
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 950–963, DOI 10.12694/scpe.v26i2.4059

DDOS ATTACK DETECTION AND PERFORMANCE ANALYSIS IN IOT NETWORK USING MACHINE LEARNING APPROACHES

DEVPRIYA PANDA*, NEELAMADHAB PADHY† AND KAVITA SHARMA‡

Abstract. The Internet is the most common connecting tool for devices, such as computers, mobile phones, smart watches, etc. These devices communicate with designated servers to provide information. Here we refer to the system that connects numerous autonomous devices known as the Internet of Things (IoT). As the devices are of diverse categories and sometimes very small, it becomes challenging to provide comprehensive security to those in need. However, the sensors on the IoT collect huge amounts of data and the huge network becomes an attractive target for assaulters. One of the several assaults on IoT is Distributed Denial of Service (DDoS). Machine learning can play a crucial role in identifying these attacks in the IoT because of its ability to analyse large amounts of data. Machine learning models can learn the pattern of legitimate traffic and later identify malicious packets that deviate from the learned pattern. Classification techniques can distinguish malicious packets from genuine ones based on several attributes associated with them. This work uses classification techniques such as Random Forest, Gradient Boosting, and XgBoost to determine the malicious packets in traffic. The analysis shows that balancing techniques such as SMOTE and ADASYN are vital in improving the performance of techniques.

Key words: ADASYN, DDoS, Gradient Boost, IoT, Random Forest, XgBoost

1. Introduction. The Internet of Things provides ubiquitous computing power across a network of devices. The devices in an IoT network can be any object, referred to as a 'thing.' It can be a sensor, an embedded device, a mobile phone, etc. The basic architecture of an IoT network can be visualized as a combination of four layers [26]. A network is built to communicate among these layers and the server to provide information and help us reach a universal objective [6]. IoT is spreading across various domains, but the issue of security is a significant concern. A small part of any extensive IoT network can be targeted to launch a DDoS attack. By overloading the network's bandwidth, a group of vulnerable IoT nodes can cripple a full-scale network with powerful servers.

In IoT, the nodes have minimal resources, whether computing power, memory, or any other. Hence, the nodes in IoT networks are very vulnerable to attacks. Attackers find the devices attached to the IoT network attractive. Various types of attacks are possible in an IoT network. One of these is 'Distributed Denial of Service' (DDoS). It can be performed using botnets such as Mirai, launched in 2016 [13]. In the same year, Mirai affected the heating systems in some buildings in Finland and crashed those systems [3]. The greater the number of vulnerable devices, the greater the probability of creating botnets for the attackers. The magnitude of the DDoS attack is proportionate to the botnet's size. When a huge number of nodes participate in a botnet, the fierceness of the attack becomes increasingly dreadful. The bots generate false requests for the server, making it busy and unable to serve genuine requests. As a result, genuine users keep waiting for the service for an uncertain period.

In a UDP-based DDoS assault, the attacker transmits many UDP packets to the victim device. After receiving, the system finds the appropriate application to service the packets. But the system getsa huge number of packets, and in attempting to service these, the server becomes unavailable for other clients, launching a DDoS attack.The second case describes how the LDAP protocol can launch a DDoS attack. In this case, the target's IP address is spoofed. An attacker pretends to be the intended victim and sends a packet requesting service. When the server tries to respond, it transmits the responses to the target machine, as spoofing forces the server to think the attacker is legitimate. Portmap attacks are launched by exploiting the vulnerability

*GIET University, Odisha, India (Corresponding author, devpriya.panda@giet.edu).
†GIET University, Odisha, India (dr.neelamadhab@giet.edu).
‡Galgotias College of Engineering and Technology, Greater Noida, India (kavitasharma_06@yahoo.co.in).

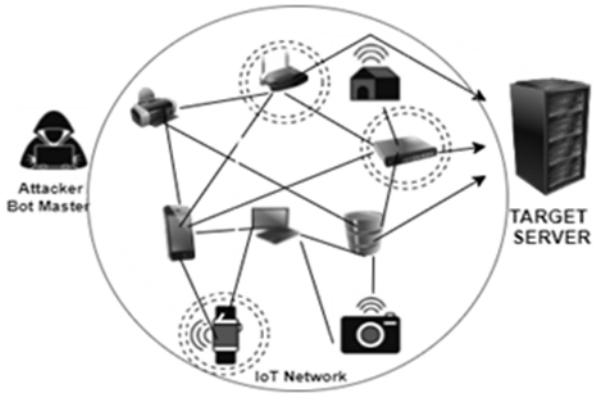

Fig. 1.1: DDoS Architecture.

of the port mapper. In port mapping, the request is expanded to include a significant response. This feature makes it suitable for DDoS attacks as it can increase the fierceness of the attack. Machine learning classification techniques are extensively used by various researchers to segregate one category of data from other categories [11]. The same concept can be applied to identify malicious packets in a chunk of packets transferred over an IoT network. Based on the DDoS attacks, several datasets have been prepared. CICDDoS2019 is one of the most recent datasets, describing several aspects [9].

In this work, this dataset is considered to make the most of it. This work analyses three types of models to detect DDoS attacks. Random forest has been used in some earlier works; the 'XgBoost' model has been considered in this work in addition to gradient Boosting and the formerly mentioned model. Also in the pre-processing, in addition to the 'Random Over Sampling' method, SMOTE and ADASYN sampling models were considered.

