applied sciences

MDPI

Article

A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques

Sapna Sadhwani, Baranidharan Manibalan, Raja Muthalagu * and Pranav Pawar

Department of Computer Science, Birla Institute of Technology and Science Pilani, Dubai Campus, Dubai International Academic City, Dubai 345055, United Arab Emirates; sapna@dubai.bits-pilani.ac.in (S.S.); f20200061@dubai.bits-pilani.ac.in (B.M.); pranav@dubai.bits-pilani.ac.in (P.P.)
* Correspondence: raja.m@dubai.bits-pilani.ac.in

Abstract: The study in this paper characterizes lightweight IoT networks as being established by devices with few computer resources, such as reduced battery life, processing power, memory, and, more critically, minimal security and protection, which are easily vulnerable to DDoS attacks and propagating malware. A DDoS attack detection model is crucial for attacks in various industries, ensuring the availability and reliability of their networks and systems. The model distinguishes between legitimate and malicious traffic by analyzing network traffic patterns and identifying anomalies. This safeguards critical infrastructure, preserves business continuity, and protects the user experience, minimizing the impact of DDoS attacks. Numerous scholars have studied the notion that protecting lightweight IoT networks essentially requires improving intrusion detection systems. This research is valuable, as it follows a tailored pre-processing methodology specific to IoT network challenges, addressing a pressing need in cybersecurity by focusing on a growing concern related to IoT devices and DDoS attacks, enhancing the security of essential network systems in various industries by effectively detecting DDoS attacks, and developing a lightweight intrusion detection system that aligns with the limited resources of IoT devices. This manuscript proposes a compact and lightweight intrusion detection system that blends machine learning classifiers with a fresh approach to data pre-processing. The handling of missing values, data standardization using Standard Scalar, feature selection using ExtraTreeClassifier wherein only the 15 best features are extracted, and anomaly detection using a classifier are performed. The network dataset of TON-IOT and BOT-IOT datasets is used for experiments, specifically binary classifications and multiple-class classification for the experiment with DDoS and all attacks, respectively. There is an imbalance between the TON-IOT and BOT-IOT attack classes. In trials using the TON-IOT and BOT-IOT datasets, the classes were balanced using several iterations of the SMOTE approach. This research provides a number of classifier types, namely logistic regression, random forest, naïve bayes, artificial neural network, and k nearest neighbor algorithms, which are used to build a lightweight intrusion detection system that is ideally suited for protecting against DDoS attacks in IoT networks. The time taken to train and predict the DDoS attacks is also implemented. Random forest performed well under TON-IOT and naïve bayes performed well under BOT-IOT under binary and multiple-class classification, achieving an accuracy of 100% with less training and prediction time.

Keywords: machine learning; BOT-IOT; TON-IOT; DDoS; SMOTE; IoT; logistic regression; KNN; ANN; random forest; naïve bayes

Citation: Sadhwani, S.; Manibalan, B.; Muthalagu, R.; Pawar, P. A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques. *Appl. Sci.* **2023**, *13*, 9937. https://doi.org/10.3390/app13179937

Academic Editors: Christos Bouras and Mohamed Benbouzid

Received: 10 July 2023
Revised: 30 August 2023
Accepted: 1 September 2023
Published: 2 September 2023

1. Introduction

IoT devices create a network of connected objects that interact with each other and with humans. They communicate wirelessly or through wired connections like Wi-Fi, Bluetooth, or cellular networks. IoT devices collect data from their surroundings, such as temperature, location, health stats, etc. The collected data is sent to cloud-based platforms or local servers for storage, analysis, and processing. IoT devices enable various applications in healthcare,

transportation, agriculture, and home automation. They improve efficiency, convenience, and productivity by taking actions based on the collected data. IoT devices contribute to the interconnectedness of our world, enabling smarter systems and providing valuable insights for decision-making, ultimately enhancing our quality of life.

1.1. DDoS Attacks

DDoS assaults directed toward IoT systems can seriously damage IoT devices [1]. According to the most current estimate, there will be up to 35 billion connected IoT devices by 2025, with roughly 50% unprotected and vulnerable to most security attacks, including DDoS attacks.

In 2017, the IoT Reaper, also referred to as IoTroop, was a botnet that exploited IoT device vulnerabilities to form a large botnet [2]. It targeted devices like IP cameras, routers, and network-attached storage (NAS) devices, infecting them. While its motives were uncertain, the Reaper botnet had the capability to execute significant DDoS attacks. Luckily, security researchers intervened and stopped its growth by gaining control of its command-and-control infrastructure.

In 2018, Satori, also called Okiru, emerged as a variant of the Mirai botnet. It focused on exploiting vulnerabilities in IoT devices, particularly Huawei routers [3]. Satori took advantage of a remote code execution vulnerability found in the routers' management interface to enlist them into the botnet. Once compromised, these devices became potential sources for launching DDoS attacks and carrying out other harmful actions.

1.2. Motivation

Given the IoT devices' and systems network's limited resources, the attackers are not unsophisticated and have expanded themselves with a variety of inventive strategies to infiltrate them. There is typically a lot of traffic when there are billions of IoT devices. As a result, it could be difficult to distinguish between a DDoS attack and an ordinary traffic increase. DDoS identification intelligence is necessary to solve these problems. Both machine learning and deep learning techniques may play predictive roles across a range of various cases to fight against DDoS attacks on IoT infrastructure, giving organizations new insights and increasing the demand for performance tools and IT development. Deep learning and machine learning algorithms are among the best and most powerful methods to protect IoT networks from various attacks. One concern regarding this derives from the fact that DDoS assaults have become more frequent and attempt to target consumer-level IoT devices, as well as from the reality that consumers lack technological expertise or awareness of inherent weaknesses.

A command server in Figure 1 contains multiple users in the internet world who access IoT devices on a day-to-day basis, which is all in turn connected with the target server. This gives the attacker an opportunity to attack the command server, which would indirectly affect the IoT devices that are connected to the network of bots. The creation of resistant solutions for IoT systems to identify attacks is one of the primary objectives for researchers to improve the rate of success in detecting dangerous threats against IoT architecture. To track and forbid unsuitable data flowing through the IoT network, security must be able to find unwanted and malicious traffic. The anomaly detection is done on edge devices, where it can be performed to analyze the data at the edge, allowing for real-time decision-making and reducing the need to transmit large amounts of data to a central server. This is often essential in IoT networks where bandwidth might be constrained. This performance is important, as performing detection at the edge saves bandwidth and reduces latency. Edge devices have constraints on computational power and memory, so efficient algorithms are needed and high performance must be achieved with minimal energy usage, especially in battery-powered devices. Quick detection is essential to respond to threats like DDoS attacks promptly and effective anomaly detection at the edge can provide early warnings of suspicious activities, enhancing overall network security. However, some ML models frequently misclassify the most hazardous traffic flows as a result of incorrect and subpar

feature selection. The main objective is to undertake extensive research that can be applied to IoT fraud detection.

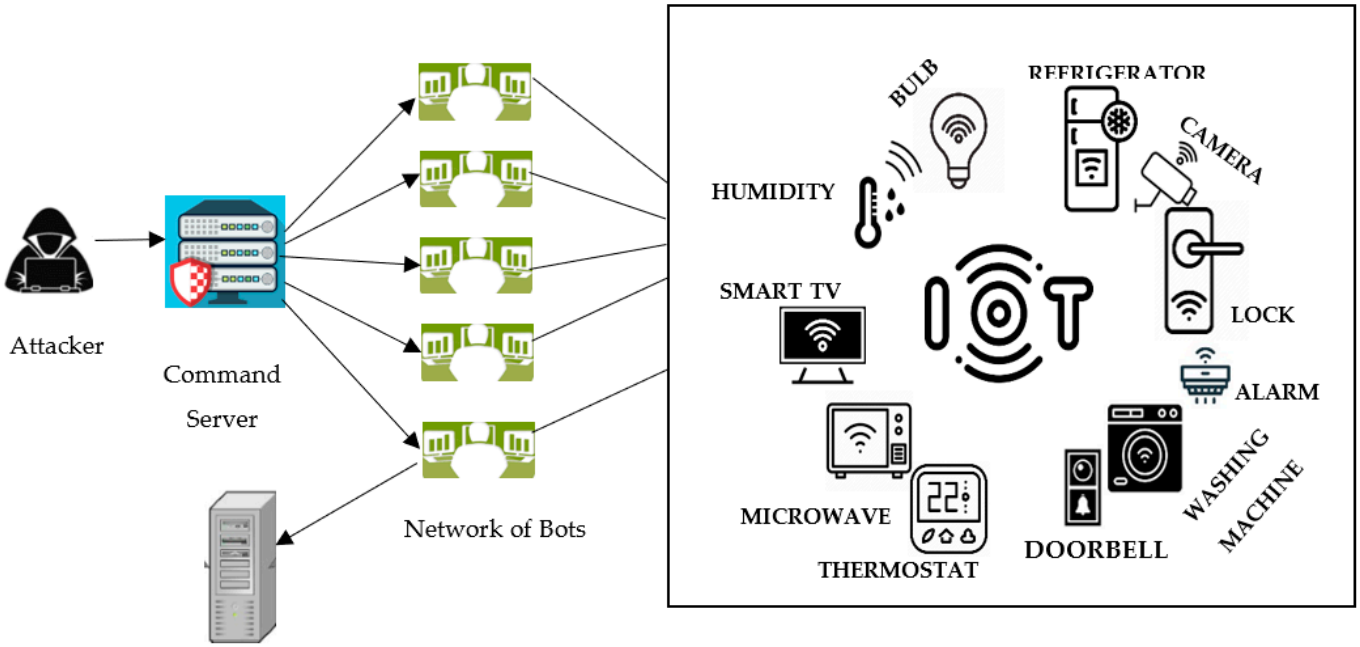


Figure 1. DDoS attack model in lightweight IoT networks.

1.3. Contributions

In the current work an unconventional data pre-processing method is suggested, firstly to find the important features required, and then the feature drop-out technique is suggested to get rid of low-priority features. This article presents a set of classifiers for anomaly detection in IoT traffic network data using machine learning models, along with comparing the performance ratings of each classifier. The identification of DDoS attack patterns and the proliferation of malware with DDoS capabilities remain the main objectives of the research. This article focuses on the detection of DDoS attack labels and malware propagation in IoT network data traffic using the suggested feature selection and pre-processing methods, which are detailed below in Section 3 of the manuscript.

This research has used BOT-IOT and TON-IOT datasets to show how effective the proposed model is with respect to its robustness and generalizability. Diverse datasets capture various network traffic patterns, attack types, and network conditions to develop models that effectively detect attacks in different scenarios. Additionally, this helps researchers gain insights into the challenges of detecting attacks in diverse IoT deployments, as different datasets reflect real-world scenarios. By leveraging two datasets, bias can be avoided, more comprehensive models can be developed, and different attack patterns and network conditions can be effectively handled. The proposed model is proven to be lightweight and much more effective than the previous state-of-the-art models, as feature selection methods like ETC are used and feature dimensionality is narrowed down by using only 15 features. The SMOTE technique is used to resample and balance the datasets to avoid overfitting. K-fold cross-validation is used to enhance the accuracy of multiple-class classification and time factor is implemented to depict the training and prediction time to detect the DDoS attacks. Labels on the dataset were used to group DDoS attack tests with other testing. The following are the research's contributions:

- A modern method of ensemble feature selection for data pre-processing for network IoT datasets.
- Withdrawal of DDoS-attack-related traffic patterns from the TON-IOT and BOT-IOT datasets.

- Comparing and analyzing the performance of machine learning algorithms like LR, NB, RF, ANN, and KNN with binary and multiple-class classification.

2. Literature Survey

To detect and categorize hostile network traffic inside an IoT network, a cutting-edge, DL-based DDoS detection system is proposed in this research [4] to defend the IoT network against new DDoS assaults. This research suggests a multi-classification method for DDoS attacks that is effective, reliable, and scalable. The suggested approach uses a Cu-LSTM-aided framework with GPU support to detect complex multi-vector DDoS attacks in IoTs using the CICDDoS2019 dataset. Cu-enabled LSTM is the employed algorithm. The suggested model showed excellent accuracy in identifying DoS assaults. However, just a small number of folds were employed for cross-validation to achieve the highest level of multi-class classification accuracy. The classifier claimed a detection accuracy of above 99.6% for DDoS threats in the multi-class characterization.

The purpose of this research is to suggest a methodology for identifying unusual DDoS attacks, using the Mirai dataset [5], and to test two well-known DDoS assaults, using the standard dataset. To normalize and clean data and extract valuable features, two methods are implemented. A framework is created for identifying and evaluating anomalous behavior using common machine learning techniques. Linear SVM, neural networks, and decision tree algorithms are employed. According to the experimental findings, a merge of random forest and decision tree was able to detect attacks with an accuracy of 99.7%. The worst-performing algorithm is linear SVM.

The purpose of this research is to offer an IDS for the detection of DDoS attacks that combines deep learning with multi-objective optimization [6]. In order to minimize the dimensionality of the data and pick features, these normalized data are fed into the jumping gene-adapted nondominated sorting genetic algorithm. This method was suggested in the earlier research, which has been submitted for publication. For the NSGA II-aJG algorithm implementation, the author considered the six most crucial goals, which are covered in more depth in the subsection. The deep learning algorithm-based model receives the reduced data that were output from the preceding stage as input data, using a sigmoid function with binary cross-entropy. CISIDS2017 datasets on DDoS were utilized as the datasets. MLP, SVM, bayes, random forest, and genetic algorithm (NSGA-II-aJG) were the algorithms employed. The proposed approach (NSGA-II-aJG) has attained an F1 score value of 99.36% and an accuracy of 99.03%.

The purpose of this article is to offer cutting-edge cyber security solutions to IoT devices and apps for smart cities. This research suggests a botnet detection system that uses network traffic flows and is DL-based [7]. In order to identify assaults coming from infected IoT devices, the botnet detection framework gathers network traffic flows, turns them into connection records, and then utilizes a DL model. Numerous tests are run on well-known and freshly released benchmark datasets in order to identify the best DL model. The datasets are displayed in order to comprehend their features. BASHLITE and Mirai botnet datasets were used. Logistic regression, NB, KNN, DT, RF, RSVM, and linear SVM were the algorithms employed. The SVM models, coupled with KNN and RF models, provided accuracy results of 100%. The KNN and RF models together provided an accuracy of 99%. However, it is noted that the SVM models, when compared to all other algorithms, are computationally expensive and take a long time to learn all the many patterns that exist in datasets during training.

