Expert Systems With Applications 215 (2023) 119330

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks

Shahbaz Ahmad Khanday, Hoor Fatima *, Nitin Rakesh

Computer Science Engineering, Sharda University, Uttar Pradesh 201306, India

ARTICLE INFO

Keywords:
Lightweight IoT networks
DDoS
Smote
BOTIoT
TONIoT

ABSTRACT

Protecting IoT networks and infrastructure is one of the top priorities in today's computing industry because of the unnerving and exponential development in cyberattacks and security breaches in IoT. Lightweight IoT networks had been one of the easiest targets for attackers in botnet formation and distributing malware. The research in the paper identifies IoT networks formed by devices with minimal computing resources, such as less battery life, processing power, memory, and more importantly, minimal security, protecting infrastructure, and defense mechanisms, as being lightweight IoT networks that are easily vulnerable to DDoS attacks and disseminating malware. It is investigated by many researchers that development and progress in intrusion detection systems is the need of an hour to safeguard lightweight IoT networks. The manuscript proposes a lightweight Intrusion detection system with a novel data pre-processing technique while using machine learning and deep learning classifiers. The manuscript introduces various types of classifiers, employed to form light-weight intrusion detection systems well suited for protection against Distributed Denial of Services attacks in IoT networks. The datasets used for the experiments and investigation are BOT-IoT and the Network dataset of TON-IoT by the University of New South Wales Sydney (UNSW) Australia. DDoS attack instances are derived from both datasets. Two different experiments are performed on each dataset i.e.; for binary classifications of attack labels, one experiment for all attacks, and another experiment for the DDoS attack only in both datasets. Attack classes in the BOT-IoT dataset compared with the TON-IoT dataset are highly imbalanced. We have used Synthetic Minority oversampling Technique (Smote()) variants for class balancing in the experiments performed on the BOT-IoT dataset.

1. Introduction

DDoS attacks aimed at IoT devices have the potential to seriously disrupt IoT metasystems. According to the latest study by Security Today and Threat Post, there will be nearly 31 billion connected IoT devices by 2020, with about half of them becoming unsecured and vulnerable to most common cyber-attacks including DDoS attacks (A.y, X.x, & T.r, 2021; Antonakakis et al., 2017). Botnets can be created using vulnerable IoT devices, posing a threat to the security of IoT ecosystems via stealthy DDoS attacks. One of the famous and well-known examples is the 2016 Dyn cyberattack, which utilized Houses installed with smart devices and linked gadgets will be recruited into "botnets" by the Mirai malware (Chaabouni et al., 2019). According to Avast Cybercriminals assault, computer systems with enormous distributed denial of service (DDoS) attacks using Mirai botnets (Anon, n.d.). Many researchers have investigated and found that the Mirai security breach has led to the emergence of new evolution botnets, such as Torii, Sling-shot in 2018, and Persirai, BrickerBot in 2017. Since 2008, Angrishi et al. and De Donno et al. (De donno et al., 2018) have provided in-depth investigations and linkages between IoT botnet malware (Angrishi, 2017).

1.1. Motivation

Since the IoT network has limited resources, the attackers are not naive and have developed several inventive techniques to breach it. IoT devices that number in the billions typically create a lot of traffic. Therefore, it can be difficult to differentiate between a regular traffic increase and a DDoS attack. Intelligence to identify DDoS is required to address these problems. Both Machine learning and Deep learning can take on predictive roles across a variety of use cases, giving organizations access to new ideas and encouraging the need for automation tools and intelligent application development for tackling DDoS attacks on IoT infrastructure. It has been determined after extensive research that machine learning and deep learning algorithms are among the most

* Corresponding author.

https://doi.org/10.1016/j.eswa.2022.119330
Received 28 April 2022; Received in revised form 12 November 2022; Accepted 19 November 2022
Available online 25 November 2022

effective and ideal ways to secure IoT networks. One such concern comes from the fact that an upgrade and rise in number of DDoS assaults try to focus on attacking consumer-level IoT devices, in addition to users' lack of technical competence or comprehension of intrinsic weaknesses. The premise that several users of IoT devices lack an intuitive user interface and make it practically impossible for customers to detect attacks on their home networks makes this situation far more challenging (Nascita et al., 2021). proposed multimodal deep learning model for mobile traffic classification due to rise of smartphone usage.

The development of immune solutions for IoT networks and devices to detect intrusions remains one of the prime targets for researchers to achieve a higher success rate in detecting malicious activities towards IoT infrastructure. To monitor and restrict inappropriate data flows in the IoT network, IoT security needs to identify undesirable and malicious traffic. To avoid illicit traffic flows in the Network infrastructure, many researchers have proposed a diversity of intrusion detection models employing machine learning as well as deep learning methods. However, due to incorrect and improper feature selection in some ML models, traffic flows that are most damaging are often misclassified. The crucial question, namely, it is necessary to conduct more research on how to select beneficial qualities for precise fraudulent traffic recognition in IoT networks.

In this study, we suggested an innovative dataset pre-processing method to extract the feature importance from the input feature vector using an ensemble strategy and then use feature importance and impurity. Then uses the feature drop-out technique to drop features with minimum importance. The manuscript introduces a set of classifiers including both machine learning and deep learning models for anomaly detection in the IoT network traffic data and comparative performance analysis of each classifier. Furthermore, the detection of DDoS attack patterns and DDoS capable malware distribution remains the primary target of the research. The manuscript leverages the proposed pre-processing method and feature selection explained in detail below in section 3 of the manuscript to focus on the detection of DDoS attack labels and malware spread in IoT network data traffic. To test our proposed model, research and testing are done on two separate datasets, BoT-IoT and TON-IoT datasets, which have a range of IoT cyber-attack labels. Two categories—one for all attack labels in both datasets and another for DDoS attack labels in both datasets—are used to categorize the tests. The contribution of the research are as follows:

- A novel data pre-processing strategy with an ensemble feature selection mechanism for IoT network datasets.
- Extraction of traffic patterns with DDoS attack label vs All attack labels contained by both experimental datasets.
- Performance evaluation and comparison of deep learning and machine learning classifiers.

1.2. Distributed Denial of Services attacks burgeoning IoT

The DDoS attack can remove or block a registered user from the services of the network and infect the nodes/devices of the network while making a zombie network of infected nodes. Doshi et al. define minimally invasive mitigation as one of the primary security challenges in IoT infrastructure that pose a tremendous issue due to its simple yet very potent attack progression, the today's Internet ecosystem (Doshi, Yilmaz, & Uludag, 2021). It is particularly challenging to identify the assaulting devices due to the scattered nature of modern DDoS attacks. To counteract the attack, however, there should be little service disruption for non-malicious customers who wish to utilize targeted services lawfully.

Many applications and creations of suitable systems and approaches are unable to endure DDoS attacks. Instead, it is becoming more intense and larger. Therefore, it's also crucial to comprehend why these DDoS defensive tactics need to be improved to stop and discourage attacks. In developing coherent applications to tackle malware distribution by bots, it is important to understand the malicious source's architecture, propagation, and flow amid DDoS infection. McDermott et al. define the botnet infection and propagation technique used in Maria by the botmaster (McDermott, Majdani, and Petrovski 2018). The botnet propagation is depicted in Fig. 2 below.

- **Botmaster** The "botmaster," who oversees the botnet infrastructure, uses the stolen machines to launch attacks intended to take down a target's network, install malware, steal login credentials, or execute CPU-concerted procedures.
- **Command and control server:** The botmaster intends to send infected commands to vulnerable IoT devices. Once compromised the IoT device becomes part of the bot army under the control of its master and can communicate it for further instruction.
- **Botnet:** A network formed by the compromised IoT devices used to distribute malware.
- **Target Server:** it resembles a target to the botmaster, can be a website an IoT device, or another network asset.

Structure of the Paper: The organization and reading map of the paper is given below in Fig. 2.

2. Literature survey

To be effective against DDoS attacks over the Internet, diverse DDoS protection solutions were put out, put into practice, and tested. For a potential remedy to the DDoS attack over an IoT network, the most prominent defense designs are to be evaluated in this area**.** The nature of IoT is covered in the literature review section, along with the reasons why IoT networks may be susceptible to DDoS attacks. Vishwakarma et al. discussed the taxonomy of anti-DDoS strategies in IoT networks and traditional DDoS attacks (Vishwakarma & Jain, 2020). IoT IoT-specific attacks are further divided into Malware distribution (Network based) and Prevention based. The table below explains the associated technologies with DDoS assaults (Table 2).

2.1. Proposed model for lightweight intrusion detection

The two most well-liked fields of artificial intelligence (AI), namely deep learning models and machine learning algorithms for binary and multiple classifications, are used in the construction of a significant number of intrusion detection systems (IDS). The proposed model presented in this paper is three machine learning algorithms including ensemble algorithms and two deep learning algorithms, while any of them could be used as a classifier. Deep learning (DL) opens up many new possibilities for traditional AI applications in harvesting meaningful information from images, pattern recognition, and computer vision. Apart from the general classification and regression problems intrusion detection systems have to play a huge role in analyzing and examining the IoT network behavior when compared to the firewalls and eventually services provided by the firewalls. It has been proved that better correctness and application performance brought on by DL's capacity for the improved intellect can be used to spot fresh disseminated strikes in IoT systems. This is vital in the context of securing lightweight IoT networks since this kind of system is susceptible to a range of security issues, including jamming, masquerading, replaying, and eavesdropping, as well as resource limitations, including out-of-memory accesses, dangerous development tools, etc (Ma et al., 2019). Deep networks can significantly increase classification and prediction accuracy in these difficult tasks. Also, Deep learning can be used even in networks with minimal resources due to its lack of human choosing feature set, unsupervised pre-training, and compression capabilities (Wang, Liu, et al., 2018, Wang, Sheng, et al., 2018). In the proposed model we started building a lightweight intrusion detection by selecting the classifiers from both DL and ML. when it comes to general binary classification, the machine learning algorithms work head-to-head when compared to the

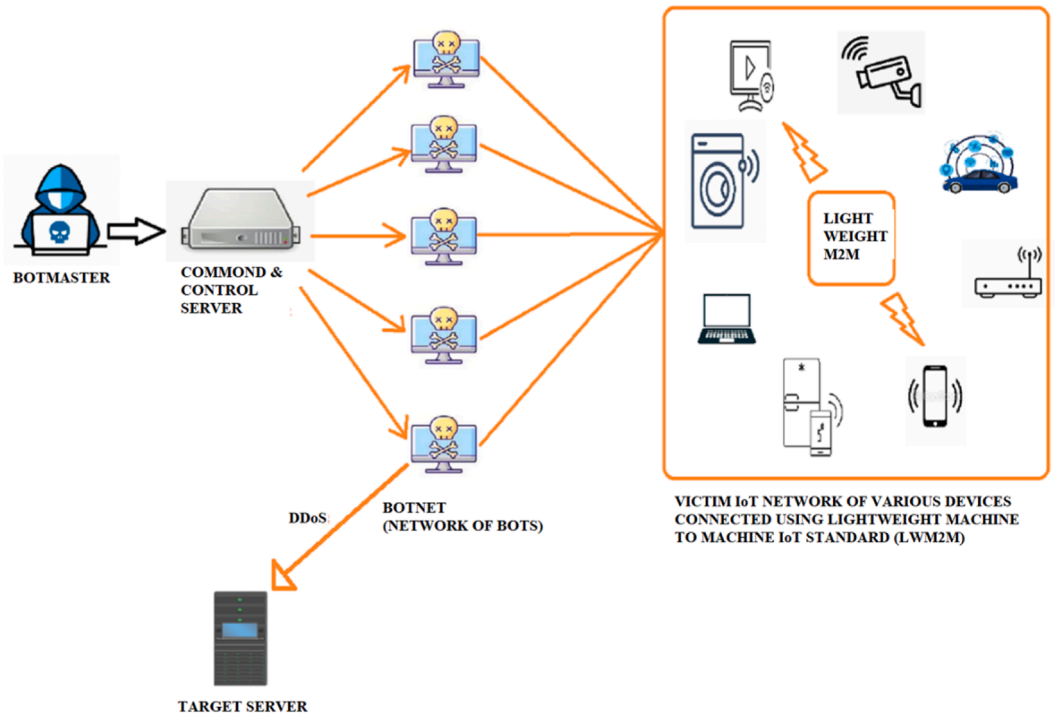

Fig. 1. DDoS attack model in lightweight IoT networks.

deep learning algorithms, the only difference lies in the structure and the nature of work. The proposed model is further defined below in Fig. 2.