After going through related literature in Section 2, the objective of this investigation is defined in Section 3. Then, the workflow is explained in Section 4. After the workflow, the results are being analysed in Section 5. The conclusion and the future scopes are mentioned at the end in Section 6.

2. Literature Survey. In an IoT environment, the security mechanisms available at the nodes are negligible. Hence, the vulnerability of these nodes is exploitable. The nodes in IoT networks are deprived of resources compared to standard computing devices because of the constraint of power and as those are designed for specific purposes. DDoS is one of the most frequent attacks performed on different IoT networks. Previous works in the context under discussion are studied and are described below.

Jain et al. in [9] suggested an ensemble of different machine learning techniques to get better performance in detecting DDoS attacks. A voting mechanism at the run time has been proposed to choose the best method for identifying the malicious data packets in the traffic. Naïve Base, Random Forest, K Nearest Neighbour, and Support Vector Machine methods have been used for classifying affected packets. But instead of considering all those at a time, the authors' proposed voting system chooses the best result.

In this work [12], the concept of Cyber Security is discussed by Khari et. al. with an overview alongside the history of Cyber Security that has evolved from information security to encompass individual confidential data. It defines the human being as not only a victim and perpetrator of cyber-criminals but also as a defender, categorizes cybercrimes and explores their effects and measures. It also includes cyber security policy and management, risk and compliance, laws and regulations, accreditation, and courses related to the profession.

Wang et al. proposed a multilayer perceptron-based framework where a concept of handling the errors generated in the classification of DDoS-affected packets has been introduced [28]. The authors have considered parameters such as the IP of the source, destination port, and some other attributes of TCP packets for classification.

Jia et.al in their work have proposed an all-inclusive framework in [10]. A combination of recurrent LSTM (Long Short–Term Memory) neural network and CNN (Convolutional Neural Network) model is prepared and applied to the CIC-DDoS2019 dataset.

Latifet. al. have used an EVFDT ('Enhanced Very Fast Decision Tree') to identify DDoS assault in the WBAN ('Wireless Body Area Network') scenario [15]. The work generates highly accurate results, and the number of false alarms is also very low.

Saini et. al. [21] used three different techniques to detect DDoS attacks in an IoT scenario. The first technique used is J48, a decision tree model. 'Random Forest' and 'Naïve Base' classifiers were being used to segregate the malicious packets. It was found that the results obtained were very accurate.

Suresh et al. [27] used two different methods to identify the important characteristics of a packet. One is chi-square, and the other is information gain. The next step involved using models like Naïve Bayes, C4.5, and KNN to detect the DDoS attack. The last model outperformed all other models for the dataset under consideration.

Shieh et.al. tried to address the OSR('Open Set Recognition') issue [24]. The attackers use various techniques to launch a DDoS attack. When a detection system is designed based on some known methods and the attacks are devised differently, then the system may fail. At least the performance would not be the same as that of detecting the known method of attack. The authors in their work have used GMM ('Gaussian Mixture Model') and BI-LSTM ('Bidirectional Long-Short Term Memory') to design the detector models. It is found that GMM is very effective in identifying new categories of attacks.

Al-Hadhrami et. al. [1] have studied several scenarios leading to DDoS attacks in the IoT. Various mitigation techniques were considered, and the effectiveness of those was analysed. Some points regarding the open problems in these scenarios were also discussed.

Marvi et. al. [16] have suggested a three-step procedure to select the features before applying the data to the detection modules. The proposed framework was being designed using a decision tree-based LGBM ('Light Gradient Boosting Machine') algorithm. Source IP, Mean ACK, Header length, etc. are the parameters the model considers. Unseen DDoS attacks can also be identified using the model.

Shrivastava et. al. investigated the role of different Android devices in connection with IoT networks [25]. The authors mainly focused on the applications running on any Android device, interfacing with IoT. They used sensitivity analysis techniques to assess the effectiveness of Android intents as a distinguishing characteristic to identify malicious apps. Additionally, a substantial number of samples gathered from Android app markets are used in the proposed study. Several criteria are assessed and contrasted using the methods currently in use.

Authors in [18] surveyed several manuscripts on attacks such as DDoS and MitM in networks such as IoT, IoMT and Vanet. The works of various authors were being studied and methods suggested for identification and/or mitigation of these attacks were listed for better illustration.

Sharma et.al. [23] in their work investigated the risks faced by Android devices which are most of the time connected to IoT networks. It has been suggested that these devices are prone to various attacks due to the lack of legitimacy audits. They achieved excellent accuracy, using machine-learning models on the M0Droid dataset.

The above discussion is summarized in Table 2.1.

3. Objective. In a DDoS attack, the attacker tries to exhaust the recipient device by sending a huge quantity of packets. Different types of packets can be used for the above purpose. Hence, discovering the attack involves identifying the packets used in the attack. Our objective is to concentrate on three categories of packets. The abnormality in the received packets is going to be identified if any of these types of packets are being used in the transmitted data.