This study focuses on IoT DDoS defense strategies and proposes FlowGuard, an edge-centric IoT defensive system, for detecting, identifying, classifying, and mitigating IoT DDoS attacks [8]. In this study, a DDoS attack detection, identification, classification, and mitigation strategy called FlowGuard was suggested. It makes use of two machine learning approaches that work together: long short-term memory (LSTM) and convolutional neural network (CNN). FlowGuard is made up of two main parts: a flow handler and a flow filter. The latter analyzes suspicious flows for DDoS attack identification and classification and

conducts filtration rule generation using a self-evolving machine. The former maintains the flow filtration rules generated by the handler and is responsible for DDoS attack detection. The CICDDoS2019 dataset was utilized. The proposed model using LSTM acquired an accuracy of 98.9% and with CNN it obtained 99.9%.

The goal of this research was to concentrate on identifying DDoS attacks launched via IoT-bot-infected devices. First, 33 different types of scans and 60 different types of DDoS attacks were generated to create a generic scanning and DDoS attack dataset [9]. The author then suggested a two-pronged machine learning strategy to stop and identify IoT botnet attacks. In order to stop IoT botnet attacks, the author created a cutting-edge deep learning model, ResNet-18, to detect scanning activity in the early stages of an attack while training a second ResNet-18 model for DDoS attack identification to recognize IoT botnet attacks in the second fold. For the dataset CICIDS-2019, BOT-IOT was utilized. It is a logistic regression algorithm. In comparison to the existing trained models, the experimental findings show that the suggested two-fold strategy may effectively prevent and detect botnet attacks. However, if the scanning detection model is unable to stop a botnet attack, the ResNetDDoS-1 model will be used to detect a DDoS attack. In order to prevent and identify IoT botnet assaults, the suggested two-fold technique exhibits 98.89% accuracy, 99.01% precision, 98.74% recall, and 98.87% F1 score.

In this study, a range of machine learning (ML) techniques with built in WEKA tools are used to investigate the detection performance for DDoS assaults [10] using the most recent CICDoS2019 datasets. CICDDoS2019 was judged to be the model that gave the best outcomes. K-NN, SVM, NB, DT, RF, and LR are the six different kinds of machine learning (ML) algorithms that were used in this study. In the evaluation presented, the DT and RF algorithms generated the best results, with accuracy results of 99% and 99%, respectively. However, because of the DT's quicker computation time (4.53 s as opposed to 84.2 s), it outperforms the RF. Unresolved issues are then presented for potential future investigation.

In order to guarantee security from DoS and DDoS attacks, a number of ideas have been put forth in this article. Algorithms for Deep Learning and Machine Learning have both been used to assess DoS and DDoS attacks [11]. For training, the UNSW Canberra Cyber Center's BOT-IOT dataset was utilized. For training and testing purposes, algorithms for machine learning used random forest, KNN, and decision trees with logistic regression as the meta-classifier. As a consequence, the best attack classification accuracies for deep learning and machine learning algorithms, respectively, are 99.5% and 99.9%.

The purpose of this research is to offer a novel machine learning technique for the detection and mitigation of botnet-based DDoS attacks on IoT networks. The suggested model addresses the security concern regarding the dangers posed by bots [12]. A model was developed using a variety of machine learning methods, including KNN, NB model, and MLP ANN, with data from the BOT-IOT dataset as the training set. Based on the accuracy, ROC, and AUC scores, a reference point chose the best algorithm. SMOTE and feature engineering were integrated with ML algorithms. The performance of three algorithms was evaluated on both the class-balanced dataset and the class-imbalanced dataset. Notably, naive bayes mode achieved the highest accuracy of 99.4%.

In this study, the objective is to present a feature engineering and machine learning framework for the IoT-CIDDS dataset to detect DDoS attacks [13]. In the initial stage, the author focused on advanced feature engineering and built algorithms for dataset enrichment in order to statistically analyze the dataset with a probability distribution and feature correlation. In the second step, the author created training, validation, and testing datasets using IoT-CIDDS and presented an ML model while performing complexity analysis of the feature-engineered dataset using five machine learning approaches. Accuracy, precision, recall, area under the curve, false positive rate, and computing time for classifier training are all considered when evaluating ML models. IoT-CIDDS is the dataset in question. Logistic regression, SVM, decision tree, MLP, and random forest algorithms are employed. The evaluation's findings showed that random forest outperformed all other classifiers, having the best detection rate, the fewest false positives, and the fastest calculation time

(98.8% accuracy). The following dataset, however, will provide more accurate results for multi-class classification than binary.

In this research, a hybrid methodology for feature selection using feature selection methods applied to machine learning classifiers is proposed. The entire dataset is subjected to the train–test split, which uses 70% of the dataset's data for training and 30% for testing [14]. The CICDDos2019 dataset, which is based on network flow features and covers two types of assaults (exploitation-based and reflection-based), was used by the author. Machine learning classifiers are used on this dataset, and the highest value of their performance parameters is noted for all attributes. The following describes how to apply chi-square feature selection to machine learning classifiers (RF, DT, KNN, and XGBoost). The first step is to apply these classifiers at intervals of five, after which a series of iterations are carried out, and the number of characteristics for which the greatest accuracy is attained is noted. These classifiers are used at an interval of one to choose an ideal window. Therefore, at the following stage, an ideal window will be created to begin iterations at intervals of one from 30 to 40 features, and determine the precise number of features for which accuracy is highest. The results show that XGBoost and ANOVA together achieve 98.374% accuracy for 15 characteristics. However, XGBoost is not a perfect algorithm since accuracy dropped from 98 to 64 as the number of features increased from 20 to 30.

This study compares and contrasts machine learning (ML) techniques for DDoS attack detection and classification. In this experiment, unsupervised data filtering using the PCA technique is used to extract key features and lighten the load on the classifiers [15]. To successfully detect DDoS attacks, four supervised classification methods are used. The 10-fold cross-validation procedure is then used to validate the results and determine how resilient the suggested approach is. The BOT-IOT dataset was utilized. Naive bayes, J48, random forest, and PCA were the algorithms employed. According to experimental findings, the random forest classification algorithm outperforms the others, with a 99.99% accuracy rate.

The goal of this study was to categorize DDoS attacks as distinct from regular attacks. The CICDDoS2019 dataset was used to train and test classification models [16]. It includes several DDoS assaults, including NTP, DNS, LDAP, and others. To perform binary classification, the categorical labels were encoded into integer format, where one represents benign and zero denotes malevolent. After indexing the string values, the data were standardized to get rid of any misclassification using the CICIDS-2019 data set. Random forest, KNN, decision tree, and ANN were the algorithms employed. The artificial neural network model had the best performance. The decision tree model had the worst performance, with a false negative rate that was significantly higher than that of the other three models. With an accuracy of 99.95%, it was noticed that the ANN model performed better than the other classifier models.

The goal of this research is to accelerate detection while maintaining a respectable degree of detection [17]. The pre-processing of the IoT-Bot dataset and the classification of the several attack types included are both covered in this paper's approach. The author compared the outcomes of random forest, k nearest neighbor, support vector machine (SVM), and logistic regression classifiers from the cuML package using GPU-accelerated versions. The author also included explanations of pre-processing procedures taken to prepare data for training. The ioT-BoT-BoT dataset is the one that was used. The algorithms SVM, logistic regression, and KNN were employed. DDoS detection was quickest using the random forest model. The collected findings indicate that the best-trained model's accuracy and recall are, respectively, 0.999 and 0.997.

The purpose of this study is to suggest a novel architecture made up of two parts: DoS/DDoS detection and DoS/DDoS mitigation [18]. This research suggested an architecture that is made for the Internet of Things and is intended to detect and mitigate DoS/DDoS assaults. The DoS/DDoS detection offers fine-granularity detection since it distinguishes between DoS and DDoS attacks and the attack's packet type. The author used

the appropriate mitigating countermeasure based on the prediction attack outcome. The author employed a multi-class classifier that incorporates the looking-back concept and is tested on the BOT-IOT dataset to identify DoS/DDoS assaults. Decision tree, random forest, KNN, MLP, RNN, and LSTM were among the algorithms utilized. The evaluation's positive findings include the looking-back-enabled random forest's 99.81% accuracy.

This work aims to classify and forecast types of DDoS attacks using machine learning [19]. The choice of a dataset for use is made in the first phase. The choice of tools and language comes in the second phase. The third stage uses data pre-processing methods to deal with the dataset's irrelevant data. The fourth phase is the labeling and extraction of features. To transform symbolic data into numerical data, encoding is used. The data is separated into a train and test set for the model in the fifth stage. To increase model effectiveness, however, the trained model also undergoes model optimization in terms of kernel scaling and kernel hyperparameter adjustment. The UNWS-NB-15 dataset was utilized. XGBoost and random forest were the algorithms employed. Random forest has an accuracy rate of 89%, compared to 90% for XGBoost.

This study suggests a modified long short-term memory deep-learning-method-based IDS for detecting DoS attacks in IoT networks [20]. To evaluate the model, benchmark datasets CICIDS-2017 and NSL-KDS were employed. The datasets underwent normalization, dimensionality reduction, and encoding in three pre-processing steps. The suggested RLSTM model detected DoS assaults on the CICIDS-2017 dataset with 99.22% accuracy. Furthermore, it achieved 99.23% precision, 99.22% recall, and 99.22% f-score rates for detecting DoS attacks on the CICIDS-2017 dataset. Using the NSL-KDD dataset, the model achieved 98.60% on all performance metrics.

This study compares features from the UNSW-NB-15 and BOT-IOT datasets based on flow and TCP in order to propose a PB-DID architecture, creating a dataset of packets from IoT traffic in the process [21]. The work differentiates between non-anomaly, DoS, and DDoS traffic by addressing issues like imbalance and overfitting. Using DL and LSTM, it was able to attain a classification accuracy of 96.3%.

The manuscript suggests a lightweight IDS that combines machine learning and deep learning classifiers with a novel data pre-processing method [22]. Datasets from TON-IOT by UNSW and BOT-IOT were used for the tests and analysis. Both datasets are used to create DDoS attack instances. For binary and multiple-class classifications of attack labels, two separate experiments are run on each dataset, one for all attacks and the other just for DDoS attacks in both datasets. In the trials carried out on the BOT-IOT dataset they used the SMOTE approach and variants for class balancing.

In all this research it can observed that multiple datasets related to DDoS attacks are used, such as CICDDoS, BOT-IOT, UNWS-NB, etc., and most of the ML and DL algorithms are giving good results for the models. It is observed that random forest, logistic regression, and naïve bayes algorithms give really good results for DDoS attacks in IoT. According to previous work, the existing models are complex and not as reliable in detecting DDoS attacks when compared to the proposed model. The research has performed data standardization along with feature selection to retrieve the 15 best features out of 45 features from both datasets over which training and testing were performed. Hence, this approach makes the model a lightweight model. The SMOTE technique was used to cross out overfitting and resample the imbalanced data. Five-fold cross-validation was performed for multiple class classification to improve accuracy. Five ML algorithms, namely logistic regression, random forest, ANN, naïve bayes, and KNN, were used to train and test the model (Table 1). Considering this above-mentioned methodology, it can be stated that this research is very much justified and effective in detecting DDoS attacks efficiently in IoT devices, and it managed to achieve higher accuracy than all other existing models. One of the most important factors that is required while detecting attacks in IoT devices, which is present in this model and is not present in any other state of the model, is training and prediction time for all algorithms.

Table 1. Literature survey.

Ref	Year	Objectives	Dataset	Algorithms	Limitations	Results
[4]	2020	To detect and categorize hostile network traffic inside an IoT network, a cutting-edge, DL-based DDoS detection system is proposed in this research to defend the IoT network against new DDoS assaults.	CICDDoS2019	Cu-enabled LSTM	A small number of folds were employed for cross-validation to achieve accuracy.	The classifier claimed a detection accuracy of above 99.6% for DDoS threats in the multi-class classification.
[5]	2020	The purpose of this research is to suggest a methodology for identifying unusual DDoS attacks.	Mirai	LSVM, neural networks, decision tree	The linear SVM algorithm performed very poorly.	According to the experimental findings, a merge of random forest and decision tree was able to detect attacks with an accuracy of 99.7%.
[6]	2020	The purpose of this research is to offer an IDS for the detection of DDoS attacks that combines deep learning with multi-objective optimization.	CISIDS2017	MLP, SVM, bayes and random forest, genetic algorithm (NSGA-II-aJG)	MLP is the worst model for detecting DDoS attacks.	The proposed approach (NSGA-II-aJG) has attained an F1 score value of 99.36% and an accuracy of 99.03%.
[7]	2020	The purpose of this article is to offer cutting-edge cyber security solutions for IoT devices and apps for smart cities.	BASHLITE and Mirai botnet datasets	Logistic regression, NB, KNN, DT, RF, RSVM, linear SVM	The SVM model is computationally expensive and takes a long time.	SVM models, coupled with KNN and RF models, provided accuracy results of 100%. KNN and RF models together provided an accuracy of 99%.
[8]	2020	This study focuses on IoT DDoS defense strategies and proposes FlowGuard to detect IoT DDoS attacks.	CICDDoS2019	LSTM, CNN	Detects unidentified malicious flows based on traffic variations only.	The proposed model using LSTM acquired an accuracy of 98.9% and with CNN it obtained 99.9%.
[9]	2021	The goal of this research was to concentrate on identifying DDoS attacks launched via IoT-bot-infected devices.	CICIDS-19, BOT-IOT	Logistic regression	There are chances of failing to detect the botnet attacks.	The suggested two-fold technique exhibits 98.89% accuracy, 99.01% precision, 98.74% recall, and a 98.87% F1 score.
[10]	2021	To build a DDoS attack protection system for IoT devices using WEKA tools and ML techniques.	CICDDoS2019	KNN, SVM, NB, DT, RF	Validation of implemented structures is critical.	In the evaluation that was presented, the DT and RF algorithms generated the best results, with accuracy results of 99% and 99%, respectively.
[11]	2021	To guarantee security from DoS and DDoS attacks using ML and DL techniques.	BOT-IOT	Random forest, decision trees	Many more algorithms can be used to experiment.	The best attack classification accuracy for deep learning and machine learning algorithms, respectively, are 99.5% and 99.9%.
[12]	2021	The purpose of this research is to offer a novel machine learning technique for the detection and mitigation of botnet-based DDoS attacks on IoT networks.	BOT-IOT	Naïve bayes	The imbalance dataset is used to train.	Three algorithms' performance was compared, and a 99.4% accurate naïve bayes model was found.
[13]	2021	To present a feature engineering and machine learning framework for the IoT-CIDDS dataset to detect DDoS attacks.	IoT-CIDDS	Logistic regression, SVM, decision tree, MLP, random forest	Much higher accuracy can be depicted with this proposed model.	The evaluation's findings showed that random forest outperformed all other classifiers, having the best detection rate, the fewest false positives, and the fastest calculation time (98.8% accuracy).