The data frame that IDS (Intrusion Detection Systems) receive are diverse. Examples of such data include system logs, network traffic, application logs, binary or raw alerts, event traces, and threat data. An IDS gathers data structurally or interactively from many sources like applications or systems using typical logging or methods including disc, memory, packet, function, and code. Operations for data gathering and identification can often be conducted independently. A network, host, or both levels of behavior can be examined and found by an IDS. A static or dynamic infiltration can be identified when intrusion detection is employed in a network. The above model proposes some essential and fundamental steps by an intrusion detection system and carries out every cycle of the procedure unless the classification of DDoS attack traffic is detected and made distinct from the normal traffic. As the initial and the foremost step of an intrusion detection system is to differentiate normal traffic from malicious or infected traffic, the proposed intrusion detection is developed to achieve the same target discussed in section 3.1 below.

2.2. Data pre-processing

2.2.1. Handling missing values

Dealing with the missing values is one of the primary and initial steps in all data pre-processing techniques. In various data pre-processing techniques, many statistical factors are used to replace the missing value or instances like mean and standard deviation. Both the datasets have minimum missing values (Ravi & Mercy Shalinie, 2020; Saritas & Yasar, 2019) and are also available in clean versions i.e. without any missing values.

2.3. Handling categorical data

Many features in both datasets are containing a variety of string values, which are not quite suited for many of the machine learning classifiers. Python provides a variety of libraries for one hot encoding and label encoding using a label encoder from sklearn.preprocessing, eventually for handling categorical data and replacing the string values with numeric alternate values depending upon the classes of the data in that particular feature. Scikit learn is one of the firm libraries in python to perform one hot encoding/binarization or digitization of the categorical data. The technique used in the proposed model is Label Encoding from sklearn.preprocessing tool.

2.4. Data standardization

There are various features in both BOT-IoT and TON-IoT datasets, which are not in a model-friendly format. Features like the source and destination IP addresses are in the format of IP addresses. The requirement is a suitable normalization of the values after label encoding. In the proposed model standardization of the input data is performed using Standard Scaler.

3. Feature selection

To assess the effectiveness of the suggested model without experiencing overfitting problems, we began the inquiry by incorporating the maximum number of valuable features from the input dataset. But selecting the maximum number of features for a traditional ML classifier improves the chances of overfitting. When merged with the SelectFromModel meta-transformer, estimation by the tree (Tree-based estimators) can be used to determine the relevance of impurity-based features. which can then be used to eliminate unnecessary features. Much like Leevy et al. in their research have dropped useful features like source and destination IP addresses following the domain knowledge (Leevy et al., 2021) while Shurman et al. in their research use source and destination IP addresses as important features for isolating and blocking the sources from sending malicious traffic in a hybrid approach (Shurman et al., 2020). Koroniotis et al. in their research employ a Correlation Coefficient to harvest the ten best features from the BoT-IoT dataset (Koroniotis et al., 2019). Additionally, modern IDS are meant to be smart enough, somewhat closely matching the intellect of the botmaster where he needs to address the vulnerable IoT devices utilizing the destination IP address to deploy the bots. To address the requirement of

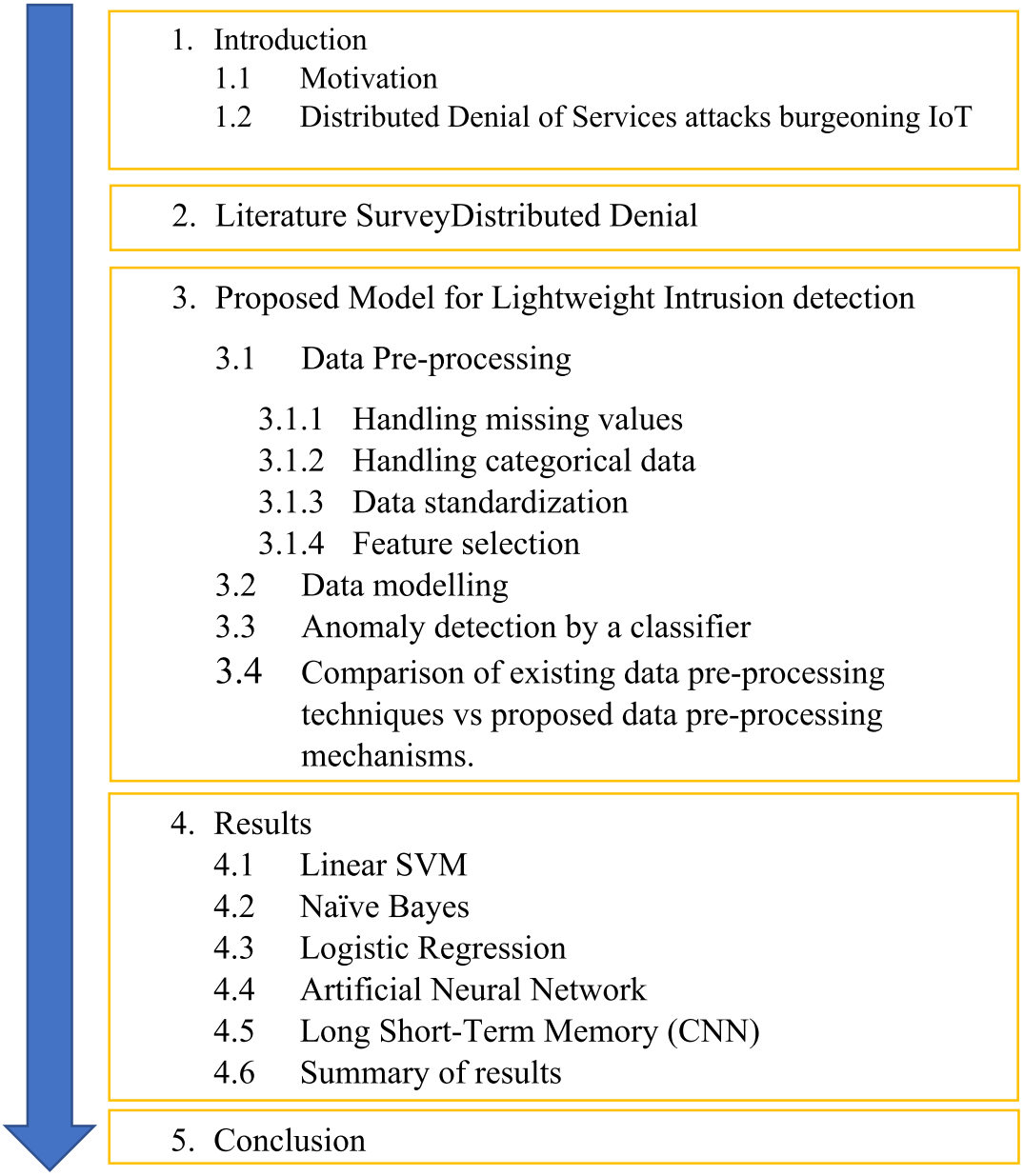

Fig. 2. Structure of the paper.

Table 1
Abbreviations.

DDoS	Distributed Denial of Services Attack
IoT	Internet of Things
AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
IDS	Intrusion Detection Systems
NIDS	Network Intrusion Detection Systems
DL-IDS	Deep Learning- Intrusion Detection Systems
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
NB	Naïve Bayes
SVM	Support Vector Machine
LSVM	Linear Support Vector Machine
LR	Logistic Regression
LSTM-CNN	Long Short-Term Memory-Convolutional Neural Network
MCT	Machine Controlled Traffic
HCT	Human Controlled Traffic
SDN	Software Defined Network
MQTT	Message Queuing Telemetry Transport
ROC	Receiver Operator Characteristic

including important features in the model building we have proposed a unique feature selection named ExtraTreeClassifier and selected twenty (20) important features from both datasets. The remaining features in the dataset are dropped. Extra Tree Classifier is a model-based approach for choosing variables based on a tree-based model evaluation to assess the significance of the attributes. Next, a meta estimator is chosen, which employs averages to increase prognosis accuracy and reduce overfitting. This estimation fits covering a broad range, or arbitrary decision trees, to varied data source sub-samples (Baby et al., 2021).

Additionally, each tree is provided a haphazard collection of several variables out from input data, based on which it should choose the most advantageous feature to divide the input data while using Gini Index. This spontaneous choice of features yields a number of de-correlated decision trees. Every attribute is given a scaled total decrease in the mathematical criteria (Gini Index if the Gini Index is applied in the design of the forest) that were used to decide which features should be divided. This value is known as the feature's Gini Importance. After grading all features in terms of their Gini Importance in lowest to the highest order, the user decides the top number of features, i.e; K-best features (Anon, 2019).

Information Gain will be the deciding factor in this case. Beginning by determining the data's entropy. The entropy formula is as follows:

$$\boldsymbol{Entropy}(\boldsymbol{S}) = \sum_{i=1} \boldsymbol{c} - \boldsymbol{pi}\log 2(\boldsymbol{pi}) \tag{1}$$

c – Total number of unique class labels
pi – Proportion of rows
output label

The formula to calculate Gain at the first tree of the forest is:

$$\boldsymbol{Gain}(\boldsymbol{S},\boldsymbol{A}) = \sum \boldsymbol{v}\varepsilon\boldsymbol{Values}(\boldsymbol{A})|\mathrm{Sv}| \ / \ |\mathrm{S}| \ \mathrm{Entropy} \ (\mathrm{Sv}) \tag{2}$$

A – Individual feature in the data frame.

The gain of each feature from the data frame is computed at each decision tree. For example

Gain (G_1) = (Entropy (S), A_1) represents the gain of the feature named A_1 at Decision tree D_1
Gain (G_2) = (Entropy (S), A_2) represents the gain of the feature named A_2 at Decision tree D_2
Gain (G_n) = (Entropy (S), A_n) represents the gain of the feature named A_n at Decision tree D_n

Henceforth Total gain for each feature is computed at the end of the process and the features with the maximum Gain are represented as important features. The relevance of the top 20 features from the BOTIoT and TONIoT datasets are shown using ExtraTreeClassifier from sklearn.ensemble. The important features from the aforementioned datasets are shown in Figs. 1 and 2.

Figs. 4 and 5 show the features that were extracted using the ExtraTreeClassifier feature selection test. Names of the features are listed vertically on the sides of Figs. 5 and 6, and a plotted graph illustrating their value on a scale from 0.00 to 0.30 is displayed. Figs. 3 and 4 exhibit the significant characteristics (features) of the TON-IoT network dataset as well as the BOT-IoT dataset.