4. Workflow. Various related works have been analysed. It is decided that more than one type of classifier will be applied to the data set under consideration. The CICDDoS2019 dataset is used in this work, which is very close to the packets transmitted in the real network. The attributes of the packets under transmission are being considered. The 'Information Gain Ranking' [29] method arranges the attributes in order. 33 attributes are selected, leaving behind other insignificant attributes. After preprocessing the data, three different classification methodologies are applied to the said dataset. The results obtained using all three methods are compared at the end. The steps described above are represented in Figure 4.1.

Table 2.1: Literature survey.

Author Name	Description	Method	Parameter	Advantages	Future Scope
Jain et. al. (2021)	The authors considered the capabilities of different machine learning classifier techniques and then used avoting mechanism for the best.	Naïve Bayes Random Forest KNN SVM	Source IP Destination IP Packet Size Source Port Flow Bytes Header length etc.	Ensemble performs better than individual classifiers	Instead of the existing dataset, a simulated dataset with a higher size is to be considered.
Wang et.al. (2020)	The authors tried to optimize the detector using feedback.	Multi-Layer Perceptron	Source IP TCP attributes Destination port.	Handling of detection errors in detecting DDoS is novel.	To extend the work for SDN
Jia et.al. (2020)	A comprehensive framework for security against DDoS attacks in IoT was proposed	LSTM and CNN	Source IP Destination IP Packet Size Source port Destination Port	Comprehensive Approach	Implementing the proposed work in parallel systems.
Latif et. al. (2015)	The authors used a version of the decision tree in WBAN, a primitive step towards IoT.	Very Fast Decision Tree	Packet Loss Delay Jitter	High accuracy Low false alarm	Simulation -based approach to be implemented in real-world scenario.
Saini et.al. (2020)	Authors have analysed the performance of the different machine-learning techniques in identifying malicious packets.	J48 Random Forest Naïve base	SRC Address DES Address PKT ID PKT AVG SIZE	High accuracy	To work on more types of attacks.
Shieh et.al. (2021)	Authors in this work tried to address the Open Set Recognition issue in DDoS attack identification.	Gaussian Mixture Model BI-LSTM	MI_dir_L5_weight MI_dir_L5_mean MI_dir_L3_variance etc.	GMM is effective in both trained and novel attacks.	The proposed model can be validated on more datasets.
Marvi et al. (2020)	Authors suggested a three-step feature selection before applying the model. LGBM procedure is used for identifying DDoS attacks.	LGBM algorithm	Source IP Mean ACK Header length etc.	The proposed framework even works for unseen DDoS attacks of some specific types.	The work is to be applied to other types of DDoS attacks.

The whole process can be broadly divided into three phases. 'Preparation is the primary phase, then 'Pre-processing' is applied, followed by 'Classification' and 'Analysis' phases.

4.1. Preparation. In the preparation phase, the dataset is selected to work on. The CICDDOS2019 dataset is a well-defined dataset that represents the context and is very close to the actual scenario. Hence, that is considered for this work. Various categories of DDoS attacks are possible on a network. Each of these datasets contains numerous attributes. So, there is a need to select the vital ones. One of the ranking algorithms

Fig. 3.1: Objective.

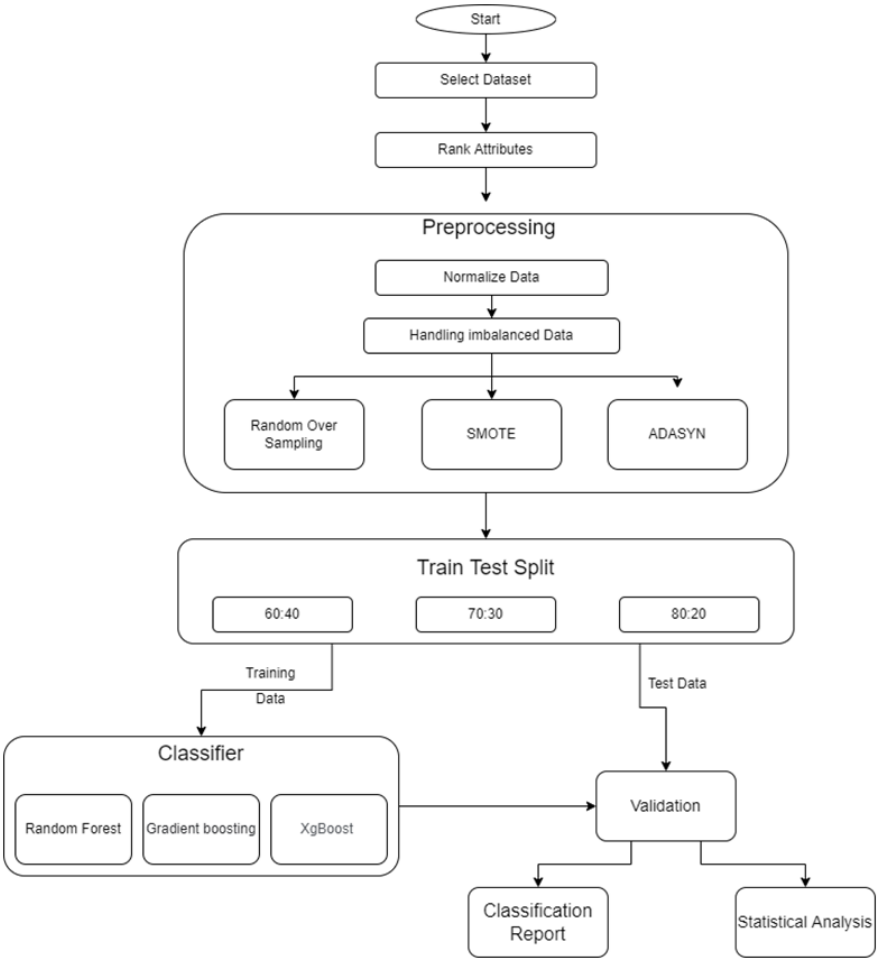

Fig. 4.1: Workflow.

