rev. d'Intell. Artif.*, **33** (2019), 203–211. https://doi.org/10.18280/ria.330306

AIMS Press