Table 1. *Cont.*

Ref	Year	Objectives	Dataset	Algorithms	Limitations	Results
[14]	2021	To provide a hybrid methodology for feature selection using feature selection methods applied to machine learning classifiers.	CICDDDoS2019	KNN, random forest, decision tree, XGBoost	XGB is not an ideal algorithm as it has 20 to 30 features and accuracy reduced from 98% to 64%.	The results show that XGBoost and ANOVA together achieve 98.374% accuracy for 15 attributes.
[15]	2021	This study compares and contrasts machine learning (ML) techniques for DDoS attack detection and classification.	BOT-IOT	Naive bayes, J48, random forest, PCA	J48 is the worst performing model.	The random forest classification algorithm outperforms the others, with a 99.99% accuracy rate.
[16]	2022	The goal of this study was to categorize DDoS attacks from regular attacks.	CICDDoS2019	Random forest, KNN, decision tree, ANN	The DT model performed the worst with a high false negative rate.	With an accuracy of 99.95%, it is noticed that the ANN model performs better than the other classifier models.
[17]	2022	The goal of this research is to accelerate detection while maintaining a respectable degree of detection.	IoT-BoT	SVM, logistic regression, KNN	GPU technology is used, which reduces training and prediction time.	The collected findings indicate that the best-trained model's accuracy and recall are, respectively, 99.9% and 99.7%.
[18]	2022	This research suggests an architecture that is made for the Internet of Things and is intended to detect and mitigate DoS/DDoS assaults.	BOT-IOT	Decision tree, random forest, KNN, MLP, RNN, LSTM	The KNN model exhibits a significant decrease at every looking-back step.	The evaluation's positive findings include looking-back-enabled random forest's 99.81% accuracy.
[19]	2022	This work aims to classify and forecast types of DDoS attacks using machine learning.	UNWS-NB-15	Random forest, XGBoost	Higher accuracy can be obtained with a better proposed model.	Random forest has an accuracy rate of 89% compared to 90% for XGBoost.
[20]	2022	This study suggests a modified long short-term memory deep-learning-method-based IDS for detecting DoS attacks in IoT networks.	CICIDS-2017, NSL-KDS	Refined LSTM and MLP	MLP models did not perform well with the CICIDS-2017 dataset.	This model achieved 99.23% precision, 99.22% recall, and 99.22% f-score rates for detecting DoS attacks on the CICIDS-2017 dataset. Using the NSL-KDD dataset, the model achieved 98.60% for all performance metrics.
[21]	2022	This study compares features from the UNSW-NB-15 and BOT-IOT datasets based on flow and TCP in order to propose a PB-DID architecture, creating a dataset of packets from IoT traffic in the process.	UNSW-NB15, BOT-IOT	LSTM	The model is very heavy.	Using DL and LSTM, the researchers were able to attain a classification accuracy of 96.3%.
[22]	2022	The research suggests a lightweight IDS that combines machine learning and deep learning classifiers with a novel data pre-processing method.	BOT-IOT, TON-IOT	Linear SVM, LR, naïve bayes, LSTM, ANN	State-of-the-art model not provided for validating the proposed model.	The LSTM model performed the best in both BOT-IOT and TON-IOT for binary and multiple classifications, with 99% and 95% accuracy, respectively.

3. Proposed Lightweight Model for Intrusion Detection

Each proposed machine learning algorithm in this study can be implemented as a classifier in the proposed model. In this research, a lightweight intrusion detection model is built by combining classifiers from ML. In broad binary classification, machine learning techniques and DL algorithms are directly in competition. IDSs get many different types of data frames. An IDS collects data systematically from various sources from systems using standard logging techniques such as code, packets, memory discs, and functions. A network, the host, or any similar activity can be detected by an IDS. An incursion, whether

it is dynamic or static, can be found in a network when an IDS is utilized. The model presented below indicates a few fundamental and crucial steps that an intrusion detection system should perform. It runs through each cycle of the technique until DDoS attack traffic is identified and separated from ordinary traffic. Since distinguishing legitimate traffic from infected traffic is the first and most crucial step of an intrusion detection system, the design of the IDS model is addressed in detail in Figure 2.

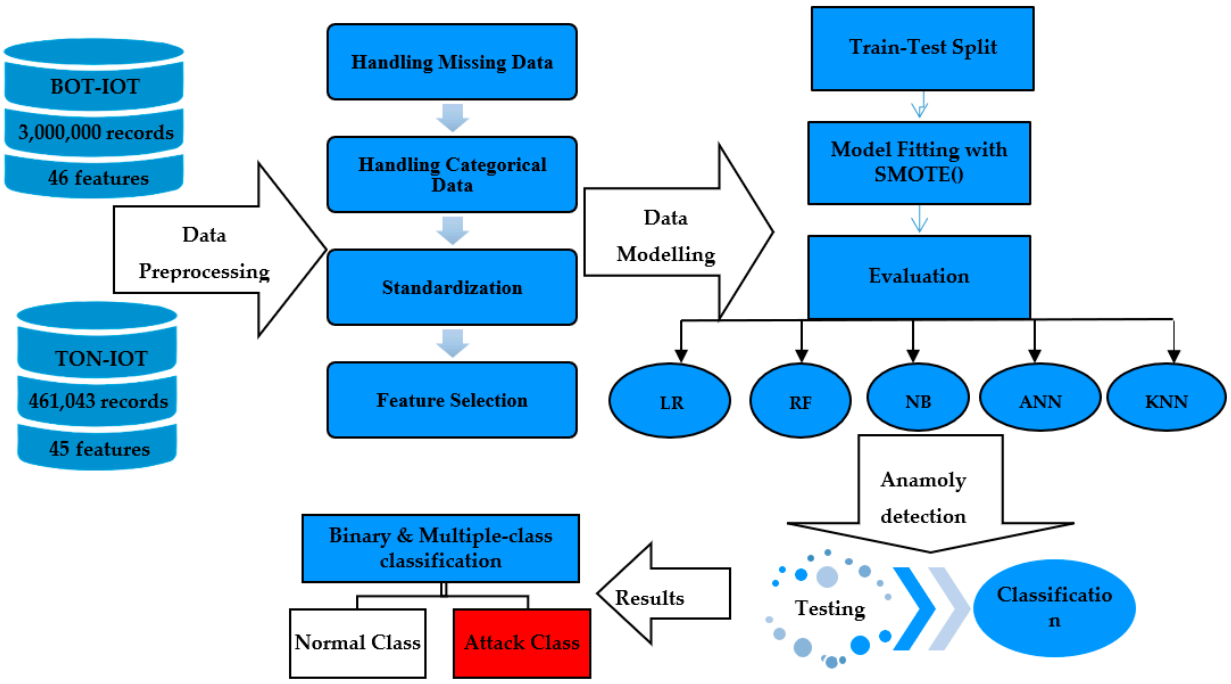


Figure 2. Proposed lightweight model for intrusion detection.

3.1. ToN-IoT Dataset

The network ToN-IoT dataset is a comprehensive collection of mixed data from various sources within the Internet of Things (IoT) and the Industrial Internet of Things (IIoT). It includes diverse data such as telemetry from connected devices, system logs from Windows and Linux, and network traffic information. The dataset is designed to evaluate the accuracy and efficiency of cybersecurity applications based on artificial intelligence. It simulates a realistic network environment by connecting virtual machines, cloud layers, blurred edges, and physical systems.

The dataset comprises both legitimate and offensive events, encompassing network systems, operating systems, and IoT services. It is represented in CSV format and includes 461,043 records with 45 different features. These features provide information about various aspects such as timestamps, IP addresses, ports, protocols, service details, duration, byte counts, connection states, DNS queries, SSL information, HTTP details, anomalies, and labels, indicating the type of behavior or attack.

In summary, the ToN-IoT dataset is a diverse and extensive collection of data from IoT and IIoT sources, allowing researchers to analyze and develop cybersecurity applications based on artificial intelligence. It provides a realistic representation of network environments and contains labeled data for different types of attacks, enabling the evaluation and development of effective cybersecurity measures.

3.2. BoT-IoT Dataset

The BoT-IoT dataset was created in the Cyber Range Lab of UNSW Canberra by designing a realistic network environment. This environment includes both normal and botnet traffic. The dataset is available in multiple file formats, such as pcap, argus, and CSV.

The files are categorized based on attack types and subcategories to facilitate the labeling process. The dataset covers various attack categories, including DDoS, DoS, OS and service scan, keylogging, and data exfiltration. DDoS and DoS attacks are further organized based on the protocol used. A 5% sample was extracted using MySQL queries to make the dataset more manageable.

This paper specifically uses the extracted 5%, which consists of four files totaling approximately 1.07 GB in size. It contains around 3 million records with 46 different features for training and testing. The features in the dataset include information such as packet sequence ID, timestamps, flags, protocols, source and destination addresses, packet and byte counts, connection states, durations, statistical values, and various network metrics. Additionally, the dataset provides attributes related to attack labels, categories, and subcategories.

In summary, the BoT-IoT dataset is a comprehensive collection of network traffic data created in a realistic environment. It contains a subset of records from various attack categories and subcategories, enabling researchers to train and test cybersecurity models. The dataset's features cover a wide range of network-related attributes, facilitating the analysis and development of effective defense mechanisms against different types of attacks.

3.3. Data Pre-Processing

3.3.1. Handling Missing Values

The first essential step when it comes to data pre-processing is managing the missing values in a dataset. Generally, when it comes to data pre-processing, statistical terms like standard deviation, mean, range, etc., are used to replace missing values. The clean versions of the TON-IOT and BOT-IOT datasets—without any missing values—were used for this study because of the datasets' low percentage of missing values. Attributes like timestamps and IP addresses have been removed as these features can lead to overfitting. The model might learn patterns specific to those IPs or particular times, which do not generalize well to new, unseen data. This can lead to poor performance when the model is applied to data from different time frames or different IP addresses. In many cases, attributes like IP addresses may not provide meaningful information for the task at hand. Including irrelevant features can introduce noise and reduce the predictive performance of the model.

3.3.2. Handling Categorical Data

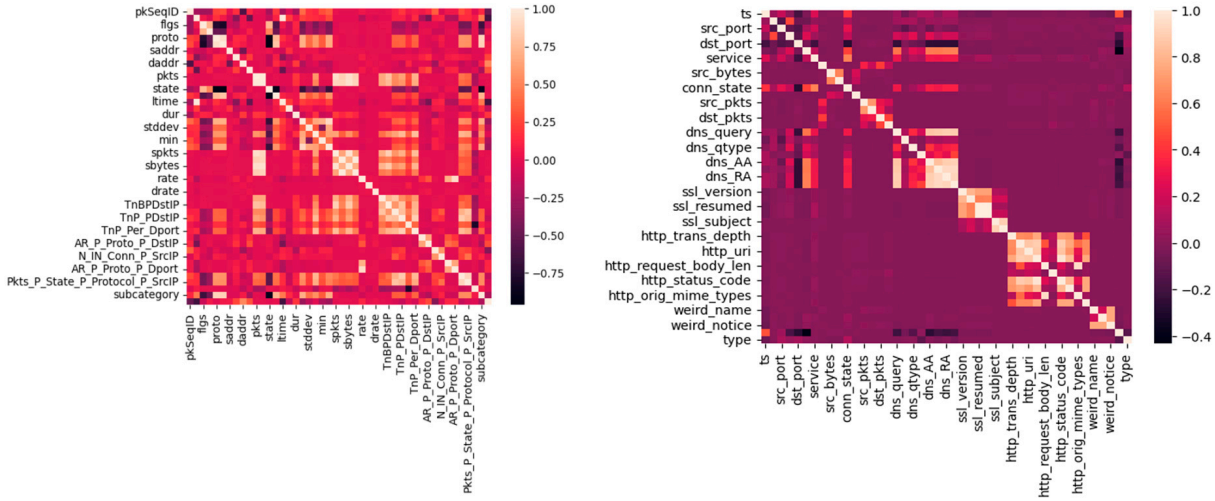
Machine learning classifiers can face difficulties when working with datasets that contain attributes with a wide range of string values. Fortunately, Python provides several libraries, such as "sklearn", which offer useful tools for handling such scenarios. One specific sub-library within scikit-learn, called "sklearn.preprocessing", can be employed to address these challenges by utilizing label coding. This approach involves managing categorical data and transforming string values into numerical representations based on the classes present in that particular feature. Scikit-learn is a reliable Python tool that offers functionality for tasks like one-hot encoding, binarization, and digitization of categorical data. In this suggested model, the label encoding technique from the "sklearn.preprocessing" tool is applied to achieve the desired transformation.

3.3.3. Data Standardization

A few of the features in the TON-IOT and BOT-IOT datasets are not in a model-friendly format. The source and destination IP addresses are among the components of the IP address format. Following label encoding, the values must be properly normalized. To normalize the input data for the suggested model, Standard Scalar is used. Identifying the attributes to standardize the IP addresses in the dataset is essential before beginning the standardization process.

In order to locate the appropriate output that will improve the results, a correlation matrix, as shown in Figure 3, is developed in accordance with the TON-IOT dataset. The "conn_state" attribute has been shown to be the ideal attribute to standardize with

IP addresses. Similarly, in the BOT-IOT dataset, the correlation matrix is created and it is observed that the "proto_number" attribute is the best to standardize, along with IP addresses.