called 'Information Gain Ranking' is being applied [29]. After finalizing the set of attributes, the final dataset is prepared and passed to the second stage.

4.2. Preprocessing . In this phase, the data are first normalized to make it appropriate for the machine learning model. The next step is to handle the imbalanced data set. As the process flow diagram demonstrates, three sampling techniques are introduced to the dataset. Random Oversampling [2] is a simple method to balance a data set. In this case, the minority class data are randomly replicated to bridge the gap.

SMOTE is an oversampling method [4]. Sampling by replacement is not being used in this technique. Rather, several examples are synthesized and used for oversampling. According to the amount of oversampling,

the necessary records are synthesized based on the k nearest neighbours.

'ADASYN' is an advanced approach for synthetic sampling [7]. The minority class data are considered to be either 'easier to learn' or 'difficult to learn'. The dataset will include more synthesized data if it falls under the first category. It helps to reduce the imbalance.

All the classification models are considered for each version of the re-sampled dataset.

4.3. Classification. After finalising the dataset, three different classification techniques are applied to it. Random Forest, Gradient Boosting, and Extreme Gradient Boosting classification models are applied to this work's dataset. The proposed model, i.e., using the XgBoost algorithm with ADASYN for classifying affected packets, outperformed the other models under investigation. All three classifiers used for classification in this work are described here.

4.3.1. Random Forest. Random Forest [8] is popularly used as a classification model. A random forest can be constructed by combining several decision trees. Here we use a voting concept to decide the class of the data under consideration. The result is based on the aggregation of the predictions.

The steps can be expressed as described in Algorithm 1.

Algorithm 1 Random Forest

Initialize $i = 1$
while $i \leq M$ **do**
  Make D_i = A sub dataset decided randomly
  Make Nd_i = A node with sub dataset D_i
  Invoke **Construct(**Nd_i**)**
end while
Construct(Nd_i**)**
if p and q belong to C and N respectively and $p = q$ **then**
  return
else
  Verify every possible splitting criteria
  Choose the feature F that has the highest information gain
  Create n sub-nodes of N {Here n represents the different possible features of F}
  Assign $c = 1$
  while $c \leq n$ **do**
    Assign $Nd + i = D_i$
    Invoke **Construct(**Nd_i**)**
    $c = c + 1$
  end while
end if

4.3.2. Gradient Boosting. 'Gradient Boosting' is an ensemble technique [17]. It is a combination of several weak learners (in this case, decision trees) to create a powerful classification model. This is a boosting ensemble technique in which several homogeneous weak learners work sequentially, improvising the model to get better results. It works by optimizing a loss function to minimize prediction errors. The basic steps are listed in Algorithm 2.

4.3.3. Extreme Gradient Boosting. Extreme Gradient Boosting (XgBoost) [5] comprises steps to optimize 'Gradient Boosting' training. 'Boosting' is different from the 'Bagging' concept. Bagging uses a voting mechanism to finalize the output, whereas boosting is an ensemble of phases. In the case of boosting, each phase learns from the previous phase. The 'XgBoost' uses the same concept, and the steps can be briefly explained as follows in Algoritm 3.

4.4. Analysis. For validating the investigation, different ratios of the train-to-learn data set are considered. The same model is applied to three different combinations of train-to-learn ratios, i.e., 60:40, 70:30, and 80:20.

After getting the result by applying the three different classifiers mentioned earlier, the result is represented using a confusion matrix. Then the different parameters, including accuracy, precision, recall, and f1-value,

Algorithm 2 Gradient Boosting

Input:

- Training set $\{(x_i, y_i)\}$
- Loss function L
- Number of iterations: M
- Base learner model: $h(x)$

A. Initialize the model:

$$f_0(x) = \arg\min_{\gamma} \sum L(y_i, \gamma)$$

B. Perform iterations to learn:

for $m = 1$ to M **do**

a. Calculate pseudo-residuals:

$$r_{i,m} = \left(\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right)_{f=f_{m-1}} \quad \text{for } i = 1 \text{ to } n$$

b. Train a weak learner:

Fit the base learner to $h_m(x)$

c. Calculate multiplier γ_m using the given optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum L(y_i, f_{m-1}(x_i) + \gamma h_m(x_i))$$

d. Update the model:

$$f_m(x) = f_{m-1}(x) + \gamma_m h_m(x)$$

end for

C. Output $f_m(x)$

are calculated. These measures are then compared with some related work. Also, the results obtained are statistically analysed, and comparing different models' performances is established.