3.1. Data modelling

The process of data modeling starts after slicing the data frame. The data frame is split into two sections 80 % for the training part and 20 % for testing for the classification model. The proposed model imported the sklearn.model_selection from train_test_split with testing the percentage of 20 % and random state = 0. For the model evaluation, 10-fold cross-validation on training data is used after the train test split and the classifier is fed with the training and testing samples.

3.2. Anomaly detection by a classifier

The output delivered by the classifier is a classification report depicting whether the instance belongs to a malicious/attack class or if the instance is the normal traffic towards a lightweight IoT network. In all tests carried out in the manuscript representation of the attack class by label and numeric one 'Class 1' and normal instances with zero 'Class 0'. The classifier also produces a ROC (Region of Convergence) Curve and accuracy percentage, classification report, and confusion metrics to analyze the outcomes of the classifier.

3.3. Comparison of existing data pre-processing techniques vs proposed data pre-processing mechanisms

This section provides a brief overview of the data-frame pre-processing techniques used in the existing literature. The variations in comparison with the proposed model can be visualized in the table given below: (See Table 3).

4. Results

In this manuscript, a set of five (5) experiments on the two different datasets are performed. In both the TONIoT network dataset (Moustafa et al., 2020) and BOTIoT dataset there are instances of various types of cyber-attacks (Koroniotis et al., 2019). The manuscript contains a series of distinguished tests Case 1, Case 2, Case 3, and Case 4. Case 1 and Case 3 contain the instances of all cyber-attacks in both datasets (BOTIoT and TONIoT network dataset) while Case 2 and Case 4 contain the instances of the DDoS (distributed Denial of Services) attack only concerning the normal traffic instances extracted from both datasets. The scenarios of cases 1, 2, 3 and 4 are further defined below:

Case 1: All attacks in the BOT-IoT dataset with SMOTE ().
Case 2: DDoS attack and Normal traffic instances in BOTIoT with SMOTE ().
Case 3: All attacks in the TON-IoT dataset.
Case 4: DDoS attack and Normal traffic instances in the TONIoT dataset.

The attack instances compared with the normal traffic flow in the BOTIoT dataset are fewer. There is the requirement to balance the attack class and normal traffic class in the data frame. SMOTE (Synthetic Minority oversampling Technique) is one of the infamous modules in

Table 2
Literature Survey.

Authors	Year	Proposed methodology	Algorithms used	Source data for Experimentation
(Zhang and Green 2015)	2015	Simulated lightweight DDoS detection algorithm in IoT	Contiki OS and COOJA network simulator	Network traffic is generated with simulated tools on the proposed testbed architecture
(Shurman, Khrais, and Yateem, 2020)	2020	Hybrid model: which combines a signature-based and an anomaly-based approach. A deep learning model is used LSTM from attacks	Simulated Hybrid approach and LSTM.	CICDDoS dataset
(Ferrag et al., 2021)	2021	The research focuses on the attacks with DDoS in the networks and proposes an IDS model using deep learning	CNN, deep neural networks, and RNN	CICDDoS dataset
(Shafiq et al., 2020)	2020	Novel feature engineering and selection metric approach named CorrAUC	Naïve Bayes Classifier, Decision tree (C4.5), Random Forrest and SVM	BoT-IoT dataset by UNSW University
(Doshi, Apthorpe, and Feamster, 2018)	2018	The datasets used for the experiments and investigation are BOT-IoT and the Network dataset of TON-IoT by the University of New South Wales Sydney (UNSW) Australia. DDoS	KNN, Random Forest, Decision tree, Support Vector Machines, and Deep Neural Networks	NDSE dataset (National Defense Science and Engineering Graduate Fellowship)
(Li et al., 2020)	2020	The authors propose a model with three sections for real-time volumetric DDoS detection. The model comprises of traffic processor with traffic monitoring, timestamps association, and a one-directional packet filter. Another two sections of the model are the Entropy calculator and the Alarm/ detection module.	Entropy-based	1999 DARPA, 2009 DARPA DDOS Dataset and the UNB CIC DDoS 2019
(Chen et al., 2020)	2020	In the paper, the authors start with the IoT model by installing sensors and harvesting sensor traffic at different locations. Then analyzing different DDoS attack types from the SDN blacklist malicious traffic. At last, the authors propose decision tree training for offline and online data.	Decision tree.	Dataset is generated by creating a heterogenous IoT network of sensors by the authors.
(Ma et al., 2019)	2019	The goal of the researchers is to highlight applications of deep learning for IoT.	–	–
(Cvitić, Peraković, Periša, & Botica, 2021)	2021	The authors' main contribution to the article is a prototype that is proposed, based on hardware categories that depend on the performance parameters of individual devices.	The main goal of the research is to distinguish Machine controlled traffic (MTC) and Human controlled traffic (HTC).	–
(Yu et al., 2012)	2012	As a partial solution, the authors suggest the Trust Management Helmet (TMH), a straightforward mitigation method that employs trustworthiness to differentiate between offenders and authorized personnel. Its most important conclusion is that, rather than identifying all attack requests when the DDoS attack pattern focuses at application tier, a server should give top priority to safeguarding the connectivity of reliable users. To arrange the service into clients' requirements, the trust of clients is assessed based on their past visits. Alsolicensing for non malicious user (even through NATs) and keeping data at user side. Using cryptography, the license is protected from fraud and advanced persistent threats.	–	–
(Doriguzzi-Corin et al., 2020)	2020	An online resource-constrained environment-friendly DL-based DDoS detection architecture that uses CNNs, which can quickly and efficiently understand how DDoS and other types of traffic behave.	CNN	ISCX2012, CIC2017 and CSECIC2018
(McDermott et al., 2018)	2018	Comparison of CNN-LSTM and Bidirectional LSTM-based models for botnet detection.	LSTM	A simulated Internet of Things network and a labeled dataset were created by the authors.
(Ravi & Mercy Shalinie, 2020)	2020	Semi-supervised deep extreme learningmachine (SDELM) to detect attack patterns.	Deep Neural network	Using the proposed test system structure, the UNB-ISCX dataset was established.
(Wani & Revathi, 2020)	2020	DDoS detection and avoidance in IoT networks using a Software-Defined Network (SDN)-based security mechanism	Micro Cluster outline detector for anomaly identification and Multi-layer Perceptron	–
(Jia et al., 2020)	2020	To discover, the findings cover diagnosing, contextualising, and preventing IoT DDoS threats.An perimeter IoT combative solution is FlowGuard.	LSTM	DDoS simulators BoNeSi, SlowHTTPTest, and CICDDoS2019
(Chen et al., 2022)	2022	To investigate the cooperative Stackelberg's game paradigm is formulated in this study to resist IoT payloads from Botnets.To calculate the optimum packet sampling rate (PSR) that will suitably deter potential attackers using the create contingency of the DDoS event.	–	–
(Cvitić et al., 2022)	2020	The researchers present a logistic tree modeling approach in Smart Home IoT devices.	LMT: Logistic model tree	To create their dataset, the researchers created a smart home IoT device testbed architecture and tools.
(Singh & Gupta, 2022)	2020	Expeditious surge and development of DDoS attacks.	–	–
	2021			Simulation setup.

(*continued on next page*)

Table 2 (*continued*)

Authors	Year	Proposed methodology	Algorithms used	Source data for Experimentation
(Mishra, Gupta, & Gupta, 2021)		The proposed model by the researchers performs entropy-based classification between traffic that is both harmful and lawful.	In a mininet emulator, At various assault levels, simulations are done with a POX controller and open flow switches.	
(Tewari & Gupta, 2020)	2021	Data security at the device level by using cryptographic procedures on RFID tags.	Introduced a new protocol that used timestamps and Bitwise operation	–
(Cvitić, Peraković, Periša, & Gupta, 2021)	2021	Logitboost model for classification.	Smart Home setup with 41 Smart Home IoT devices forming a network.	–
(Otoum, Liu, & Nayak, 2022)	2022	The suggested module increases detection accuracy by combining the Stacked-Deep Polynomial Network (SDPN) and the Spider Monkey Optimization algorithm (SMO).	Stacked-Deep Polynomial Network (SDPN)	NSL-KDD
(Mirsky et al., 2018)	2018	Kitsune's core algorithm (KitNET) uses a collection of neural networks to discern between conventional and anomalous traffic patterns	Autoencoders	–
(Bovenzi et al., 2020)	2020	H2ID, a two-stage hierarchical Network Intrusion Detection method, is suggested by the research. Invasion classification is handled out along with H2ID using soft-output algorithms, while oddity identification is accomplished via a cutting-edge, flexible methodology based on a Multidisciplinary Deep AutoEncoder (M2-DAE).	Multimodal deep autoencoders	–

python to deal with the imbalance of classes in the data frame. In the proposed model SMOTE class balancing technique is used in the BOTIoT dataset to oversample the minority class, which is normal traffic flow with the attack class. The instances of normal traffic instances are 477 concerning the 3,700,000 attack instances.

In every case, a set of five classifiers are used for determining the classification and prediction of the data frames. In each case, a Linear Support vector Machine, Naïve Bayes, Logistic Regression, Artificial Neural Network, and a CNN-LSTM (Long Short-Term Memory Convolutional Neural Network) are used making it a combination of deep learning and machine learning classifiers. The goal of conducting the research is to check and compare the effectiveness of the various classification and prediction algorithms. The results derived in all of the above tests are depicted in a table containing a Confusion matrix and a ROC (receiver operating characteristic) curve. The performance of all five classifiers is evaluated by the performance matrices containing Precision, Recall, F1-score, and, Support.

4.1. Linear SVM

A supervised machine learning method called SVM can be applied to both classification and regression problems. Even though we could also point out problems with regression, categorization fits the bill the best. The SVM method seeks to identify an *N*-dimensional space hyperplane that can classify data points. Using SVM in the given experiment for simple binary classification was the main goal to achieve the expected outcome. A linear Support Vector Classifier model is being used to train the SVM model. The form taken by linear SVM is given below:

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b}\sum_{i=1}^{n}\boldsymbol{wixi} + \boldsymbol{b} \tag{1}$$

The dimensionality of the feature vector $\boldsymbol{x}$ is given by **n**. The soft margin, which disincentivizes every trial by the hinge loss, is maximized to achieve the best weight vector **w** and bias **b** (Ladicky & Torr, 2019). The stated configuration has been effectively modified to assess the SVM model's optimal performance. The SVM classifier was initially trained using the default values, but it was later found that increasing the maximum iterations led to a longer training period.

We found that the precision decreased as the number of folds increased, but performed with 99 % accuracy with a 10-fold model evaluation. The input data frame is divided into the classes/groups class 0 and class 1, respectively. Fig. 6, Fig. 7, Fig. 8, and, Fig. 9 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis and a true positive rate on the vertical Y-axis (See Table 4).

4.2. Naïve Bayes

Using the Bayes theorem, naive Bayes classification is a Bayesian-based classification technique. NB determines the conditional probability of a class label given a dataset under the premise that the data are evenly disseminated. The Bayes theorem also provides a logical approach for computing this conditional probability given feature independence conjectures, such as the existence of each particular set as an independent feature of all other features in the dataset. Islam et al. (2007) state The Bayes theorem as:

$$\boldsymbol{P}\left(\frac{\boldsymbol{u}}{\boldsymbol{V}}\right) = \frac{\boldsymbol{P}\left(\frac{\boldsymbol{V}}{\boldsymbol{u}}\right)\boldsymbol{P}(\boldsymbol{u})}{\boldsymbol{P}(\boldsymbol{V})} \tag{1}$$

***P*(u/V)** – it is the Posterior probability of h if h is provided
***P*(V/u)** – is the probability of V given u as the maximum likelihood.
***P*(u)** – defines the prior probability of hypothesis 'u'.
***P*(V)** – defines the probability of training data used in the data frame (Evidence).