(**a**) (**b**)

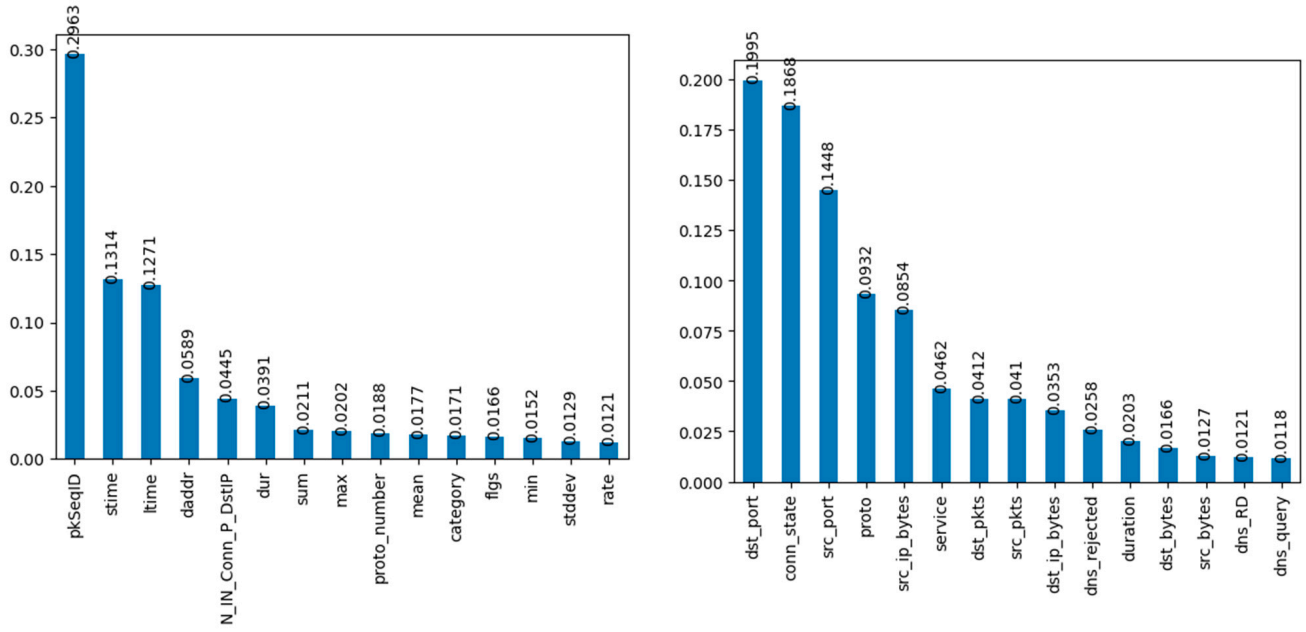
Figure 3. (**a**) Correlation matrix of BOT-IOT; (**b**) correlation matrix of TON-IOT.

3.4. Feature Selection

To assess the suggested model in the paper without encountering overfitting, important characteristics from the input dataset were added. However, selecting a large number of features for a traditional ML classifier increases the risk of overfitting. To address this, a novel feature selection approach called ExtraTreeClassifier was developed to include key features in the model development. Twenty critical features were selected from both datasets, and the remaining features were discarded. The ExtraTreeClassifier algorithm is significant for detecting DDoS attacks due to its ability to handle high-dimensional data with both categorical and numerical features without requiring feature scaling. It effectively addresses imbalanced datasets, which is crucial in DDoS attack detection where attack instances are typically fewer than normal instances.

The algorithm's computational efficiency makes it suitable for real-time or high-speed network environments, enabling prompt detection of DDoS attacks. It excels at capturing complex non-linear relationships between features and enabling accurate identification of DDoS attack patterns amidst normal traffic. The algorithm's resistance to overfitting minimizes false positives and negatives in detecting DDoS attacks. With the capacity to handle large feature sets, the ExtraTreeClassifier enables comprehensive analysis of datasets with numerous network traffic attributes, enhancing the effectiveness of DDoS attack detection. Its ensemble nature and randomness in feature selection make it robust against noisy or irrelevant features, thereby improving overall performance and reliability.

Easy implementation and integration into existing network security systems make the ExtraTreeClassifier convenient for deployment in real-world scenarios, facilitating the utilization of DDoS attack detection. By evaluating the qualities based on their Gini importance from least to most important, the top K attributes are chosen by the user. The relevance of the top 15 features, as shown in Figure 4, from the TON-IOT and BOT-IOT datasets is compared using ExtraTreeClassifier from the sklearn.ensemble module.



(**a**) (**b**)

Figure 4. (**a**) Fifteen important features of BOT-IOT; (**b**) fifteen important features of TON-IOT.

3.5. Data Modeling

The data modeling process begins by dividing the data frame into segments. In the suggested model, the classification model is trained on 80% of the data frame and evaluated model on the remaining 20%. To accomplish this, the model imports the "train_test_split" function from "sklearn.model_selection" with a 20% testing rate. After splitting the data into training and testing samples, they are provided to the classifier for training and evaluation. A SMOTE technique is implemented to oversample the attack class.

SMOTE is a significant technique for detecting DDoS attacks due to its ability to address class imbalance. Solving this imbalance problem on these 2 datasets can avoid overfitting the model. It generates synthetic samples of the minority class (DDoS attacks), effectively balancing the dataset and improving training and classification performance.

SMOTE mitigates bias towards the majority class, enabling accurate detection of DDoS attack patterns. By oversampling the minority class, it enhances the classifier's ability to identify and distinguish DDoS attacks from normal traffic, reducing the risk of false negatives. It improves the classifier's generalization ability by providing diverse samples, enabling effective handling of unseen attack patterns. It is compatible with various machine learning algorithms and reduces the need for collecting real-world attack instances. Its effectiveness in improving DDoS attack detection has been demonstrated in numerous studies; it enhances network security and mitigates potential damages caused by such attacks. Furthermore, a 5-fold split is employed only for the multiple-class classification to enhance accuracy during the assessment of the model.

3.6. Anomaly Detection by a Classifier

Anomaly detection by classifier is a technique used to detect DDoS attacks in IoT devices by categorizing network traffic as normal (Class 0) or attack (Class 1). The classifier is trained on labeled data, encompassing normal traffic patterns and known DDoS attack patterns. This enables the classifier to identify anomalous behaviors associated with attacks. It leverages features extracted from network traffic, such as packet sizes, flow rates, and communication patterns, to differentiate between normal and attack instances. The classifier's performance is evaluated using metrics like accuracy, precision, recall, and F1 score to measure its effectiveness in minimizing false positives and false negatives.

Anomaly detection by classifier complements signature-based methods by identifying previously unseen or zero-day attacks that lack known patterns. It strengthens network security in IoT devices by providing an additional layer of defense against DDoS attacks, ensuring a timely response and preserving the availability and performance of IoT services.

4. Results and Discussion

This paper has implemented its experiment with the help of Google Colab for working on the TON-IOT dataset and Jupyter Notebook for working on the BOT-IOT dataset. Two Python programming tools were used for this experiment as Google Colab gives limited data usage capacity to run the experiment, which is not sufficient to run the BOT-IOT dataset as well. Hence, the Jupyter Notebook was used. This experiment is run on Intel(R) Core (TM) i7-9750H CPU @ 2.60 GHz, 2592 Mhz, six core(s), twelve logical processor(s) with 16 GB RAM, and NVIDIA GeForce GTX 1660 Ti with Max-Q Design 4 GB graphic card. It was very easy to import the network dataset of TON-IOT as it had limited records (461,043 records) compared to the BOT-IOT dataset, which had four parts and had to be concatenated together to create a 3-million-record dataset. Initially, both of the datasets were experimented on with only 20 features and still gave excellent results compared to the state-of-the-art models, but it was observed that even with 15 features the model was able to compute excellent results that prove how efficient the proposed model is, with the help of feature selection and resampling methods.

The TON-IOT and BOT-IOT datasets were used to model the dataset using five machine learning algorithms for DDoS attacks and regular traffic examples. When compared to attack traffic flow, normal instances are seen to occur less frequently in the TON-IOT dataset than in BOT-IOT. Therefore, it is necessary to balance the data frame's attack class and normal traffic class. Hence, the SMOTE technique was implemented, which was discussed in the previous section. Initially, in TON-IOT, Class 0 had 239,945 records and Class 1 had 16,055 records, but after resampling Class 0 and Class 1 both had 239,945 records. Similarly, in BOT-IOT, Class 0 had 1,541,298 records and Class 1 had only 382 records, but after resampling Class 0 and Class 1 had 1,541,298 records. The proposed model is compared with other existing models (Table 2).

Table 2. Comparison of existing data pre-processing techniques and proposed procedures.

Paper	Year	Handling Missing Values	Handling Categorical Data	Standardization	Feature Selection
[23]	2022	Minkowski Dist.	-	-	SMOTE
[24]	2021	Imputation	One Hot Encoding	Min-Max Scalar	Chi-Square Test
[22]	2022	-	Label Encoding	Standard Scalar	ExtraTreeClassifier
Proposed Model	**2023**	**-**	**One Hot, Label Encoding**	**Standard Scalar, Min-Max Scalar**	**ExtraTreeClassifier**

Each time, a collection of five algorithms is employed to decide how to classify and predict the data frames. The following machine algorithms are applied in each situation: logistic regression, naive bayes, artificial neural network, random forest, and k nearest neighbor (Table 3). The research's objective is to evaluate and contrast the performance of various categorization and prediction algorithms. The results of all the tests mentioned above are shown in a table that also includes an area under the curve score, a receiver operating characteristic (ROC) curve, and a confusion matrix with Class 0 (normal) and Class 1 (attack), along with the time taken to train and predict the data, categorizing two cases each for both BOT-IOT and TON-IOT datasets.

- Case 1: multiple-class classification using 15 features of TON-IOT with SMOTE;
- Case 2: binary classification using 15 features of TON-IOT with SMOTE;
- Case 3: multiple-class classification using 15 features of BOT-IOT with SMOTE;
- Case 4: binary classification using 15 features of BOT-IOT with SMOTE.

Table 3. Hyperparameter table for all algorithms used in the paper.

Classifier	Hyperparameter	Parameter	Value
ExtraTreeClassifier	ExtraTreeClassifier	criterion	Gini
		max_depth	None
		min_weight_fraction_leaf	0.0
		max_leaf_nodes	None
		min_impurity_decrease	0.0
		bootstrap	False
		oob_score	False
		n_jobs	None
		random_state	None
		verbose	0
		warm_start	False
		class_weight	None
		ccp_alpha	0.0
		max_samples	None
SMOTE	SMOTE	sampling_strategy	'auto'
		random_state	60
		n_jobs	None
Logistic Regression	LogisticRegression	dual	False
		fit_intercept	False
		intercept_scaling	1
		class_weight	None
		random_state	42
		max_iter	100
		multi_class	'auto'
		verbose	0
		warm_start	False
		n_jobs	None
		l1_ratio	None
Random Forest	Random Forest Classifier	max_depth	None
		min_weight_fraction_leaf	0.0
		max_leaf_nodes	None
		min_impurity_decrease	0.0
		bootstrap	False
		oob_score	False
		n_jobs	None
		random_state	50
		verbose	0
		warm_start	False
		class_weight	None
		ccp_alpha	0.0
		max_samples	None

Table 3. *Cont.*

Classifier	Hyperparameter	Parameter	Value
Naïve Bayes	GaussianNB	random_state	0
Artificial Neural Network	model = Sequential() model.add(Dense(32, input_dim = 15, activation = 'relu')) model.add(Dense(16, activation = 'relu')) model.add(Dense(2, activation = 'softmax')) model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])	model.add(Dense(input_dim=, acitivation=))	(32, 15, 'relu')
		model.add(Dense(activation=))	(16, 'relu')
		model.add(Dense(activation=))	(2, 'softmax')
		model.compile(loss=, optimizer=, metrics=)	'categorical_crossentropy', 'adam', ['accuracy']
K Nearest Neighbour	KneighborsClassifier	n_neighbors	10
		weights	'uniform'
		algorithm	'auto'
		metric	'minkowski'
		metrix_params	None
		n_jobs	None

The precision-, recall-, F1-score-, and accuracy-containing performance matrices are used to assess the performance of all five methods.

4.1. Logistic Regression

A supervised machine learning approach used for classification tasks is logistic regression. It is a linear model that converts the input data into a likelihood of belonging to a particular class using a logistic function. The logistic function, also called the sigmoid function, produces values between zero and one on the basis of an s-shaped curve. The approach learns the ideal weights for the input characteristics that minimize the error between the predicted probability and the true labels in order to train a logistic regression model. A method known as maximum likelihood estimate is used for this. By running the input features through the logistic function and categorizing the data based on the resulting probability, the model may be trained to make predictions on new data. The confusion matrix is given below for TONIOT and BOTIOT, representing binary and multiple-class classification for each (Case 1 to Case 4) for 15 features (Table 4).

Table 4. Confusion matrix for TONIOT (Case 1 and Case 2), BOTIOT (Case 3 and Case 4), and LR (15 features).

Case 1 (All Attacks)			**Case 2 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	59,922	0	Class 0 (Normal)	60,015	0
Class 1 (Attack)	0	32,287	Class 1 (Attack)	0	3985
Case 3 (All Attacks)			**Case 4 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	733,605	0	Class 0 (Normal)	385,324	0
Class 1 (Attack)	0	100	Class 1 (Attack)	0	97

In Case 1, which corresponds to multiple-class classification for the TON-IOT dataset, it is observed that the FN value is higher than the FP value, and therefore there are chances that normal traffic can be detected as an attack. However, the TP and TN values compensate for the FP and FN values, giving a fairly decent accuracy. Similarly, in Case 3 with the BOT-IOT dataset for multiple-class classification with a higher number of records than TON-IOT, there is a fairly high margin with TP and TN values when compared to FP and FN values and hence it gives much higher accuracy. In Cases 1 and 4, with TON-IOT and

BOT-IOT datasets for binary classification, both FP and FN values are zero, and therefore tend to receive the maximum accuracy. The subsequent ROC curves are given for BOT-IOT and TON-IOT, with 15 features for multiple-class and binary classifications (Figure 5). It is observed that Figure 5b,d, which correspond to the binary classification for TON-IOT and BOT-IOT, respectively, have steep ROC curves that touch the top left corner, which makes these two cases very good models. However, in Figure 5a with the TON-IOT dataset for multiple-class classification, it is observed that the curve is closer to the diagonal, hence making it a rather poor model compared to Figure 5c. For Figure 5c, with the BOT-IOT dataset for multiple-class classification, it observed that the curve is rather more distant from the diagonal, and therefore it is a better model with greater accuracy. Similarly, the AUC scores are variable with respect to their ROC curves as observed below.