5. Experimentation & Result Analysis. The CICDDoS2019 dataset is being considered in this work for classifying malicious and benign packets. The three methodologies explained in the earlier section are used. The dataset has 88 attributes for each of the entries. Using the ranking process explained earlier, 33 attributes have been considered in this work. In that process, the attributes selected are based on the entropy value. The decrease in entropy of each attribute is considered to evaluate the gain with respect to the target, which in turn helps in ranking the attributes. The method associates each of the attributes with a weight. For this work, we fixed a standard weight value and considered the 33 attributes with a weight value greater than the threshold. The attributes considered are: *Flow ID, Destination IP, Source IP, Avg Fwd Segment Size, Fwd Packet Length Mean, Average Packet Size, Sub flow Fwd Bytes, Total Length of Fwd Packets, Packet Length Mean, Destination Port, Flow Bytes/s, Bwd Packets/s, Bwd Header Length, Sub flow Bwd Packets, Packet Length Variance, Bwd IAT Mean, Flow IAT Mean, Flow Packets/s, Flow Duration, Init Win bytes forward, Total Length of Bwd Packets, Sub flow Bwd Bytes, Fwd Packets/s, Avg Bwd Segment Size, Bwd Packet Length Mean, Protocol* etc.

Before the implementation of any of the classification techniques, the dataset is first normalised. The dataset is normalised to make the classifiers work better than non-normalized data. As in the non-normalized data, the attributes tend to have values on different scales. Hence, the statistical technique of normalization is used to convert the values to a particular range.

Then the dataset is made to undergo another statistical method called sampling. In this work, the over-sampling method is chosen as the minor class has a relatively lower count than the major class data. The classification techniques are applied after the sampling. 'Random oversampling', 'SMOTE' and 'ADASYN' are the three sampling methods applied to the dataset.

Algorithm 3 Extreme Gradient Boosting Algorithm

Input:

- Training set $\{(x_i, y_i)\}$
- Loss function L
- Number of learners with count M
- Learning rate denoted as α

A. Initializing the model:

$$f_0 = \arg\min_{\theta} \sum L(y_i, \theta)$$

B. Loop to update the model:

for $q = 1$ to M **do**

a. Evaluate gradients:

$$g_q(x_i) = \left(\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right)$$

b. Evaluate Hessians:

$$h_q(x_i) = \left(\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2} \right)$$

c. Use the following training set to fit a base learner:

$$\left\{ x_{i,q} - \left(\frac{g_q(x_i)}{h_q(x_i)} \right) \right\} \text{ by resolving the following problem}$$

$$\phi_q = \arg\min_{\phi} \sum \frac{1}{2} h_q(x_i) \left(\left(-\frac{g_q(x_i)}{h_q(x_i)} \right) - \phi(x_i) \right)^2$$

$f_q(x) = \alpha \phi_q(x)$

d. Update the model:

$$f_q(x) = f_{q-1}(x) + f_q(x)$$

end for

C. Output:

$$f(x) = \sum f_q(x)$$

After the dataset is sampled, three different classification techniques are applied. Those are Random Forest, Gradient Boosting, and XgBoosting. Each version of sampling is combined with each classification technique under consideration in this work. Hence, nine different models are considered; for example, SMOTE is combined with Random Forest, Gradient Boosting, and XgBoosting to form three different classification models. After applying each of the methods, the quality of classification is judged by the measures widely used, and those are explained further.

First, the numbers of 'True Malignant', 'False Malignant', 'True Benign', and 'False Benign' instances are recorded for each type of packet.

Based on these values, 'precision', 'accuracy', 'recall' and 'F1 Value ' are calculated to verify results obtained by applying the methods. The concepts behind these measures are discussed next.

Accuracy represents the percentage of the total number of packets identified as compared to the total number of samples under consideration. It can be expressed as equation quation 5.1.

$$.\text{Accuracy} = \frac{TM + TB}{TM + FM + FB + TB} \tag{5.1}$$

Another measure, 'precision' is used to represent the total number of truly identified positive cases against the total number of predicted positive cases. It may be described as equation 5.2.

$$\text{Precision} = \frac{TM}{TM + FM} \tag{5.2}$$

Table 5.1: Result Analysis of Random Forest with different cases of Sampling

MODEL	SPILT RATIO	TM	FM	FB	TB	ACCU RACY	PRECI SION	RECALL	F1	TOTAL
RF- NO SAMPLING	60:40	6490	1107	1756	2405	0.76	0.85	0.79	0.82	11758
	70:30	4657	840	1336	1985	0.75	0.85	0.78	0.81	8818
	80:20	6283	382	916	1298	0.85	0.94	0.87	0.91	8879
RF-ROS	60:40	6108	1718	1451	5994	0.79	0.78	0.81	0.79	15271
	70:30	4848	802	1145	4657	0.83	0.86	0.81	0.83	11452
	80:20	6245	725	864	2901	0.85	0.90	0.88	0.89	10735
RF-SMOTE	60:40	6199	1727	1374	6070	0.80	0.78	0.82	0.80	15370
	70:30	4734	1069	1069	4314	0.81	0.82	0.82	0.82	11186
	80:20	6130	687	785	3130	0.86	0.90	0.89	0.89	10732
RF-ADASYN	60:40	6994	1435	1794	6257	0.80	0.83	0.80	0.81	16480
	70:30	6619	968	802	4696	0.86	0.87	0.89	0.88	13085
	80:20	6254	687	764	2939	0.86	0.90	0.89	0.90	10644