The Naive Bayes method, which is employed as a classifier in the aforementioned statement, is viewed as a strategy comprised of a family of algorithms that share the common principle that each pair of features under classification is autonomous of every one.

Fig. 10, Fig. 11, Fig. 12, and, Fig. 13 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis and a true positive rate on the vertical Y-axis (See Table 5).

4.3. Logistic Regression

Mathematically logistic regression can be defined as a model which forecasts P(Y = 1) as a function of X. In general, logistic regression pertains to binary logistic regression with binary data points either 0 or 1 determining the outcome into two groups, but it can perform multiple classifications as well for more than two desired outcomes (Brzezinski & Knafl, 1999).

Under the LR model with parameters as, we may derive the likelihood and log-likelihood of the data X, y.

$$\boldsymbol{h}\theta\,(\boldsymbol{x}) = \boldsymbol{\sigma}(\theta^{\boldsymbol{T}}\boldsymbol{X}) \tag{1}$$

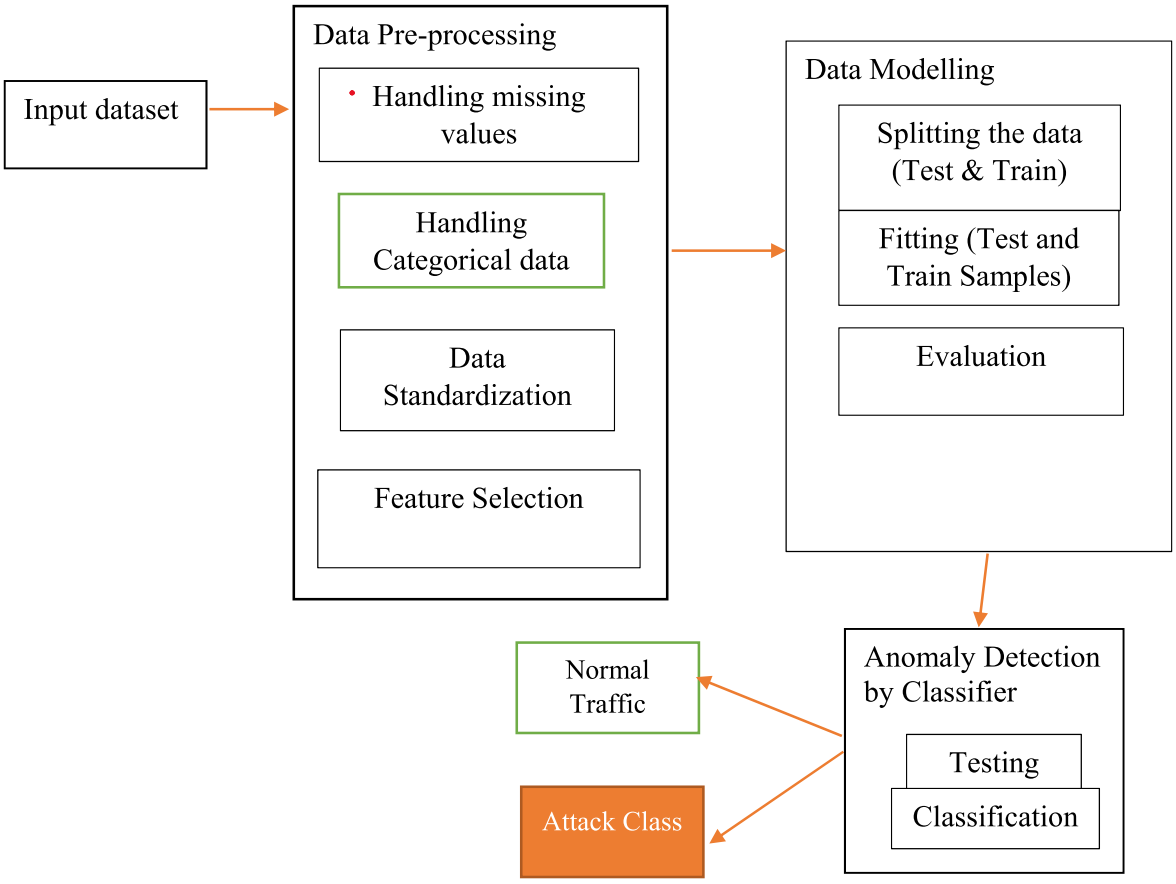

Fig. 3. Proposed model for intrusion detection.

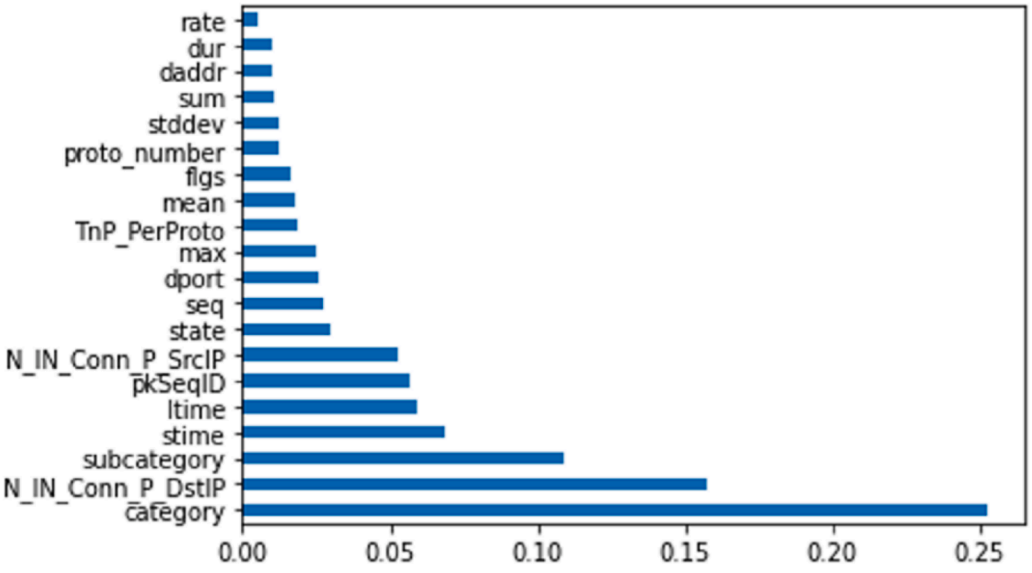

Fig. 4. Twenty best features in BOT-IoT.

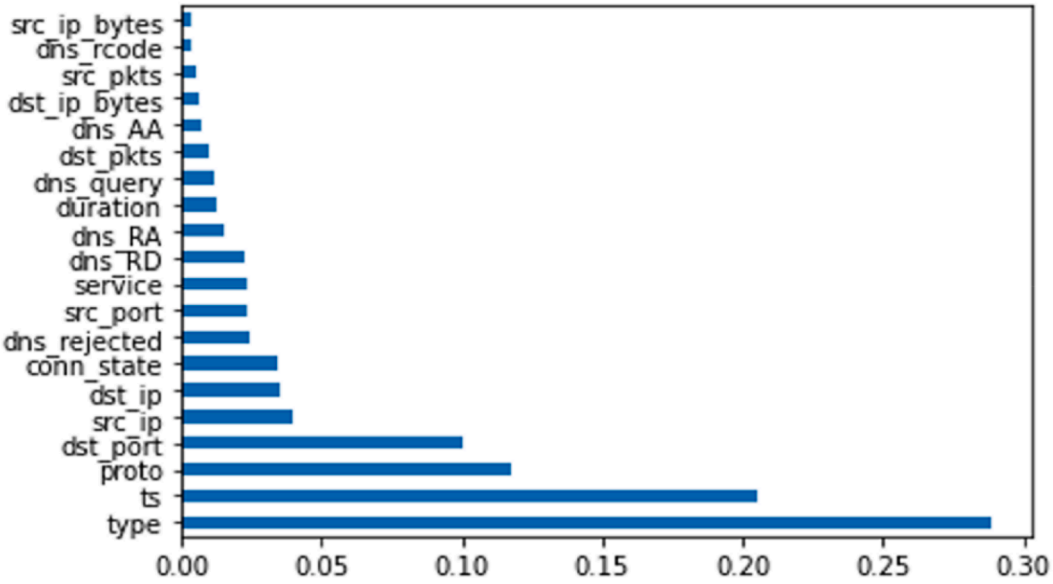

Fig. 5. Twenty best features in TON-IoT.

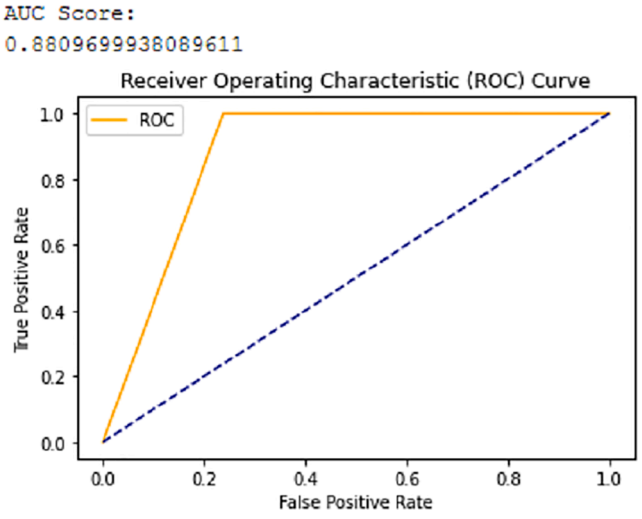

Fig. 6. ROC curve and accuracy percentage In BOT-IoT data-frame with all attacks.

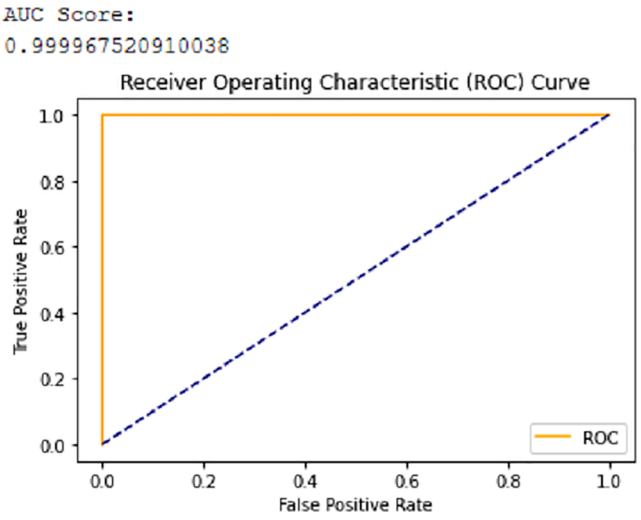

Fig. 7. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

Table 3
Comparison of existing data pre-processing techniques and proposed data pre-processing procedure.