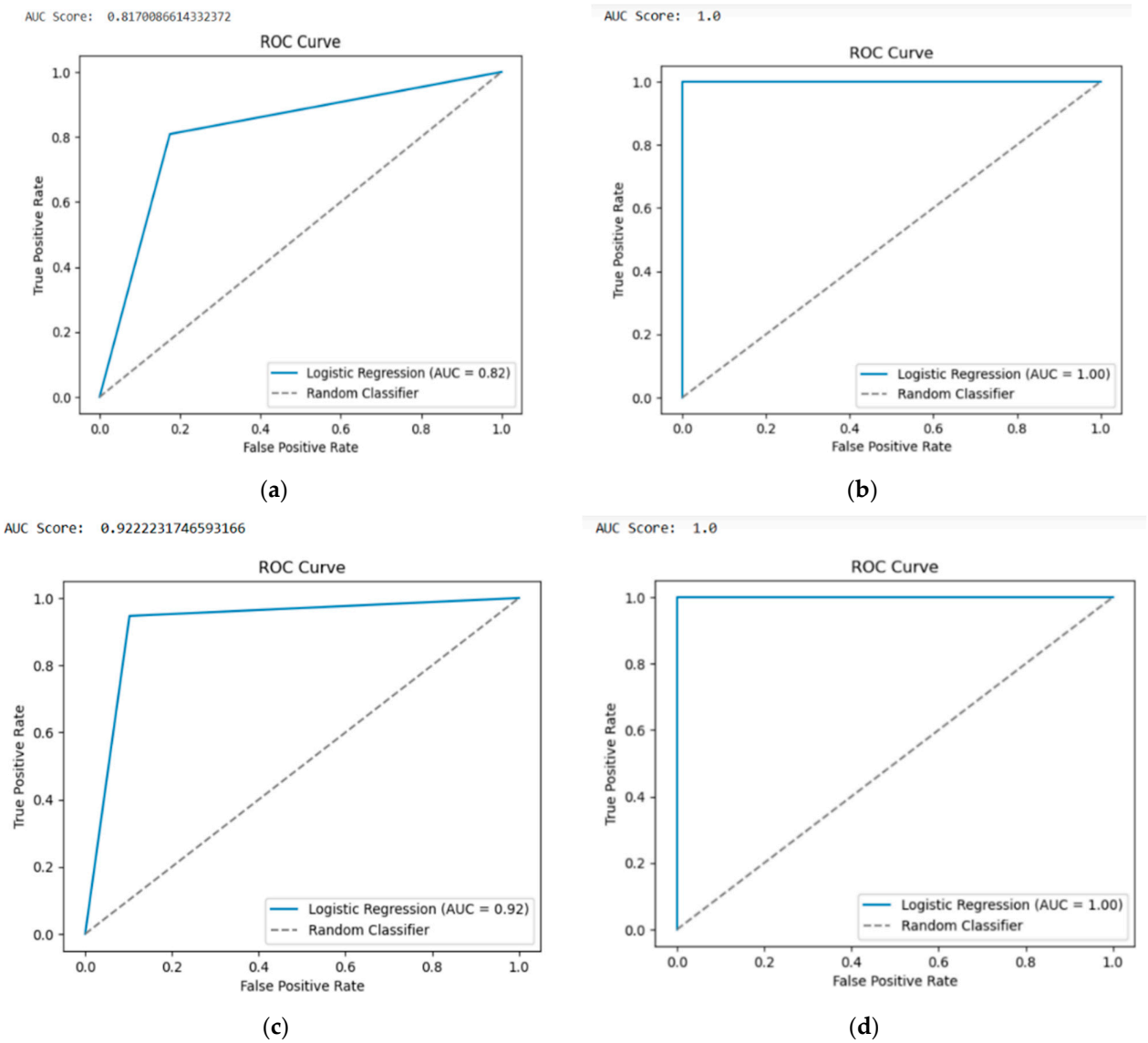


Figure 5. (**a**) ROC curve for TON-IOT for all attacks (15 features), LR; (**b**) ROC curve for TON-IOT DDoS attacks (15 features), LR; (**c**) ROC curve for BOT-IOT all attacks (15 features), LR; (**d**) ROC curve for BOT-IOT DDoS attacks (15 features), LR.

Logistic regression's simplicity, interpretability, and efficiency are its key benefits. It can also manage jobs involving binary and multiple classes of classification. However, when the input data cannot be separated linearly or when the connection between the input characteristics and the output variable is non-linear, logistic regression may not work

well. Overall, the algorithm of logistic regression is helpful for classification problems in machine learning, especially when the data are linearly separable and the link between the input characteristics and the output variable is straightforward.

$$y' = bx + a \tag{1}$$

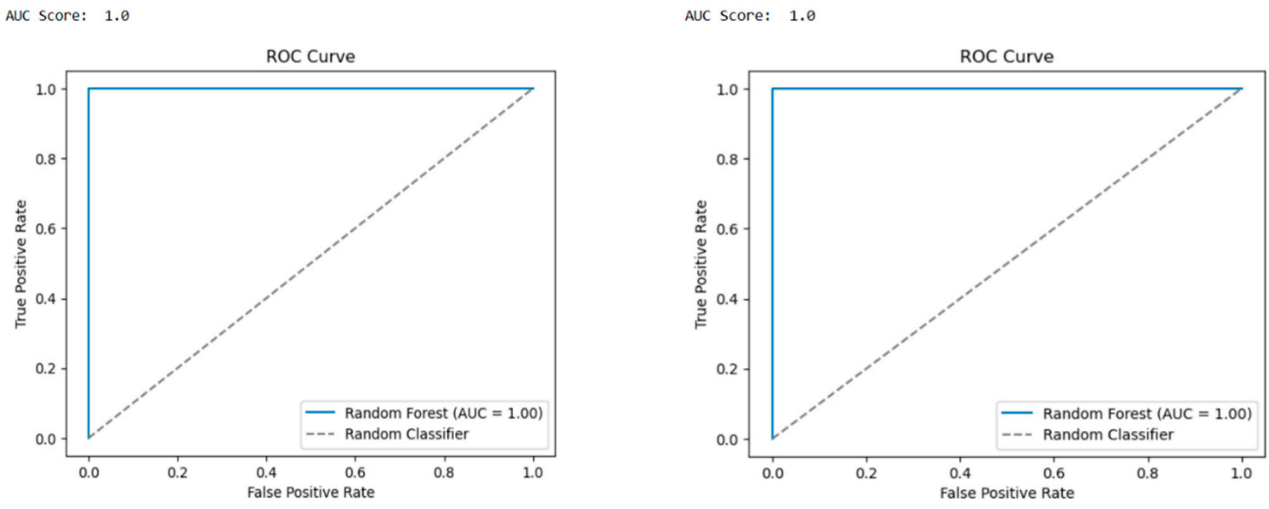
4.2. Random Forest

A supervised machine learning technique called random forest is used for both classification and regression tasks. It is a member of the family of ensemble learning algorithms that combine a number of ineffective learners to produce a more robust model. A random forest is a group of decision trees, where each tree is trained on a random subset of the training data and a random subset of features is considered for splitting at each node of the tree. This lessens overfitting and improves the generalization capabilities of the model.

The confusion matrix is given below for TON-IOT and BOT-IOT, representing binary and multiple-class classification for each (Case 1 to Case 4) for 15 features (Table 5). It is observed that in all cases the FP and FN values are zero, therefore, maximum accuracies will be achieved and this determines that the model is robust. The subsequent ROC curves are given for BOT-IOT and TON-IOT with 15 features for multiple-class and binary classifications. When the steepness of the ROC curve reaches the top-left corner then it is considered an excellent model, hence in this case all of them are very good models (Figure 6). The AUC scores provide a summary measure of the model's discriminatory power, so a higher AUC indicates a more effective DDoS attack detection model.

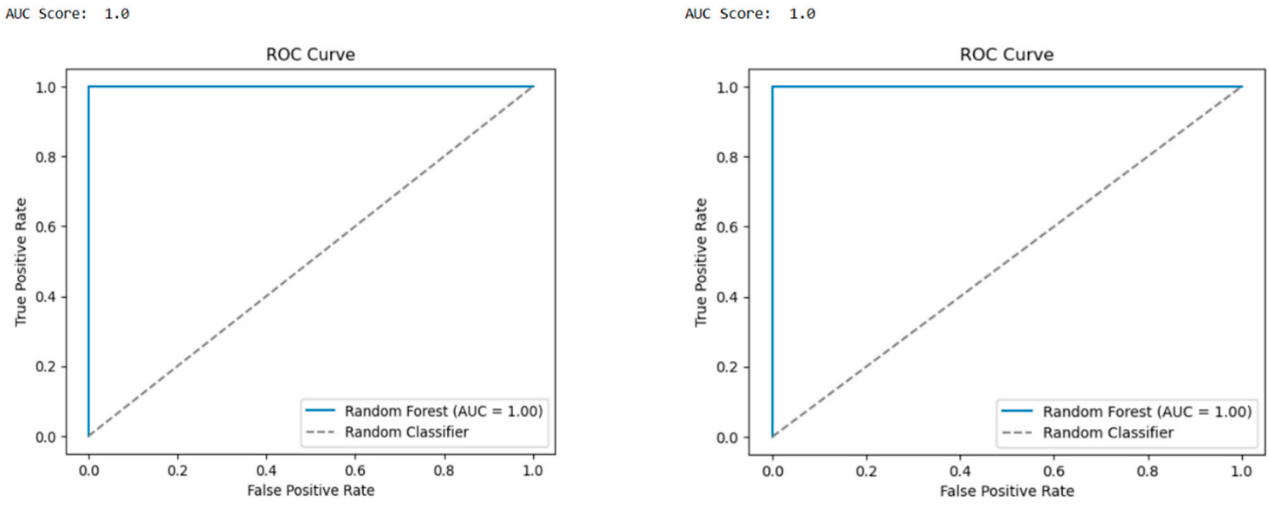
Table 5. Confusion matrix for TON-IOT (Case 1 and Case 2), BOTIOT (Case 3 and Case 4), and RF (15 features).

Case 1 (All Attacks)			**Case 2 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	49,613	10,506	Class 0 (Normal)	60,055	0
Class 1 (Attack)	11,451	48,430	Class 1 (Attack)	0	3945
Case 3 (All Attacks)			**Case 4 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	262,924	31,101	Class 0 (Normal)	385,326	0
Class 1 (Attack)	16,015	277,926	Class 1 (Attack)	0	26



(**a**) (**b**)

Figure 6. *Cont.*



(**a**) (**b**)

Figure 6. (**a**) ROC curve for TON-IOT all attacks (15 features), RF; (**b**) ROC curve for TON-IOT DDoS attacks (15 features), RF; (**c**) ROC curve for BOT-IOT all attacks (15 features), RF; (**d**) ROC curve for BOT-IOT DDoS attacks (15 features), RF.

Hence, all the models are proven effective in this case with the random forest algorithm. With a random forest model, a prediction is made by passing input data through each decision tree in the forest, and the forecast that receives the most votes across all the trees is chosen as the result. Random forest's key benefits include its high accuracy, resistance to noise and outliers, and capacity for handling big datasets with plenty of characteristics.

In order to choose features or comprehend the data, it can also offer estimates of feature relevance. The number of trees and the maximum depth of each tree are two hyper-parameters that may need to be adjusted because random forest can be computationally expensive. Additionally, it could struggle with unbalanced datasets and with data that have a lot of dimensions.

4.3. Naïve Bayes

A supervised machine learning method called naïve bayes is utilized for categorization problems. It is founded on the Bayes theorem, according to which the likelihood of a hypothesis—in this case, a class label—given the evidence (the input features) is inversely proportional to the likelihood of the evidence given the hypothesis, multiplied by the prior probability of the hypothesis. Given the class label, naïve bayes assumes that the input features are independent of one another. The approach learns the prior probability of each class label as well as the conditional probability of each input feature given each class label. A method known as maximum likelihood estimate is used for this.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (2)$$

The confusion matrix is given for TONIOT and BOTIOT, representing binary and multiple-class classification for each (Case 1 to Case 4) for 15 features (Table 6). In Case 1, which corresponds to multiple-class classification for the TON-IOT dataset, it is observed that the FP value is higher than the FN value, and hence there are chances that normal traffic can be detected as an attack. However, the TP and TN values do not really compensate for the FP and FN values as the margin between them is fairly small, so the accuracy would result in being minimal compared to other models and does prove to be not effective.

Similarly, in Case 3 with the BOT-IOT dataset for multi-class classification with a higher number of records than TON-IOT, there is a fairly high margin for TP and TN values compared to FP and FN values and hence it gives much higher accuracy. In cases 1 and 4,

with TON-IOT and BOT-IOT datasets for binary classification, both FP and FN values are zero, and therefore tend to obtain the maximum accuracy. It is observed that Figure 7b,d correspond to the binary classification for TON-IOT and BOT-IOT, respectively, and the steep ROC curve touches the top left corner, which makes these two cases very good models.

Table 6. Confusion matrix for TON-IOT (Case 1 and Case 2), BOTIOT (Case 3 and Case 4), and NB (15 features).