Table 5.2: Result Analysis of Gradient Boosting with different Cases of Sampling

MODEL	SPILT RATIO	TM	FM	FB	TB	ACCU RACY	PRECI SION	RECALL	F1	TOTAL
GB- NO SAMPLING	60:40	6970	727	947	3214	0.86	0.91	0.88	0.89	11858
	70:30	5848	649	551	2771	0.88	0.90	0.91	0.91	9819
	80:20	6582	649	802	2642	0.86	0.91	0.89	0.90	10675
GB-ROS	60:40	6650	1176	107	7337	0.92	0.85	0.98	0.91	15270
	70:30	5004	532	360	5557	0.92	0.90	0.93	0.92	11453
	80:20	8863	764	507	2901	0.90	0.92	0.95	0.93	13035
GB-SMOTE	60:40	6994	832	336	7108	0.92	0.89	0.95	0.92	15270
	70:30	5199	337	465	5452	0.93	0.94	0.92	0.93	11453
	80:20	8901	534	725	3474	0.91	0.94	0.92	0.93	13634
GB-ADASYN	60:40	7032	756	260	6841	0.93	0.90	0.96	0.93	14889
	70:30	5710	283	360	4832	0.94	0.95	0.94	0.95	11185
	80:20	8939	678	514	3901	0.92	0.93	0.95	0.94	14032

The third measure used for the analysis of classification is known as 'recall'. It is the total number of positive cases identified as true against the total positive samples. It is expressed as equation 5.3.

$$\text{Recall} = \frac{TM}{TM + FB} \tag{5.3}$$

Another measure, the 'F1 Score', was used to check the balance between the previous two measures. It may be measured as equation 5.4.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

The readings and the measures obtained in this work are mentioned in Tables 5.1, 5.2 and 5.3.

The classification techniques are applied to the non-sampled data along with the sampled data. The findings are represented in the tables above. The random forest classifier performs most accurately when combined with ADASYN and with the split ratios 70:30 and 80:20. But the precision is highest when no sampling is applied with a split ratio of 80:20. The recall is highest when RF-SMOTE is used with 80:20 ratios, as well as for the RF-ADASYN combination with 70:30 and 80:20 split ratios. In the case of Random Forest, the maximum accuracy gained is 86%.

Table 5.3: Result Analysis of Extreme Gradient Boosting with Different Cases of Sampling

MODEL	SPILT RATIO	TM	FM	FB	TB	ACCURACY	PRECISION	RECALL	F1	TOTAL
XG-NO SAMPLING	60:40	8490	707	565	2596	0.90	0.92	0.94	0.93	12358
	70:30	8102	487	898	2736	0.89	0.94	0.90	0.92	12223
	80:20	8092	687	764	2336	0.88	0.92	0.91	0.92	11879
XG-ROS	60:40	8108	680	683	6299	0.91	0.92	0.92	0.92	15770
	70:30	8443	764	525	3901	0.91	0.92	0.94	0.93	13633
	80:20	8345	764	525	3901	0.90	0.92	0.94	0.93	13535
XG-SMOTE	60:40	8681	345	565	5879	0.94	0.96	0.94	0.95	15470
	70:30	8489	640	573	3825	0.91	0.93	0.94	0.93	13527
	80:20	8398	640	573	3825	0.91	0.93	0.94	0.93	13436
XG-ADASYN	60:40	8261	527	422	5879	0.94	0.94	0.95	0.95	15089
	70:30	9245	424	278	3749	0.95	0.96	0.97	0.96	13696
	80:20	9245	524	278	3749	0.94	0.95	0.97	0.96	13796