Authors	Handling Missing values	Handling Categorical data	Standardization	Feature Selection
(Otoum et al., 2022)	Minkowski distance	–	–	SMO(sequential minimal optimization) Algorithm
(Gad, Nashat, & Barkat, 2021)	Imputation	One Hot Encoding	Min-Max Scaler	Chi-Square Test
(Laghrissi et al., 2021)	–	–	–	Principle ComponentAnalysis (PCA)
(Pokhrel, Abbas, and Aryal, 2021)	Dropout technique	Type conversion/Transformation	Min-Max Scaler	Chi-Square Test
Proposed model	–	Label Encoding	Standard Scaler	Extra Tree Classifier

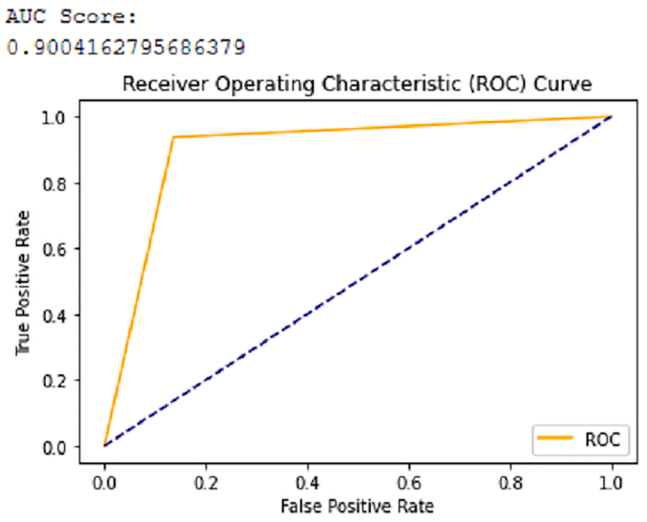

Fig. 8. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

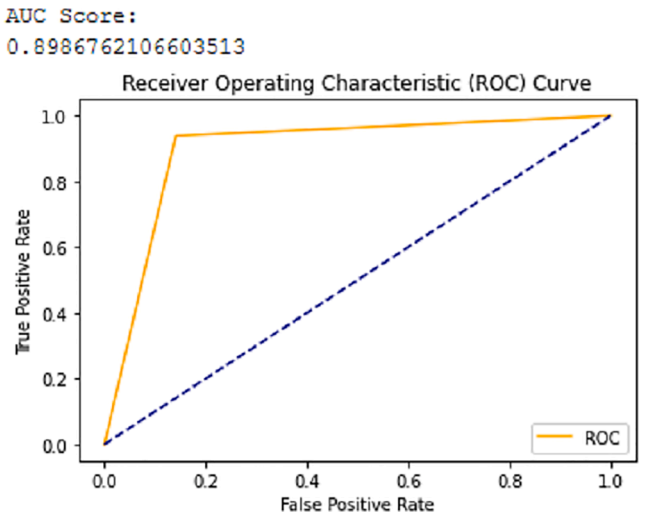

Fig. 9. ROC curve and accuracy percentage in TON-IoT data-frame with DDoS attack.

h - hypothesis,
x - input feature vector.
θ– Logistic Regression parameter.
σ – sigmoid function/threshold function.
r is the threshold definition sigmoid function.

$$\boldsymbol{\sigma}(\boldsymbol{r}) = \frac{1}{1+\boldsymbol{e}^{-\boldsymbol{r}}} \tag{2}$$

where **r** is the term of $\theta^{\mathbf{T}}\mathbf{x}$ from the previous equation, and the result is somewhere in the middle (0:1) [26].

Fig. 10, Fig. 11, Fig. 12, and, Fig. 13 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis and a true positive rate on the vertical Y-axis.

Fig. 14, Fig. 15, Fig. 16, and, Fig. 17 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis a and true positive rate on the vertical Y-axis (See Table 6).

4.4. Artificial Neural Networks (ANNs)

Before a decent model is developed, there are many trials and errors. Using the Dense class in Keras, the expectations are to create a Fully Connected network structured-forward neural network that transfer the weight values of each neuron as output to the following layer after processing the inputs from neurons in the previous layer (Saritas & Yasar, 2019). The Neuron is the dense layer's first argument. The activation argument can be used to set the activation function. In this research, the Rectified Linear Unit as the activation function is utilized. Other solutions exist, such as Sigmoid or TanH, but RELU is a more generalized and superior option over the rest of the options. After the definition of the model, a compilation of the model is required. For a general-purpose ANN model compilation, Tensorflow is employed in the tests. The process of compiling parameters for model training and predictions is known as compilation. There is the requirement to define a loss function that will be used to calculate the weights for each layer. The optimizer changes the learning rate and cycles through different weight sets. As the loss function in this situation, we'll utilize Binary Cross Entropy. In the proposed model ADAM is used, which is a fast stochastic gradient descent technique, for the optimizer. The other

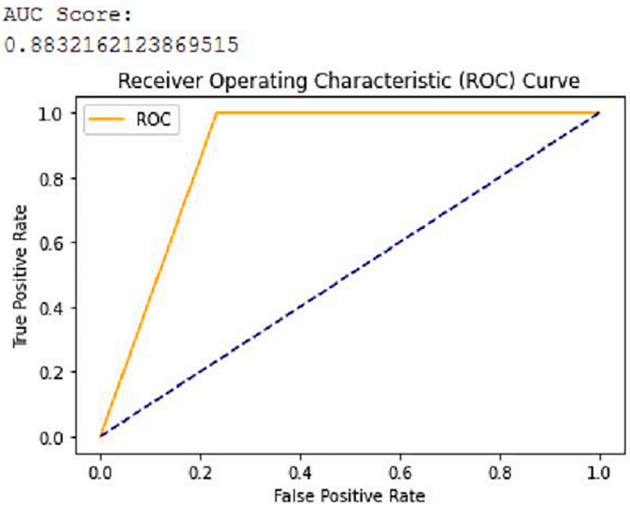

Fig. 10. ROC curve and accuracy percentage In BOT-IoT data-frame with all attacks.

Table 4
Confusion matrix of data-frame with all attacks (Case 1 and Case 3) and DDoS attacks only (Case 2 and Case 4).

Case 1(All attacks)			Case 2 (DDoS)		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	117,982	349,148	Class 0 (Normal)	771,574	0
Class 1 (Attack)	147	1,467,189	Class 1 (Attack)	50	769,676
Case 3 (All attacks)			**Case 4 (DDoS)**		
Class 0 (Normal)	51,684	8158	Class 0 (Normal)	51,507	8466]
Class 1 (Attack)	2034	30,333	Class 1 (Attack)	1982	30,254

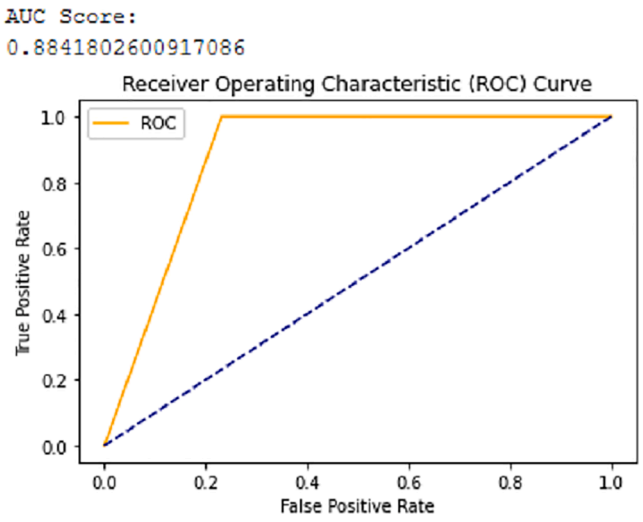

Fig. 11. ROC curve and accuracy percentage In BOT IoT data-frame with DDoS attack.

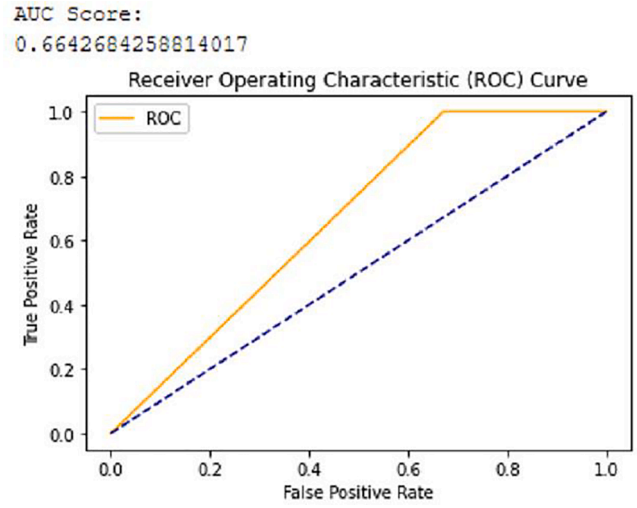

Fig. 12. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

requirements of parameters for building decent classification models are the number of dense units used, activation function type, and kernel initializer.

Fig. 18, Fig. 19, Fig. 20 and Fig. 21 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis and a true positive rate on the vertical Y-axis (See Table 7).

4.5. Long short term memory (LSTM-RNN)

The vanishing gradient issue that bedevils conventional recurrent neural networks is addressed by LSTM RNNs by including gating functions throughout its state kinetics. A hidden vector (**h**) and a memory vector (**m**) are stored in an LSTM for synchronizing transitions and outputs throughout each time step. (Karim, Majumdar, & Darabi, 2019),

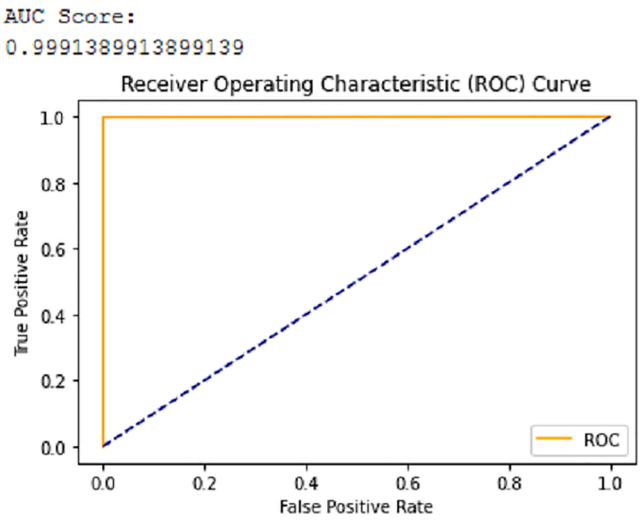

Fig. 13. ROC curve and accuracy percentage in TON-IoT data-frame with DDoS attack.

The Model has trained with a batch size equal to 10 on the best features of both the TONIoT and BOTIoT datasets. The neural network has ten input neurons (the same number as the characteristics in the first layer), intermediate (hidden) layers with ten, ten, five neurons, and one output neuron for binary classification recurrent neural networks have a vanishing gradient problem, thus large short-term memory recurrent neural networks offer an improvement (Karim et al., 2019). The ROC graphs with the accuracy percentage for both the data frames are given below in the graphs. The graph on the left-hand side is Case 1 and Case 3 and the graph on the right-hand side for Case 2 and Case 4.

Fig. 22, Fig. 23, Fig. 24 and Fig. 25 are ROC curves plotted with the accuracy percentage of each test at the top of each figure. With a false positive rate on the horizontal X-axis and a true positive rate on the vertical Y-axis (See Table 8).

4.6. Summary of the results

The above table shows the reports derived from each classifier and with outcomes in the form of parameters namely Accuracy, precision, recall, and F1 score. The formula to compute all the four parameters of the classification procedure are given below:

1. Accuracy Percentage $= \left(\frac{\text{True positives+True Negatives}}{\text{True positives+True Negatives+False Positives+False negative}}*100\right)$
2. Recall $= \left(\frac{\text{True positive}}{\text{True positive+False negative}}*100\right)$
3. Precision $= \left(\frac{\text{True positives}}{\text{True positives + False Positives}}*100\right)$
4. F1 Score $= \frac{(2*\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})}$

Table 9 has the name of two cases used in the investigation on the BOT_IoT dataset in its first column, followed by the name of classifiers in the second column. In the third column of the table is the accuracy percentage of every classifier in classifying the attack labels over the normal traffic labels. The fourth column is the class label which presents 1 as attack and 0 as normal. The remaining four columns are Precision, and Recall. F1-Score and Support indicate the performance of each

Table 5
Confusion matrix of data-frame with all attacks (Case 1 and Case 3) and DDoS attacks only (Case 2 and Case 4).