Case 1 (All Attacks)			**Case 2 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	34,056	25,997	Class 0 (Normal)	60,059	0
Class 1 (Attack)	96	32,060	Class 1 (Attack)	0	3941
Case 3 (All Attacks)			**Case 4 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	733,624	0	Class 0 (Normal)	385,300	0
Class 1 (Attack)	0	81	Class 1 (Attack)	0	121

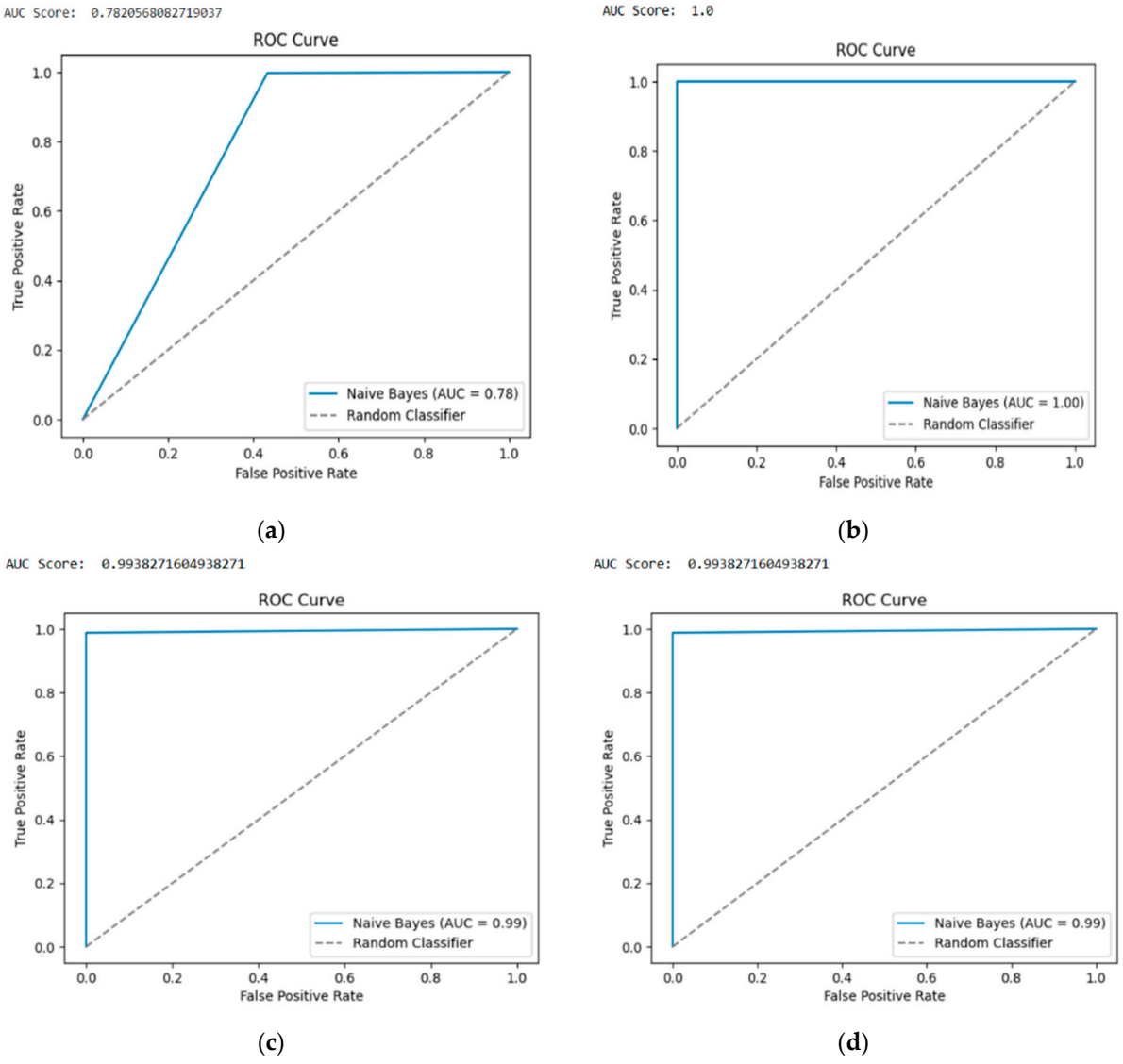


Figure 7. (**a**) ROC curve for TON-IOT all attacks (15 features), NB; (**b**) ROC curve for TON-IOT DDoS attacks (15 features), NB; (**c**) ROC curve for BOT-IOT all attacks (15 features), NB; (**d**) ROC curve for BOT-IOT DDoS attacks (15 features), NB.

In Figure 7a with the TON-IOT dataset for multiple-class classification, it is observed that the curve is closer to the diagonal, hence making it a very poor model compared to Figure 7c with BOT-IOT dataset for multiple-class classification. For Figure 7c, it is observed that the curve is almost touching the top left corner, and hence it is very much a better model with greater accuracy. Similarly, the AUC scores are variable with respect to their ROC curves as observed below.

By computing the posterior probabilities of each class label given the input features and selecting the class label with the highest probability as the prediction, the model may be used to make predictions on new data after it has been trained. Naïve bayes' key benefits are its ease of use, effectiveness, and capacity for high-dimensional data handling.

4.4. Artificial Neural Network (ANN)

A form of machine learning technique called artificial neural networks (ANN) is based on the structure and operation of the human brain. A number of tasks, including classification, regression, and prediction, are carried out using ANNs. An ANN is made up of layers of interconnected processing nodes called neurons at the highest level. The output layer generates the final prediction or output after receiving input data from the input layer. There may be one or more hidden layers that carry out intermediary computations between the input and output layers.

Each neuron in an ANN receives input values, processes them, and generates an output. A neuron computes by adding a bias term, multiplying the input values by weights, and then passing the output through an activation function. The parameters' weights and biases are acquired by training, and the activation function chooses the neuron's output depending on the weighted sum of its inputs.

In order to reduce the error between the projected output and the actual output, the ANN modifies the weights and biases during training. The confusion matrix is given below for TON-IOT and BOT-IOT, representing binary and multiple-class classification for each (Case 1 to Case 4) for 15 features (Table 7). It is observed that in Case 1 and Case 3, the FP value is zero and the FN value is one, so this makes the model less robust for multiple-class classification compared to Case 2 and Case 4, with FP and FN values as zero. Therefore, it obtains the maximum accuracy.

Table 7. Confusion matrix for TON-IOT (Case 1 and Case 2), BOTIOT (Case 3 and Case 4), and ANN (15 features).

Case 1 (All Attacks)			**Case 2 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	60,119	0	Class 0 (Normal)	59,988	0
Class 1 (Attack)	1	59,880	Class 1 (Attack)	0	4012
Case 3 (All Attacks)			**Case 4 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	292,999	0	Class 0 (Normal)	142,529	11,245
Class 1 (Attack)	1	293,888	Class 1 (Attack)	4804	149,682

The subsequent ROC curves are given for BOT-IOT and TON-IOT with 15 features for multiple-class and binary classifications. When the steepness of the ROC curve reaches the top left corner then it is considered an excellent model, hence, in this case Figure 8b,d touch the top left corner. However, Figure 8a,c are almost touching the top left, making them a little less effective as models but still optimal. The AUC scores provide a summary measure of the model's discriminatory power; a higher AUC indicates a more effective DDoS attack detection model. Hence, Figure 8a,c give 0.99, but Figure 8b,d give 1.00 as the AUC score.

ANN is significant for detecting DDoS attacks in IoT devices due to its ability to analyze complex patterns and handle high-dimensional, heterogeneous data. It learns and adapts to evolving attack patterns, ensuring accurate detection of emerging DDoS

techniques in IoT environments. ANN's parallel processing enables real-time detection, mitigating potential damages and maintaining IoT service availability.

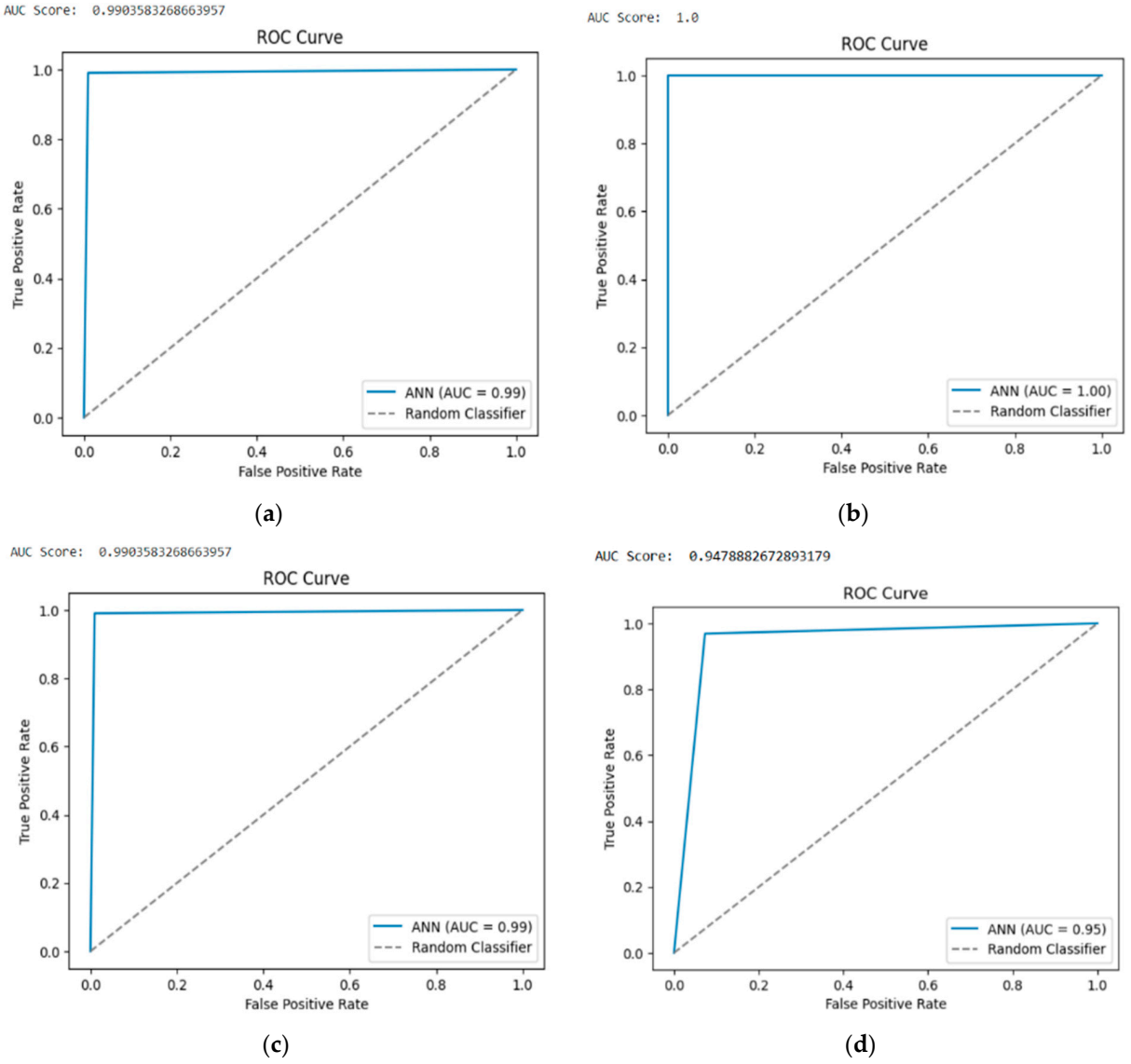


Figure 8. (**a**) ROC curve for TON-IOT all attacks (15 features), ANN; (**b**) ROC curve for TON-IOT DDoS attacks (15 features), ANN; (**c**) ROC curve for BOT-IOT all attacks (15 features), ANN; (**d**) ROC curve for BOT-IOT DDoS attacks (15 features), ANN.

Combining ANN with other techniques improves detection accuracy and reduces false positives. By reducing reliance on signature-based approaches, ANN enables the detection of zero-day attacks and novel vectors. It can be implemented on IoT devices or in the cloud, facilitating distributed detection and response mechanisms. Overall, ANN provides a robust and adaptable approach, enhancing the resilience and security of IoT networks against DDoS attacks.

$$Y = w_1 X_l + w_2 X_2 + b \tag{3}$$

4.5. K Nearest Neighbor

The supervised machine learning method k nearest neighbor (KNN) is used for classification and regression tasks. Since it is non-parametric, it does not assume anything about the distribution of the data at its core. In its most basic form, KNN predicts the class

or value of a test point by locating the k nearest neighbors to it in the training data and using their characteristics to determine that test point's characteristics. Euclidean distance is a widely used distance metric that may be used to determine the separation between two places. KNN predicts a test point's class for classification tasks based on the majority class of the k's closest neighbors. For regression tasks, KNN predicts the value of a test point based on the average value of the k nearest neighbors. The value of k is a hyperparameter that can be chosen by the user. While a bigger value of k can result in smoother choice boundaries but may be less sensitive to local patterns in the data, a smaller value of k can result in more complex decision boundaries and may be more susceptible to noise in the data.

$$(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (4)$$

The confusion matrix is given below for TON-IOT and BOT-IOT representing binary and multiple-class classification for each (Case 1 to Case 4) for 15 features (Table 8). It is observed that in all cases the FP and FN values are zero. Therefore, maximum accuracies will be achieved and this determines that the proposed model is robust. The subsequent ROC curves are given for BOT-IOT and TON-IOT with 15 features for multiple-class and binary classifications. When the steepness of the ROC curve reaches the top left corner then it is considered an excellent model, hence, in this case all of them are very good models (Figure 9). AUC scores provide a summary measure of the model's discriminatory power; a higher AUC indicates a more effective DDoS attack detection model. Hence, all of the models are proven effective in this case, with KNN. KNN is a significant algorithm for detecting DDoS attacks in IoT devices.

Table 8. Confusion matrix for TON-IOT (Case 1 and Case 2), BOTIOT (Case 3 and Case 4), and KNN (15 features).

Case 1 (All Attacks)			**Case 2 (DDoS Attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	59,925	0	Class 0 (Normal)	60,055	0
Class 1 (Attack)	0	32,284	Class 1 (Attack)	0	3945
Case 3 (All Attacks)			**Case 4 (DDoS attacks)**		
True/Predict	Class 0 (Normal)	Class 1 (Attack)	True/Predict	Class 0 (Normal)	Class 1 (Attack)
Class 0 (Normal)	733,603	1	Class 0 (Normal)	385,326	0
Class 1 (Attack)	0	101	Class 1 (Attack)	0	95

KNN offers an intuitive approach to identifying anomalous patterns in IoT network traffic based on the similarity principle. By measuring distances between data points, KNN can classify traffic as normal or malicious, making it effective in detecting DDoS attacks by identifying deviations from normal patterns. KNN can be trained using labeled datasets, improving detection accuracy over time. Its versatility allows integration with other algorithms, enhancing overall detection performance. In summary, KNN provides an effective and interpretable approach for detecting DDoS attacks in IoT devices, enhancing security and mitigating potential disruptions caused by malicious activities.