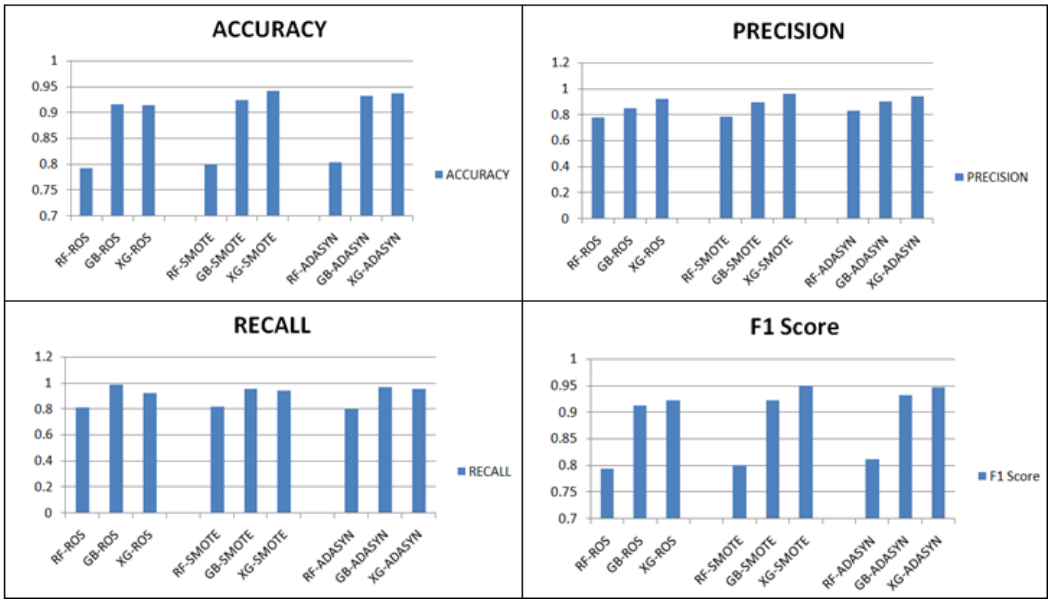

Fig. 5.1: Overall analysis with 60:40 split

The next classifier under consideration is the gradient-boosting method. It provides the highest accuracy of 94% when combined with the ADASYN sampler, and the train test split is 70:30. It also gives the most precise result for the same combination. Though recall is highest for the Gradient Boosting-Random Oversampling combination with a 60:40 ratio, this combination doesn't have better accuracy or precision. From the observation, it may be concluded that the Gradient Boosting-ADASYN model with a 70:30 split of data is the best combination with a sufficient value of recall and f1 score.

The last technique used for classification is 'Extreme Gradient Boosting'. When this is combined with the ADASYN sampler with 70:30 split ratios, the best accuracy of 95% is obtained. The same combination also has the best precision of 96% with better recall and f1 score, i.e., 97% and 96%.

The analysis of three different cases (train-test data ratio-wise, i.e., 60:40, 70:30, and 80:20) is further represented using Figures 5.1, 5.2, and 5.3.

The analysis shows that the XgBoost–ADASYN model results in the highest accuracy and F1 score. Also,