Case 1(All attacks)			Case 2 (DDoS)		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	1,124,484	342,646	Class 0 (Normal)	592,847	178,727
Class 1 (Attack)	28	11,467,278	Class 1 (Attack)	0	769,726
Case 3 (All attacks)			**Case 4 (DDoS)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	19,664	40,178	Class 0 (Normal)	59,935	0
Class 1 (Attack)	2	32,365	Class 1 (Attack)	7	4078

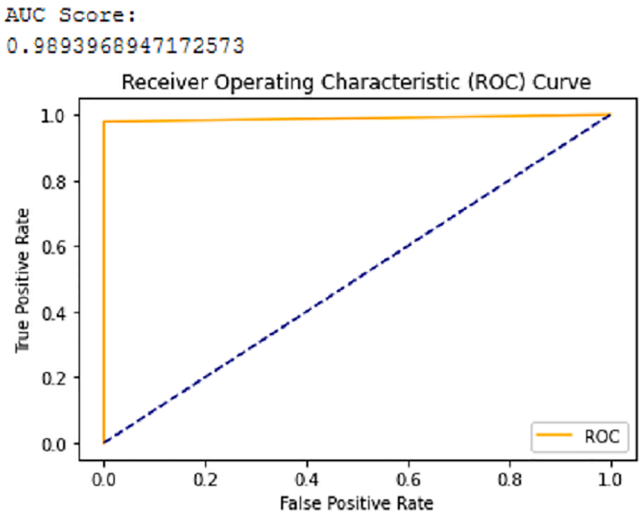

Fig. 14. ROC curve and accuracy percentage in BOT-IoT data-frame with all attacks.

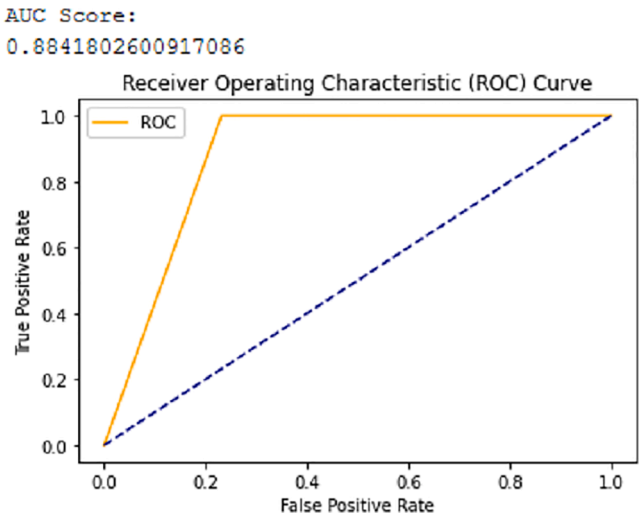

Fig. 15. ROC curve and accuracy percentage In BOT IoT data-frame with DDoS attack.

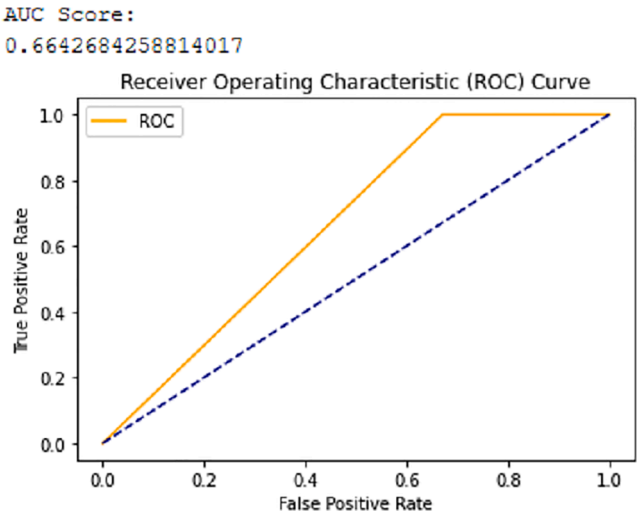

Fig. 16. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

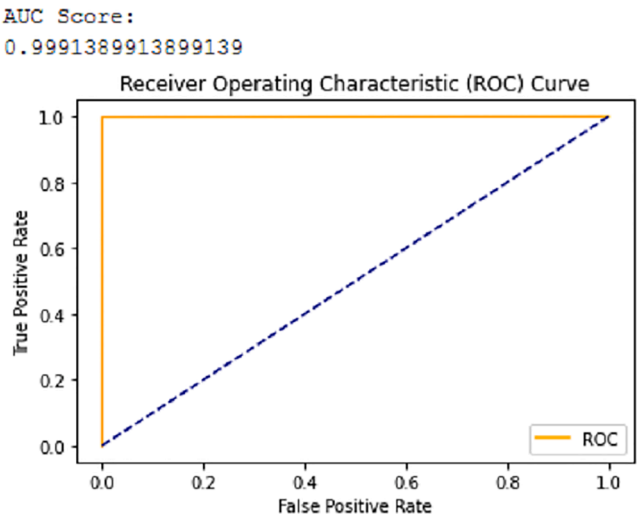

Fig. 17. ROC curve and accuracy percentage In TON-IoT data-frame with DDoS attack.

Table 6
Confusion matrix of data-frame with all attacks (Case 1 and 3) and DDoS attacks only (Case 2 and 4).

Case 1(All attacks)			Case 2 (DDoS)		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	1,467,130	0	Class 0 (Normal)	767,815	3759
Class 1 (Attack)	31,116	1,436,190	Class 1 (Attack)	2767	766,959
Case 3 (All attacks)			**Case 4 (DDoS)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	19,664	40,178	Class 0 (Normal)	59,935	0
Class 1 (Attack)	2	32,365	Class 1 (Attack)	7	4058

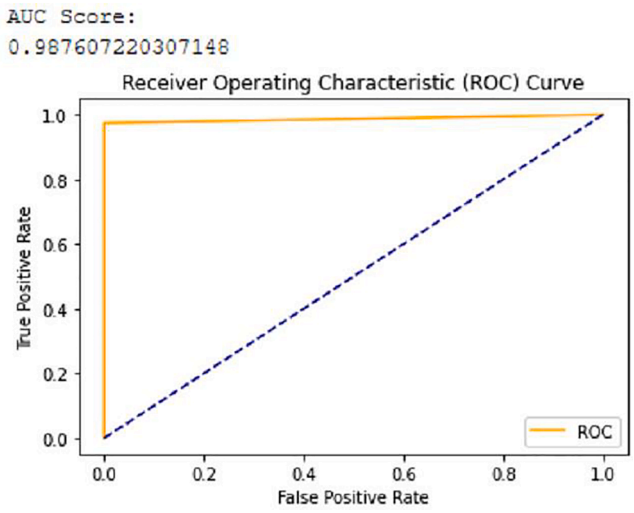

Fig. 18. ROC curve and accuracy percentage In BOT-IoT data-frame with all attacks.

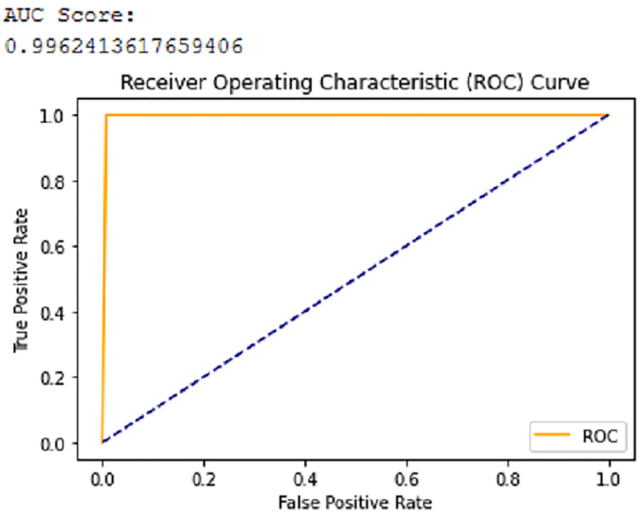

Fig. 20. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

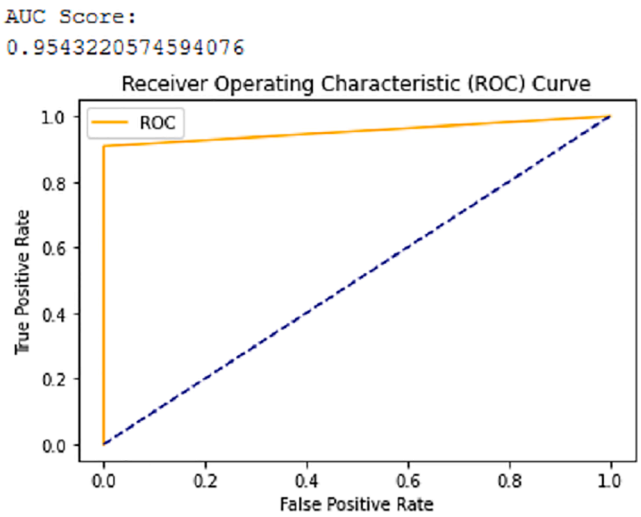

Fig. 19. ROC curve and accuracy percentage In BOT IoT data-frame with DDoS attack.

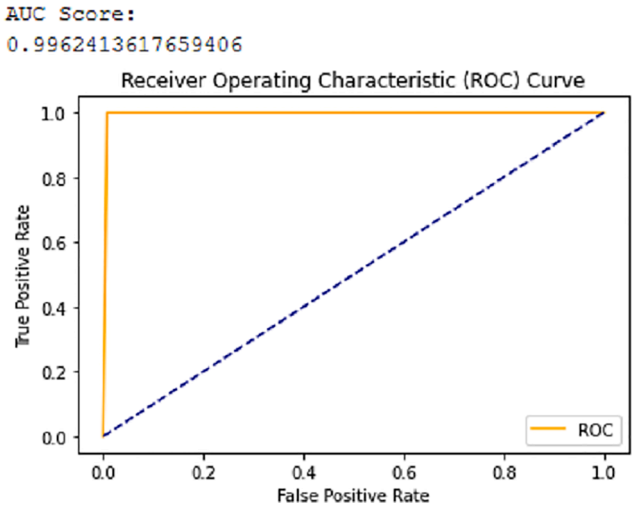

Fig. 21. ROC curve and accuracy percentage in TON-IoT data-frame with DDoS attack.

Table 7
Confusion matrix of data-frame with all attacks (Case 1 and 3) and DDoS attacks only (Case 2 and 4).

Experiment 1(All attacks)			Experiment 2 (DDoS)		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	1,467,130	0	Class 0 (Normal)	771,574	3759
Class 1 (Attack)	36,368	1,430,938	Class 1 (Attack)	70,319	699,407
Experiment 3 (All attacks)			**Experiment 4 (DDoS)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	59,394	448	Class 0 (Normal)	59,967	**6**
Class 1 (Attack)	1	32,366	Class 1 (Attack)	11	32,225

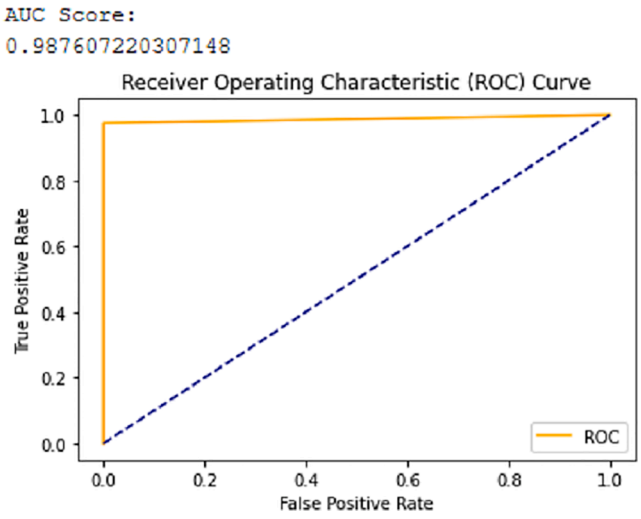

Fig. 22. ROC curve and accuracy percentage in BOT-IoT data-frame with all attacks.