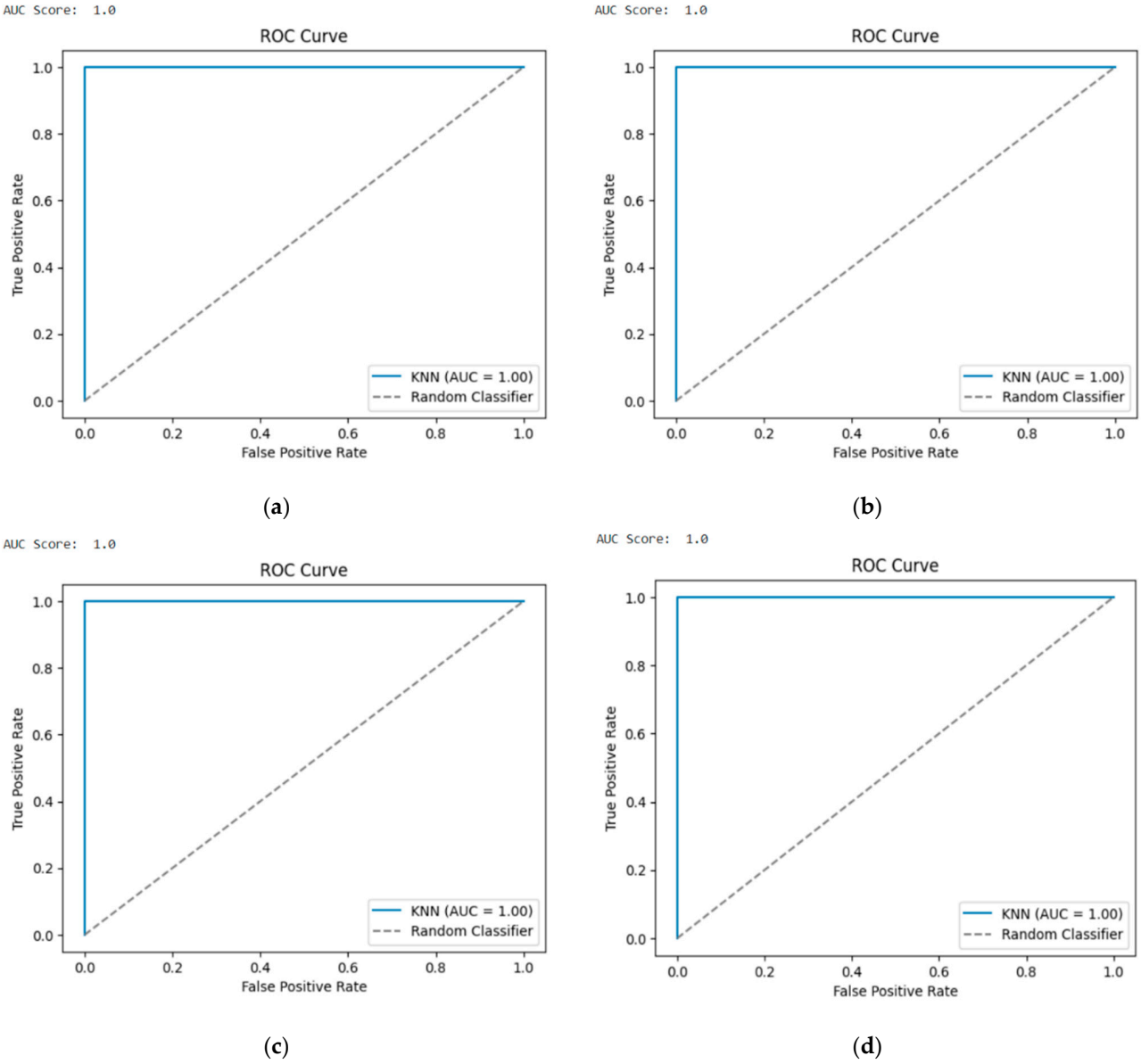


Figure 9. (**a**) ROC curve for TON-IOT all attacks (15 features), KNN; (**b**) ROC curve for TON-IOT DDoS attacks (15 features), KNN; (**c**) ROC curve for BOT-IOT all at-tacks (15 features), KNN; (**d**) ROC curve for BOT-IOT DDoS attacks (15 features), KNN.

4.6. Evaluation Using Metrics

Many performance metrics are used to evaluate ML- and DL-based IDSs. The research aims to compare the performance when feature selection is used and when it is not. Accuracy is the key measure of evaluation.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

Other metrics include recall, precision, F1 score, training time, prediction time, and total time. F1 score takes both FP and FN into account, unlike recall and precision.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{F1 Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{8}$$

Discussing the evaluation metrics used for the proposed model, precision is a crucial metric in detecting DDoS attacks in IoT devices, as it measures the accuracy of identifying true positives among instances labelled as attacks. It reflects the system's ability to minimize false positives, which occur when normal traffic is wrongly classified as DDoS attacks. High precision indicates a low rate of false positives, ensuring that legitimate IoT services are not disrupted by unnecessary alarms or resource allocation. Balancing precision with other performance metrics is essential for an effective detection system that accurately identifies DDoS attacks while minimizing false positives. Overall, precision plays a significant role in securing IoT networks against DDoS attacks by accurately identifying true positives and reducing false positives.

Recall, also known as sensitivity or true positive rate, is crucial for detecting DDoS attacks in IoT devices. It quantifies the system's capability to correctly identify actual DDoS attacks from all the instances that are truly positive. In other words, recall indicates the system's effectiveness in minimizing false negatives, which occur when DDoS attacks go undetected or are misclassified as normal traffic. A high recall rate signifies a low occurrence of false negatives, ensuring that a significant number of DDoS attacks are accurately identified and reducing the risk of leaving malicious activities undetected. In summary, recall plays a critical role in DDoS attack detection for IoT devices by measuring the system's ability to capture actual attacks. Achieving a high recall rate reduces false negatives and enables comprehensive detection, enabling prompt mitigation of attacks. This is a system that safeguards IoT networks against DDoS threats.

The F1 score is a very important metric for evaluating the performance of a DDoS attack detection system in IoT devices. It combines precision and recall, providing a balanced assessment of the system's ability to accurately identify DDoS attacks and capture the majority of actual attacks. The F1 score is particularly useful when dealing with class imbalance between normal traffic and DDoS attacks. A high F1 score indicates a robust and reliable detection system that achieves a balance between precision and recall, minimizing false alarms and undetected attacks. Overall, the F1 score is critical for evaluating the effectiveness of DDoS attack detection in IoT devices, ensuring a well-rounded approach that minimizes both false positives and false negatives, thereby securing IoT networks effectively.

Accuracy is a crucial metric for assessing the performance of DDoS attack detection in IoT devices. It measures the system's correctness by evaluating the proportion of correctly classified instances, including true positives and true negatives. A high accuracy rate indicates the system's effectiveness in distinguishing between DDoS attacks and normal traffic, reducing false positives and false negatives. It demonstrates the system's ability to make accurate decisions and maintain IoT service availability. In summary, accuracy is a valuable metric for evaluating DDoS attack detection in IoT devices as it provides an overall measure of correct classification. A high accuracy rate reflects the system's ability to identify both DDoS attacks and normal traffic accurately.

From the discussion based on the relevance of these evaluation metrics, when the naïve bayes algorithm under the TON-IOT dataset is taken into consideration for multiple class classification, it is observed that there is a significant difference between the normal and attack traffic metrics, which gives a significant reduction in accuracy value. Similarly, for logistic regression for both TON-IOT and BOT-IOT datasets under multiple class classification, there is a very minimal difference between the normal and attack traffic metrics.

Based on the results (Table 9), the TON-IOT dataset with the 15 best features for multiple-class classification shows poor results for naïve bayes, with an accuracy of 77%.

With respect to time, ANN gives the best results. It detects an attack with 99% accuracy, with training time of 10.23 ms and prediction time of 16.456 ms. However, taking accuracy into consideration, random forest is the best model for multiple-class classification with a result of 100%, with training and prediction time of 56.404 ms and 0.795 ms, respectively. Meanwhile, under the same dataset with 15 features for binary classification, all models give the same accuracy of 100%. However, with respect to time, naïve bayes is considered the best, with training and prediction time of 0.3554 ms and 0.0334 ms, respectively, with 100% binary classification accuracy.

Table 9. Evaluation using metrics for TON-IOT (15 features).

		Accuracy	Class	Precision	Recall	F1 Score	Training Time (ms)	Predict Time (ms)
Case 1: TON-IOT with All Attacks	Logistic Regression	82%	0	0.81	0.83	0.82	8.3514	0.0319
			1	0.82	0.81	0.81		
	Random Forest	100%	0	1.00	1.00	1.00	56.404	0.795
			1	1.00	1.00	1.00		
	Naïve Bayes	77%	0	0.75	0.81	0.78	0.2776	0.03385
			1	0.79	0.73	0.76		
	ANN	99%	0	0.99	0.99	0.99	10.23	16.456
			1	0.99	0.99	0.99		
	KNN	98%	0	0.99	0.97	0.98	2.208	26.923
			1	0.95	0.99	0.97		
Case 2: TON-IOT with DDoS	Logistic Regression	100%	0	1.00	1.00	1.00	18.882	0.0186
			1	1.00	1.00	1.00		
	Random Forest	100%	0	1.00	1.00	1.00	14.3046	0.3173
			1	1.00	1.00	1.00		
	Naïve Bayes	100%	0	1.00	1.00	1.00	0.3554	0.0334
			1	1.00	1.00	1.00		
	ANN	100%	0	1.00	1.00	1.00	52.1	2.9
			1	1.00	1.00	1.00		
	KNN	100%	0	1.00	1.00	1.00	0.15322	176.45
			1	1.00	1.00	1.00		

The BOT-IOT dataset with the 15 best features for multiple-class classification gives 100% accuracy for all the models except for ANN and logistic regression, with 99% and 92% accuracy, respectively. Clearly logistic regression is the least preferred model with respect to time, however, it is an efficient model with respect to time, with training and prediction times of 350.867 ms and 0.0334 ms, respectively. Here, once again naïve bayes is considered the best among the five algorithms with respect to time and accuracy together (Table 10). It takes 2.9326 ms to train and 0.3043 ms to predict an attack with 100% accuracy. Moreover, under the same dataset with 15 features for binary classification, all the models get the same accuracy of 100%, except ANN with 95%. Here again naïve bayes is considered the best with respect to time of completion. It takes 1.5244 ms to train and 0.1685 ms to predict an attack with 100% accuracy.

When this research is compared with previously existing state-of the-art models, the proposed model receives an accuracy of 100% with just 15 features compared to other previous existing models, with NB and RF being the best models. Most importantly, the research has implemented time as an important factor in detecting DDoS attacks. The model is proven to be efficient and optimal compared to other previously existing models in multiple aspects (Table 11).

According to research [25], training time and prediction time are implemented in a TON-IOT telemetry dataset, where the training is performed on each set of IoT devices as well as overall binary and multi-class classification. The best model, CART, has given 6 s training and testing time, whereas the proposed model has given <1 s training and testing

time combined. This research has proven to be much more effective in terms of accuracy and the time factor for detecting the DDoS attacks in IoT devices.

Table 10. Evaluation using metrics for BOT-IOT (15 features).

		Accuracy	Class	Precision	Recall	F1 Score	Training Time (ms)	Predict Time (ms)
Case 1: BOT-IOT with All Attacks	Logistic Regression	92%	0	0.94	0.90	0.92	5.765	0.1046
			1	0.90	0.95	0.92		
	Random Forest	100%	0	1.00	1.00	1.00	651.873	2.9125
			1	1.00	1.00	1.00		
	Naïve Bayes	100%	0	1.00	1.00	1.00	2.9326	0.3043
			1	1.00	0.99	0.99		
	ANN	99%	0	0.99	0.99	0.99	569.412	59.432
			1	0.99	0.99	0.99		
	KNN	100%	0	1.00	1.00	1.00	1.64415	3325.77
			1	1.00	1.00	1.00		
Case 2: BOT-IOT with DDoS	Logistic Regression	100%	0	1.00	1.00	1.00	16.912	0.04588
			1	1.00	1.00	1.00		
	Random Forest	100%	0	1.00	1.00	1.00	194.65	1.6478
			1	1.00	1.00	1.00		
	Naïve Bayes	100%	0	1.00	1.00	1.00	1.5244	0.1685
			1	1.00	1.00	1.00		
	ANN	95%	0	0.97	0.93	0.95	350.867	34.567
			1	0.93	0.97	0.95		
	KNN	100%	0	1.00	1.00	1.00	0.77648	940.176
			1	1.00	1.00	1.00		

Table 11. Comparison between existing state-of-art models and the proposed model.

Citation	Year	BOT-IOT	TON-IOT	Model	Binary Classification	Multiple-Class Classification	Feat. Selection	No. of Features	K-Fold	Acc. (%)	Time
[19]	2022	YES	-	RF	YES	-	-	46	-	99.90	-
[20]	2022	YES	-	NB	YES	-	PCA	25	-	99.40	-
[21]	2022	YES	-	LSTM	YES	-	-	46	-	96.30	-
[22]	2022	YES	YES	ANN, LSTM	YES	YES	ETC(20)	20	YES	99.00	-
[25]	2020	-	YES	CART	YES	-	-	N/A	YES	88.00	YES
Proposed Model	**2023**	**YES**	**YES**	**NB, RF**	**YES**	**YES**	**ETC(15)**	**15**	**YES**	**100.00**	**YES**

A stringent validation process was employed to evaluate the effectiveness of the proposed intrusion detection system for multiple-class classification. Specifically, a five-fold cross-validation technique was utilized to ensure a robust assessment of the model's performance. This method involved partitioning the dataset into five equal subsets or "folds". The model was then trained on four of these folds and validated on the remaining one, iterating this process five times so that each fold served as the validation set once. By employing five-fold cross-validation, the research ensured that the results were less susceptible to variations in the data, providing a more reliable and unbiased estimation of the model's ability to generalize to unseen data. This approach helped in reducing overfitting and offered a comprehensive view of how the model performed across different segments of the dataset, thereby adding rigor and credibility to the findings related to multiple-class classification of other attacks within lightweight IoT networks.