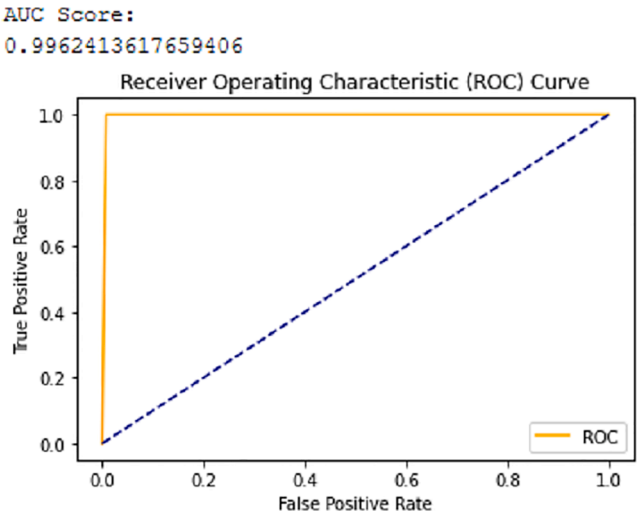

Fig. 24. ROC curve and accuracy percentage in TON-IoT data-frame with all attacks.

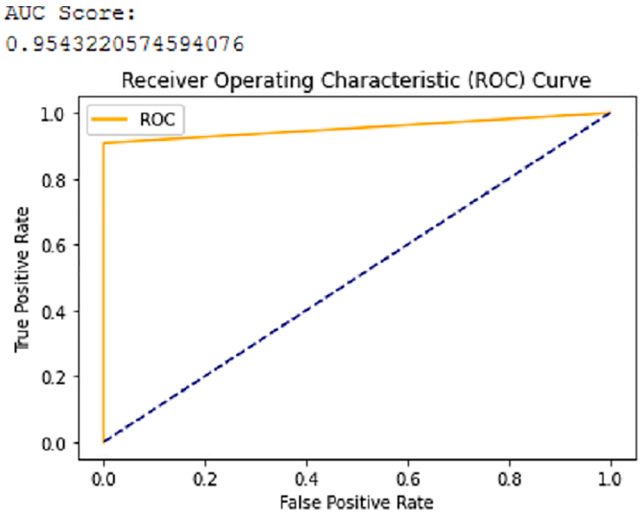

Fig. 23. ROC curve and accuracy percentage In BOT IoT data-frame with DDoS attack.

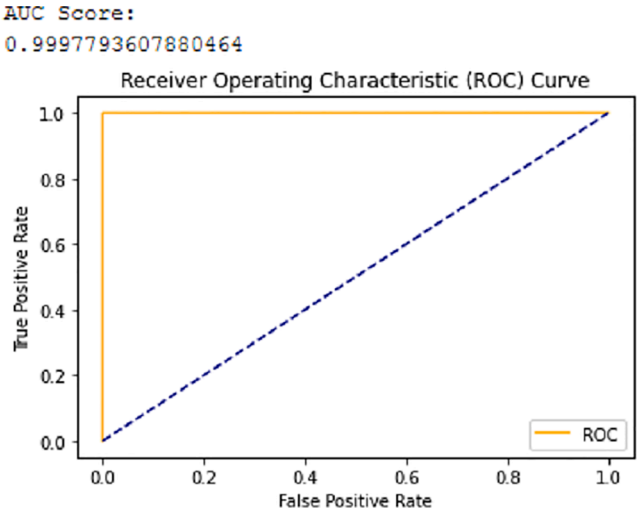

Fig. 25. ROC curve and accuracy percentage In TON IoT data-frame with DDoS attack.

Table 8
Confusion matrix of data-frame with all attacks (Case 1 and 3) and DDoS attack only (Case 2 and 4).

Experiment 1(All attacks)			Experiment 2 (DDoS)		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	1,467,130	0	Class 0 (Normal)	771,574	0
Class 1 (Attack)	36,368	1,430,938	Class 1 (Attack)	70,319	699,407
Experiment 3 (All attacks)			**Experiment 4 (DDoS)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	59,394	448	Class 0 (Normal)	59,967	6
Class 1 (Attack)	1	32,366	Class 1 (Attack)	11	32,225

Table 9
Description of the accuracy, precision, F1-score, and support obtained in experiment 1 and experiment 2 on the BOT-IoT dataset.

		Accuracy	Class Label	Precision	Recall	F1-Score	Support
Case 1: BOTIoT dataset with all attacks	Linear SVC	88 %	0	1.00	0.76	0.86	1,467,130
			1	0.81	1.00	0.89	1,467,306
	Naïve Bayes	88 %	0	1.00	0.77	0.87	1,467,130
			1	0.81	1.00	0.99	1,467,306
	Logistic Regression	98 %	0	0.98	1.00	0.99	1,467,130
			1	1.00	0.98	0.99	1,467,306
	ANN	98 %	0	0.98	1.0	0.99	1,467,130
			1	1.00	0.98	0.99	1,467,306
	LSTM	98 %	0	0.98	1.00	0.99	1,467,130
			1	1.00	0.98	0.99	1,467,306
Case 2: BOTIoT dataset with DDoS attack instances.	Linear SVM	99 %	0	1.0	1.0	1.0	771,574
			1	1.0	1.0	1.0	769,726
	Naïve Bayes	88 %	0	1.00	0.77	0.87	771,574
			1	0.81	1.00	0.90	769,726
	Logistic Regression	99 %	0	1.00	1.00	1.00	771,574
			1	1.00	1.00	1.00	769,726
	ANN	95 %	0	0.92	1.00	0.96	771,574
			1	1.00	0.91	0.95	769,726
	LSTM	95 %	0	0.92	1.00	0.96	771,574
			1	1.00	0.91	0.95	769,726

Table 10
Description of the Accuracy, Precision, F1-Score, and Support obtained in Case 3 and Case 4 on the TON-IoT dataset.

		Accuracy	Class Label	Precision	Recall	F1-Score	Support
Case 3:	Linear SVM	90 %	0	0.96	0.86	0.91	59,842
TON-IoT dataset with all attacks			1	0.79	0.94	0.86	32,367
	Naïve Bayes	66 %	0	1.00	0.33	0.49	59,842
			1	0.45	1.0	0.62	32,367
	Logistic Regression	66 %	0	1.0	0.33	0.49	59,842
			1	0.45	1.0	0.62	32,367
	ANN	99 %	0	1.0	0.99	1.0	59,842
			1	0.99	1.00	0.99	32,367
	LSTM	99 %	0	1.0	0.99	1.0	59,842
			1	0.99	1.0	0.99	32,367
Case 4:	Linear SVM	88 %	0	0.96	0.86	0.91	59,973
TON-IoT dataset with DDoS attack instances.			1	0.78	0.94	0.85	32,236
	Naïve Bayes	99 %	0	1.0	1.0	1.0	59,935
			1	1.0	1.0	1.0	4065
	Logistic Regression	99 %	0	1.0	1.0	1.0	59,935
			1	1.0	1.0	1.0	4065
	ANN	99 %	0	1.0	1.0	1.0	59,973
			1	1.0	1.0	1.0	32,236
	LSTM	99 %	0	1.0	1.0	1.0	59,973
			1	1.0	1.0	1.0	32,236

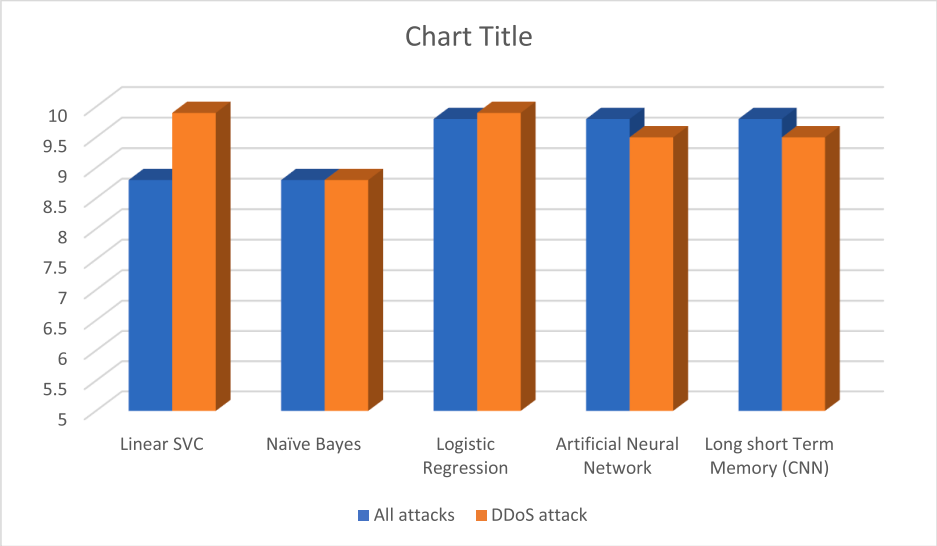

Fig. 26. Accuracy of Linear SVC, Naïve Bayes, Logistic Regression, ANN, and CNN-LSTM obtained in Experiment 1 and Experiment 2 (BOT-IoT).

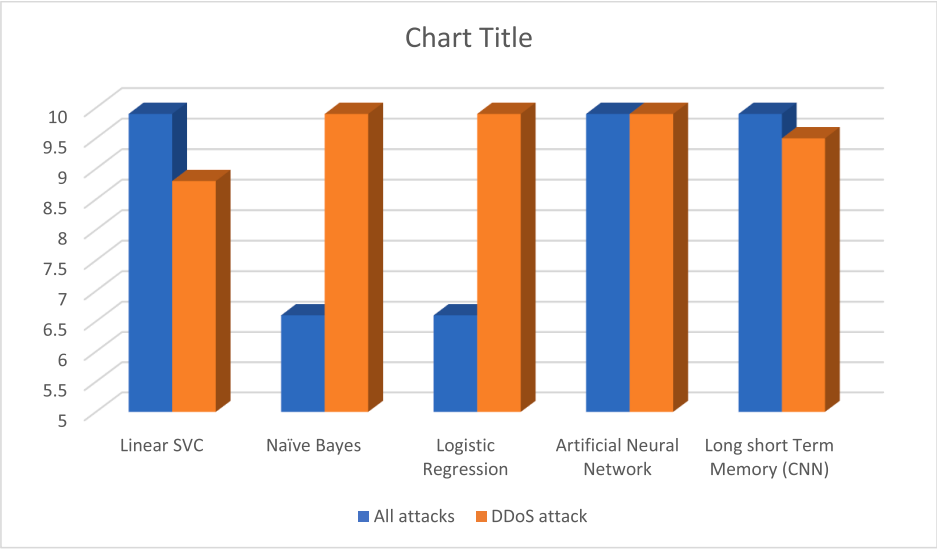

Fig. 27. Accuracy of Linear SVC, Naïve Bayes, Logistic Regression, ANN, and CNN-LSTM obtained in Case 3 and Case 4 (TON-IoT).

classifier.

Similar to Table 9, Table 10's first column lists the names of the two cases that were investigated using the TON IoT network dataset, with the names of the classifiers provided in the second column. The accuracy rate of each classifier in distinguishing between attack labels and regular traffic labels is shown in the third column of the table. The class designation, presented as 1 for attack and 0 for normal in the fourth column, lists these two values. Precision and Recall are among the final four columns. Each classifier's effectiveness is shown by its F1-Score and Support.

The bar graphs in Figs. 26 and 27 show the accuracy scores that each classifier achieved. The graphs' scale runs from 5 (representing 50 %) to 10 (representing 100 %).

5. Conclusion and future directions

IoT security requires the identification of assaults in the network to keep an eye on and restrict undesired traffic flows. Many academics have presented ML method models to restrict attack traffic flows in the IoT network. Several machine learning algorithms, on the other hand, are prone to misclassifying predominantly harmful traffic flows due to insufficient feature selection. However, one important issue that needs to be researched further is how to pick useful features for accurate malicious traffic identification in IoT networks. A new framework model has been presented for this purpose. The goal of the research is to employ a hybrid approach using a set of machine learning algorithms along with a set of deep learning models, to compare the effectiveness of the proposed algorithms as suitable classifiers for the detection of DDoS

and other common cyber-attacks in the lightweight IoT networks. However deep learning-based IoT IDS can be assisted by a comprehensive pre-processing and meaningful scaling down of features (feature selection) in the input traffic data could reduce the computational overhead at the device level reserved for the future. Future efforts will focus on examining multiple concurrent effects of assaults and broadening the application of this IDS system to cover other forms of intrusions where the impact of changing traffic intensity is less obvious or is concealed by attackers. The proposed model is developed as a passive intrusion detection system suitable for Software Defined Network or intermediate Fog layer of the IoT network. The future direction to upgrade the proposed model using Extensible Artificial Intelligence is (a) On the SDN controller, further botnet detection and mitigation measures can be employed to identify botnets and stop hosts from delivering them via networks in real-time scenarios. This will enable effective botnet attack mitigation on the server by monitoring traffic flow on all linked hosts. (b) to employ inclusive and interpretable machine learning and deep learning models for automatic feature selection techniques (selecting important features and dropping redundant features) from the input data frame, while developing a lightweight Intrusion Detection model (c) considering the fact of resource-constrained IoT devices and its evolution to the Internet of Everything (IoE) the proposed model in near future could be updated and implemented in the network side or at Fog layer instead of Software Defined Networks, as long as the computation requirements of the model are matched by the IoT network.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Angrishi, K. (2017). Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets.

Anon. (2019). ML | Extra tree classifier for feature selection. *GeeksforGeeks*. Retrieved 12 August 2022 (https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/).

Anon. (n.d.) Mirai Botnet DDoS attack: What is the Mirai Botnet? | Avast. Retrieved 12 August 2022 (https://www.avast.com/c-mirai).

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Alex Halderman, J., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C. Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai Botnet. Pp. 1093–1110.

Baby, D., Sujitha, D., Jude, H., & Anishin, M. (2021). Leukocyte classification based on feature selection using extra trees classifier: A transfer learning approach. *Turkish Journal of Electrical Engineering and Computer Sciences, 29*(8), 2742–2757. https://doi.org/10.3906/elk-2104-183

Bovenzi, G., Giuseppe A., Domenico C., Valerio P., & Antonio Pescapé. (2020). A Hierarchical hybrid intrusion detection approach in IoT scenarios. Pp. 1–7 in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*.

Brzezinski, J. R., & Knafl, G. J. (1999). Logistic regression modeling for context-based classification. Pp. 755–59 in *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials, 21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Chen, X., Xiao, L., Feng, W., Ge, N., & Wang, X. (2022). DDoS defense for IoT: A Stackelberg game model-enabled collaborative framework. *IEEE Internet of Things Journal, 9*(12), 9659–9674. https://doi.org/10.1109/JIOT.2021.3138094

Chen, Y.-W., Sheu, J.-P., Kuo, Y.-C., & Van Cuong, N. (2020). Design and implementation of IoT DDoS attacks detection system based on machine learning. *European Conference on Networks and Communications (EuCNC), 2020*, 122–127. https://doi.org/10.1109/EuCNC48522.2020.9200909

Cvitić, I., Perakovic, D., Gupta, B. B., & Choo, K.-K. (2022). Boosting-based DDoS detection in internet of things systems. *IEEE Internet of Things Journal, 9*(3), 2109–2123. https://doi.org/10.1109/JIOT.2021.3090909

Cvitić, I., Peraković, D., Periša, M., & Botica, M. (2021). Novel approach for detection of IoT generated DDoS traffic. *Wireless Networks, 27*(3), 1573–1586. https://doi.org/10.1007/s11276-019-02043-1

Cvitić, I., Peraković, D., Periša, M., & Gupta, B. (2021). Ensemble machine learning approach for classification of IoT devices in smart home. *International Journal of Machine Learning and Cybernetics, 12*(11), 3179–3202. https://doi.org/10.1007/s13042-020-01241-0

Donno, D.e., Michele, N. D., Giaretta, A., & Spognardi, A. (2018). DDoS-capable IoT malwares: Comparative analysis and Mirai investigation. *Security and Communication Networks, 2018*, e7178164.

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martínez-del-Rincón, J., & Siracusa, D. (2020). Lucid: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Transactions on Network and Service Management, 17*(2), 876–889. https://doi.org/10.1109/TNSM.2020.2971776

Doshi, K., Yilmaz, Y., & Uludag, S. (2021). Timely detection and mitigation of stealthy DDoS attacks via IoT networks. *IEEE Transactions on Dependable and Secure Computing, 18*(5), 2164–2176. https://doi.org/10.1109/TDSC.2021.3049942

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer internet of things devices. In *2018 IEEE security and privacy workshops (SPW)* Pp. 29–35.

Ferrag, M. A., Shu, L., Djallel, H., & Choo, K.-K. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics, 10*(11), 1257. https://doi.org/10.3390/electronics10111257

Gad, A. R., Nashat, A. A., & Barkat, T. M. (2021). Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access, 9*, 142206–142217.

Islam, M. J., Jonathan Wu, Q. M., Ahmadi, M., & Sid-Ahmed, M. A. (2007). Investigating the performance of Naive-Bayes classifiers and K-nearest neighbor classifiers. In *2007 International conference on convergence information technology (ICCIT 2007)* Pp. 1541–46.

Jia, Y., Zhong, F., Alrawais, A., Gong, B., & Cheng, X. (2020). FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet of Things Journal, 7*(10), 9552–9562. https://doi.org/10.1109/JIOT.2020.2993782

Karim, F., Majumdar, S., & Darabi, H. (2019). Insights into LSTM fully convolutional networks for time series classification. *IEEE Access, 7*, 67718–67725. https://doi.org/10.1109/ACCESS.2019.2916828

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems, 100*, 779–796. https://doi.org/10.1016/j.future.2019.05.041

Ladicky, L., & Torr, P. H. S. (2019). Locally linear support vector machines.

Laghrissi, F. E., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data, 8*(1), 65. https://doi.org/10.1186/s40537-021-00448-4

Leevy, J. L., Hancock, J., Khoshgoftaar, T. M., Peterson, J. M. (2021). An easy-to-classify approach for the Bot-IoT dataset. In *2021 IEEE third international conference on cognitive machine intelligence (CogMI)*. Pp. 172–79.

Li, J., Liu, M., Xue, Z., Fan, X., & He, X. (2020). Rtvd: A real-time volumetric detection scheme for Ddos in the internet of things. *IEEE Access, 8*(9000545), 36191–36201. https://doi.org/10.1109/ACCESS.2020.2974293

Ma, X., Yao, T., Menglan, H., Dong, Y., Liu, W., Wang, F., & Liu, J. (2019). A survey on deep learning empowered IoT applications. *IEEE Access, 7*, 181721–181732. https://doi.org/10.1109/ACCESS.2019.2958962

McDermott, C. D., Majdani, F., & Petrovski, A. V. (2018). Botnet detection in the internet of things using deep learning approaches. In *2018 international joint conference on neural networks (IJCNN)* Pp. 1–8.

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection.

Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommunication Systems, 77*(1), 47–62. https://doi.org/10.1007/s11235-020-00747-w

Moustafa, N., Keshk, M., Debie, E., & Janicke, H. (2020). Federated TON_IoT windows datasets for evaluating AI-based security applications.

Nascita, A., Montieri, A., Aceto, G., Ciuonzo, D., Persico, V., & Pescapé, A. (2021). XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures. *IEEE Transactions on Network and Service Management, 18*(4), 4225–4246. https://doi.org/10.1109/TNSM.2021.3098157

Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: A deep learning-based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies, 33*(3), e3803. https://doi.org/10.1002/ett.3803

Pokhrel, S., Abbas, R., & Aryal, B. (2021). IoT security: Botnet detection in IoT using machine learning.

Ravi, N., & Mercy Shalinie, S. (2020). Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. *IEEE Internet of Things Journal, 7*(4), 3559–3570. https://doi.org/10.1109/JIOT.2020.2973176

Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. *International Journal of Intelligent Systems and Applications in Engineering, 7*(2), 88–91. https://doi.org/10.18201//ijisae.2019252786

Shafiq, M., Tian, Z., Bashir, A. K., Xiaojiang, D.u., & Guizani, M. (2020). CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine learning techniques. *IEEE Internet of Things Journal*, 1.

Shurman, M., Khrais, R., & Rahman Yateem, A. (2020). DoS and DDoS attack detection using deep learning and IDS. *International Arab Journal of Information Technology, 17*, 655–661. https://doi.org/10.34028/iajit/17/4A/10

Singh, A., & Gupta, B. B. (2022). Distributed Denial-of-Service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: Issues, challenges, and future research directions. *International Journal on Semantic Web and Information Systems (IJSWIS), 18*(1), 1–43. https://doi.org/10.4018/IJSWIS.297143

Tewari, A., & Gupta, B. B. (2020). Secure timestamp-based mutual authentication protocol for IoT devices using RFID tags. *International Journal on Semantic Web and Information Systems (IJSWIS), 16*(3), 20–34. https://doi.org/10.4018/IJSWIS.2020070102

Vishwakarma, R., & Jain, A. K. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication Systems, 73*(1), 3–25. https://doi.org/10.1007/s11235-019-00599-z

Wang, W., Liu, J., Pitsilis, G., & Zhang, X. (2018). Abstracting massive data for lightweight intrusion detection in computer networks. *Information Sciences, 433–434*, 417–430. https://doi.org/10.1016/j.ins.2016.10.023

Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2018). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access, 6*, 1792–1806. https://doi.org/10.1109/ACCESS.2017.2780250

Wani, A., & Revathi, S. (2020). DDoS detection and alleviation in IoT using SDN (SDIoT-DDoS-DA). *Journal of The Institution of Engineers (India): Series B, 101*(117–28). https://doi.org/10.1007/s40031-020-00442-z

Yu, J., Fang, C., Lu, L., & Li, Z. (2012). A lightweight mechanism to mitigate application layer DDoS attacks.

Zhang, C., & Green, R. (2015). Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network. In *Proceedings of the 18th Symposium on Communications & Networking, CNS '15* (pp. 8–15). San Diego, CA, USA: Society for Computer Simulation International.

А.у, Менциев, Пахаев Х.х, & АйгуМов Т.г. (2021). 'УГРОЗЫ БЕЗОПАСНОСТИ УЗКОПОЛОСНОГО ИНТЕРНЕТА ВЕЩЕЙ И МЕРЫ ПРОТИВОДЕЙСТВИЯ'. *ИнЖенерный Вестник Дона* 10 (82): 32–41.