5. Case Study: Application of IDS in Energy and Utilities IoT

The paper has primarily focused on assessing the effectiveness of the IDS framework as an IoT network security algorithm. Its suitability has been theoretically validated through experimentation on the TON-IOT and BOT-IOT datasets, and through comparisons

with other approaches. This section delves into the practical implementation of the IDS framework in a real-world application, providing readers with a more comprehensive understanding of its advantages and impact. Energy and utilities IoT integrate sensors and devices with user-interactive software to optimize energy distribution. They offer applications like smart grid monitoring and asset management. While they enhance efficiency, security is a key concern due to sensitive data. By incorporating an intrusion detection system (IDS), the energy and utilities IoT network can detect and prevent attacks, safeguarding devices and infrastructure. This integration ensures data protection, mitigates risks, and enables secure energy services. This paper provides research conducted in the energy and utilities IoT sector. The integration of renewable energy and optimization of energy usage are vital for sustainable energy transitions and combating climate change.

The Internet of Things (IoT) offers numerous applications in the energy sector, including energy supply, transmission, distribution, and demand management. By leveraging IoT, energy efficiency can be improved and the use of renewable energy consumption can be reduced. This paper presents a review of the existing literature on the application of IoT in energy systems, with a focus on smart grids [26]. Additionally, it discusses enabling technologies such as cloud computing and data analysis platforms. The challenges associated with deploying IoT in the energy sector, including privacy and security concerns, are also examined along with potential solutions like blockchain technology. This survey serves as a valuable resource for energy policymakers, energy economists, and managers, providing them with an understanding of the role of IoT in optimizing energy systems.

This paper explores the commonly utilized technologies, protocols, and architectures in the energy and utilities sector. Sources can be increased, and it also examines the environmental impact, the application of sensors, and it highlights the challenges encountered in energy and utilities IoT. (Figure 10) This research reveals that energy and utilities IoT applications typically involve three key components or stages:

- Data acquisition through sensors and energy monitoring devices.
- Connectivity and data transmission across the network.
- Utilization of software for data storage, processing, security, visualization, and analysis.

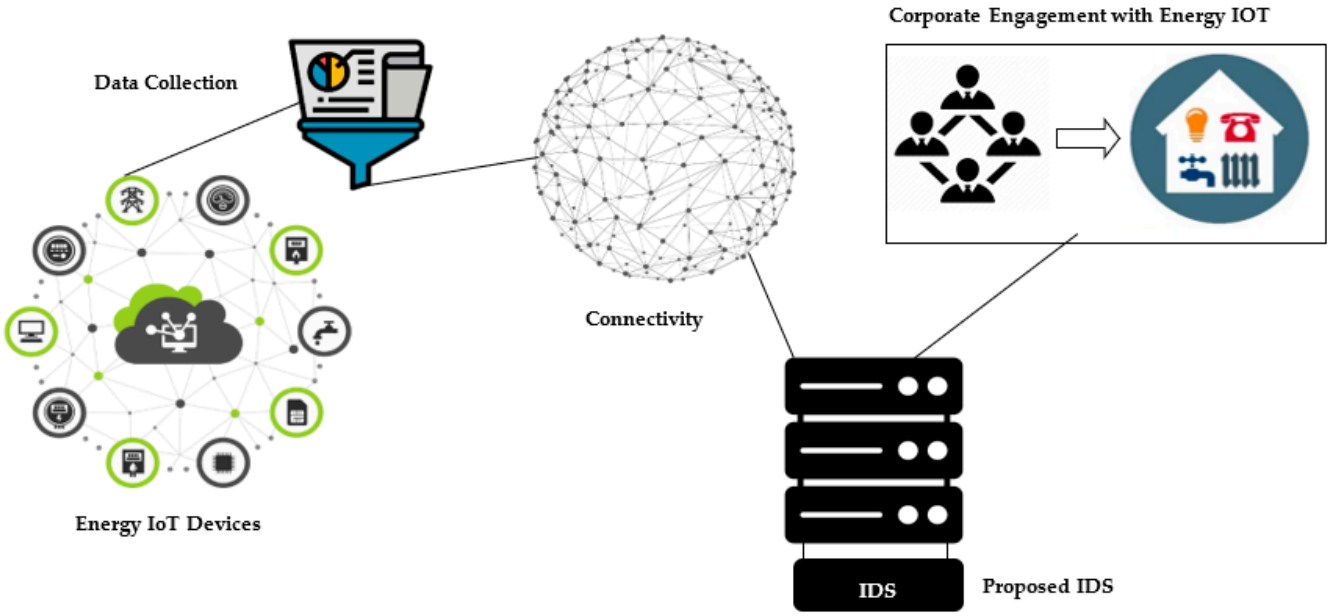


Figure 10. Application of the proposed IDS model in the energy and utilities IoT sector.

In the context of energy and utilities, these components enable the collection of relevant data, establish seamless communication, and facilitate the effective management and analysis of energy-related information.

The IDS aims to enhance network security and should be integrated into the fourth step of energy and utilities IoT implementation. IoT devices, such as smart meters, sensors, and monitoring devices, are deployed throughout the energy infrastructure. These devices collect data related to energy consumption, grid performance, asset health, environmental conditions, and other utility parameters. They continuously gather and transmit this data in real time or at regular intervals. The data collected by IoT devices is transmitted to centralized systems or cloud platforms for storage, analysis, and processing. These platforms provide the necessary infrastructure for handling the vast amount of data generated by the IoT devices. Cloud-based solutions offer scalability, data security, and accessibility for energy and utility companies.

IoT devices in the energy and utilities industry are designed to be connected to a network infrastructure. They typically utilize wireless communication protocols, such as cellular networks, Wi-Fi, LANs/WANs, etc., to transmit data to centralized systems or cloud platforms. This connectivity enables seamless data transmission and remote monitoring. At each designated node, the collected and transmitted data from every device should be evaluated by the IDS, similar to the testing process with BOT-IOT and TON-IOT records. The nodes can be programmed to raise an alarm and discard the data packet if the IDS identify it as an attack. IoT devices empower corporate companies to actively manage their energy consumption. Through the utilization of smart building solutions and IoT-enabled devices, companies can monitor and control energy usage, establish energy preferences, and receive personalized recommendations. This enables them to optimize energy efficiency, reduce costs, and contribute to sustainability efforts. By leveraging IoT technologies, corporate entities can take proactive measures in energy management, enhancing operational efficiency and environmental responsibility. The IDS is lightweight and can protect network nodes without disrupting their operations, even with limited computational power. Its high accuracy ensures reliable results, enabling automated actions when an alarm is triggered. By conducting real-time attack detection at each node, the IDS efficiently safeguards energy and utilities IoT networks [26]. One of the primary challenges faced by the energy IoT sector is ensuring robust security measures. The integration of IDS can effectively address this challenge and provide energy administrators with the confidence to embrace proposed IoT frameworks in energy systems. By implementing IDS, the sector can enhance the security posture of IoT devices and infrastructure, fostering trust and encouraging widespread adoption. This, in turn, leads to significant advancements in operational efficiency, grid reliability, and ultimately contributes to the overall improvement of energy services and public welfare.

6. Conclusions

The study in the paper characterizes lightweight IoT networks as being established by devices with few computer resources, such as reduced battery life, processing power, memory, and, more critically, minimal security and protection, which are easily vulnerable to DDoS attacks and propagating malware. Numerous scholars have studied the notion that protecting lightweight IoT networks urgently requires improving intrusion detection systems. This manuscript proposes a compact intrusion detection system that blends machine learning classifiers with a fresh approach to data pre-processing. The study provides a number of classifier types to build lightweight intrusion detection systems that are ideally suited for defense against DDoS attacks in IoT networks. The dataset from the TON-IOT from UNSW Network and BOT-IOT are used. The dataset is used to produce DDoS assault samples. Binary and multiple-class classifications of the experiment for the DDoS attacks and all attacks, respectively, are undertaken in TON-IOT and BOT-IOT datasets. The attack classes of the TON-IOT and the normal attacks of BOT-IOT are imbalanced. In this study, multiple iterations of the SMOTE technique are used to balance classes in the experiments using the TON-IOT and BOT-IOT dataset.

Binary classification and multiple-class classification between DDoS and normal traffic evaluated the performance of five ML methods (RF, LR, KNN, ANN, NB) based on the

accuracy, recall, precision, F1 score, training time, prediction time, and total time. The datasets used are TON-IOT network train/test dataset and BOT-IOT dataset. The performance analysis was done with 15 features of both BOT-IOT and TON-IOT datasets using ExtraTreeClassifier. A comparison between existing state-of-the-art models and the proposed model was made. It was observed that, in the TON-IOT dataset for multiple-class classification for 15 features, random forest is the best algorithm, with 100% accuracy, whereas in binary classification naïve bayes is the best, with 100%. In the BOT-IOT dataset, it was observed that for both multiple-class and binary classification naïve bayes obtains an accuracy of 100%. Comparison was made between the existing state-of-art models and the proposed model.

The future work for this research is going to be as follows. Firstly, implementing the same process on a different IoT-related dataset and testing the existing method. Secondly, implementing deep learning techniques to potentially create better models to detect attacks. Thirdly, creating a front-end application to detect any attacks that would come across any IoT devices.

Author Contributions: Conceptualization, S.S. and B.M.; Methodology, S.S. and B.M.; Software, B.M.; Validation, R.M. and P.P.; Formal analysis, R.M.; Investigation, R.M. and P.P.; Resources, B.M.; Writing—original draft, S.S. and B.M.; Writing—review & editing, S.S., R.M. and P.P.; Supervision, R.M. and P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mahadik, S.; Pawar, P.M.; Muthalagu, R. Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT). *J. Netw. Syst. Manag.* **2023**, *31*, 2. [CrossRef]
2. Hussain, F.; Abbas, S.G.; Husnain, M.; Fayyaz, U.U.; Shahzad, F.; Shah, G.A. IoT DoS and DDoS Attack Detection using ResNet. In Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5–7 November 2020; pp. 1–6. [CrossRef]
3. Esmaeili, M.; Goki, S.H.; Masjidi, B.H.K.; Sameh, M.; Gharagozlou, H.; Mohammed, A.S. ML-DDoSnet: IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8481452. [CrossRef]
4. Badamasi, U.M.; Khaliq, S.; Babalola, O.; Musa, S.; Iqbal, T. A Deep Learning based approach for DDoS attack detection in IoT-enabled smart environments. *Mach. Learn.* **2020**, *8*, 93–99.
5. Aysa, M.H.; Ibrahim, A.A.; Mohammed, A.H. IoT Ddos Attack Detection Using Machine Learning. In Proceedings of the 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 22–24 October 2020; pp. 1–7. [CrossRef]
6. Roopak, M.; Tian, G.Y.; Chambers, J. An Intrusion Detection System against DDoS Attacks in IoT Networks. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 562–567. [CrossRef]
7. Sriram, S.; Vinayakumar, R.; Alazab, M.; Soman, K.P. Network Flow based IoT Botnet Attack Detection using Deep Learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 189–194. [CrossRef]
8. Jia, Y.; Zhong, F.; Alrawais, A.; Gong, B.; Cheng, X. FlowGuard: An Intelligent Edge Defense Mechanism against IoT DDoS Attacks. *IEEE Internet Things J.* **2020**, *7*, 9552–9562. [CrossRef]
9. Hussain, F.; Abbas, S.G.; Pires, I.M.; Tanveer, S.; Fayyaz, U.U.; Garcia, N.M.; Shah, G.A.; Shahzad, F. A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks. *IEEE Access* **2021**, *9*, 163412–163430. [CrossRef]
10. Alzahrani, R.J.; Alzahrani, A. Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic. *Electronics* **2021**, *10*, 2919. [CrossRef]
11. Kumar, P.; Bagga, H.; Netam, B.S.; Uduthalapally, V. SAD-IoT: Security Analysis of DDoS Attacks in IoT Networks. *Wirel. Pers. Commun.* **2022**, *122*, 87–108. [CrossRef]
12. Pokhrel, S.; Abbas, R.; Aryal, B. IoT Security: Botnet detection in IoT using Machine learning. *arXiv* **2021**, arXiv:2104.02231.

13. Malik, M.; Dutta, M. Feature Engineering and Machine Learning Framework for DDoS Attack Detection in the Standardized Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 8658–8669. [CrossRef]
14. Gaur, V.; Kumar, R. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. *Arab. J. Sci. Eng.* **2022**, *47*, 1353–1374. [CrossRef]
15. Chopra, A.; Behal, S.; Sharma, V. Evaluating Machine Learning Algorithms to detect and classify DDoS attacks in IoT. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021.
16. Amrish, R.; Bavapriyan, K.; Gopinaath, V.; Jawahar, A.; Kumar, C.V. DDoS Detection using Machine Learning Techniques. *J. IoT Soc. Mob. Anal. Cloud* **2022**, *4*, 24–32. [CrossRef]
17. Motylinski, M.; MacDermott, Á.; Iqbal, F.; Shah, B. A GPU-based machine learning approach for detection of botnet attacks. *Comput. Secur.* **2022**, *123*, 102918. [CrossRef]
18. Mihoub, A.; Fredj, O.B.; Cheikhrouhou, O.; Derhab, A.; Krichen, M. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Comput. Electr. Eng.* **2022**, *98*, 107716. [CrossRef]
19. Mohmand, M.I.; Hussain, H.; Khan, A.A.; Ullah, U.; Zakarya, M.; Ahmed, A.; Raza, M.; Rahman, I.U.; Haleem, M. A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks. *IEEE Access* **2022**, *10*, 21443–21454. [CrossRef]
20. Adefemi Alimi, K.O.; Ouahada, K.; Abu-Mahfouz, A.M.; Rimer, S.; Alimi, O.A. Refined LSTM Based Intrusion Detection for Denial-of-Service Attack in Internet of Things. *J. Sens. Actuator Netw.* **2022**, *11*, 32. [CrossRef]
21. Zeeshan, M.; Riaz, Q.; Bilal, M.A.; Shahzad, M.K.; Jabeen, H.; Haider, S.A.; Rahim, A. Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and BOT-IOT Data-Sets. *IEEE Access* **2022**, *10*, 2269–2283. [CrossRef]
22. Khanday, S.A.; Fatima, H.; Rakesh, N. Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. *Expert Syst. Appl.* **2023**, *215*, 119330. [CrossRef]
23. Otoum, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning–based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3803. [CrossRef]
24. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on TON-IOT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
25. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
26. Hossein Motlagh, N.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of Things (IoT) and the Energy Sector. *Energies* **2020**, *13*, 494. [CrossRef]