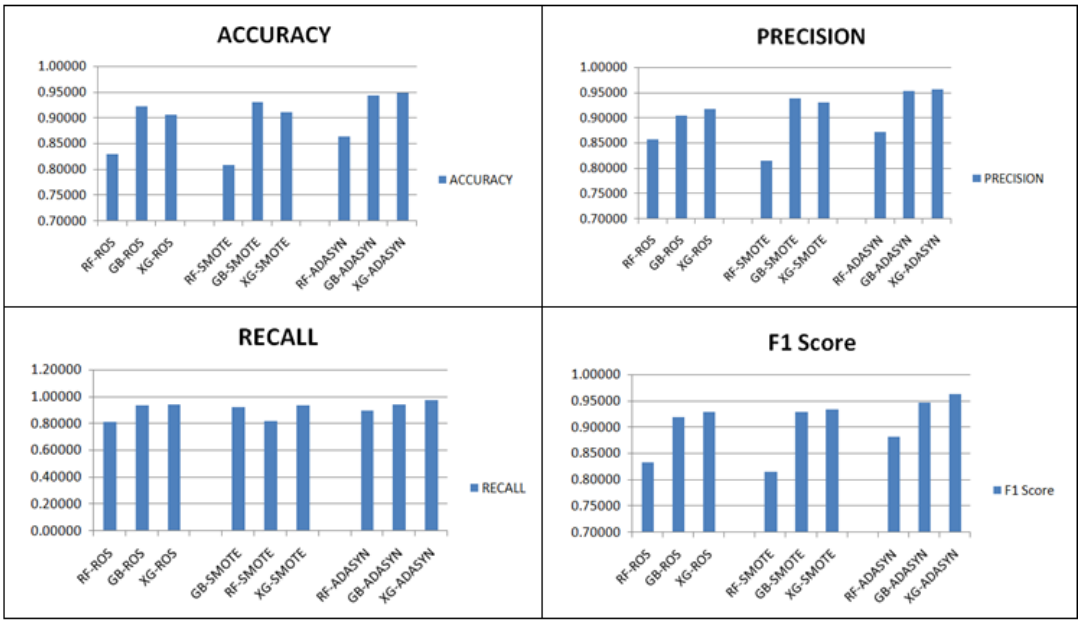

Fig. 5.2: Overall analysis with 70:30 split

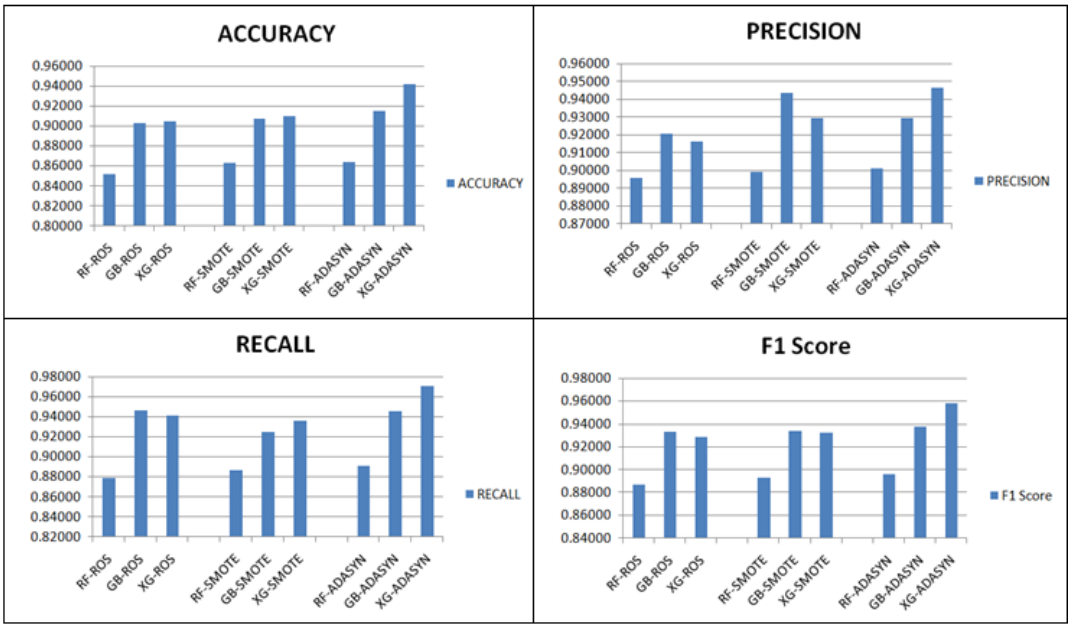

Fig. 5.3: Overall analysis with 80:20 split

this model yields the second-best precision and recall metrics. The same classifier, i.e., XgBoost gives the best precision when used with the SMOTE technique.

Statistical Analysis. It is evident from the plot in Figure 5.4 that the median improvement in GB-SMOTE accuracy is close to 0.915, and the median rise in XG-SMOTE accuracy is near 0.93. These results

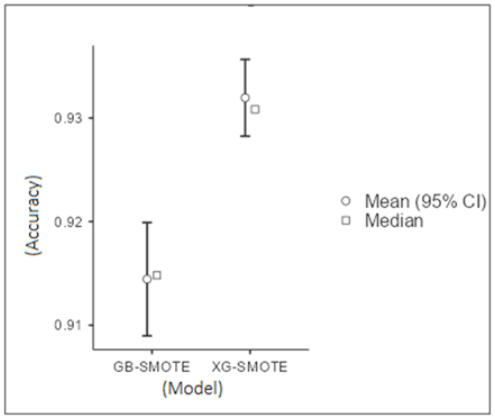

Fig. 5.4: Model vs accuracy plot

Table 5.4: Wilcoxon's Paired Samples T-Test

Model 1	Model 2	Wilcoxon W	P value	Mean difference
GB-SMOTE-60-40	GB-ADASYN-60-40	8.00	0.049	-0.0105
XG-ROS-60-40	XG-SMOTE-60-40	0.00	0.002	-0.0253
GB-SMOTE-60-40	XG-SMOTE-60-40	0.00	0.002	-0.0172
GB-ADASYN-60-40	XG-ADASYN-60-40	8.00	0.049	-0.00392
GB-ROS-60-40	XG-ROS-60-40	55.0	0.002	0.00935
GB-SMOTE-70-30	GB-ADASYN-70-30	1.00	0.004	-0.0125
XG-ROS-70-30	XG-SMOTE-70-30	27.00	1.00	4.23e-4
GB-SMOTE-70-30	XG-SMOTE-70-30	54.0	0.004	0.0224
GB-ADASYN-70-30	XG-ADASYN-70-30	17.0	0.322	-0.00401
GB-ROS-70-30	XG-ROS-70-30	55.0	0.006	0.0202
GB-SMOTE-80-20	GB-ADASYN-80-20	42.0	0.160	0.00596
XG-ROS-80-20	XG-SMOTE-80-20	24.0	0.770	0.00263
GB-SMOTE-80-20	XG-SMOTE-80-20	40.0	0.232	0.00812
GB-ADASYN-80-20	XG-ADASYN-80-20	1.00	0.004	-0.0283
GB-ROS-80-20	XG-ROS-80-20	9.00	0.064	-0.00689

suggest that XG-SMOTE outperforms GB-SMOTE in terms of effectiveness. We still need to determine whether our discovery is statistically significant. For a paired sample t-test, the Wilcoxon rank test [20] [19] is performed with the null hypothesis H_0 set to "there is no difference in accuracy among the two models". The total of the signed ranks, as given by equation 5.5, is the test statistic W in the Wilcoxon signed-rank test.

$$W = \sum_{i=1}^{N} [\text{sgn}(x_{2,i} - x_{1,i}) \cdot R_i] \tag{5.5}$$

The i^{th} value of N measurement pairs is indicated by $x_i = (x_{1,i}, x_{2,i})$, while the pair's rank is shown by R_i.

The test statistic W and P values for the paired sample T-test are shown in Table 5.4. At the 5% significance level, the null hypothesis can be rejected if the P value is less than 0.05. It is therefore inclined to believe the alternative hypothesis, according to which model 2 performs better than model 1 or vice versa.

From the Table 5.4, it is observed that the GB-ADASYN model outperforms the GB-SMOTE and the XG-ADASYN model is better than the first one. So it may be concluded that the XG-ADASYN model works better than other models under consideration.

Table 5.5: Comparison with existing work

Author	Percentage with Method & Parameter	
Kumar et. al.	88.8% (RF)(Accuracy)	77.78% (Naïve Bayes) (Accuracy)
Sharafaldin et. al.	77%(RF) (Precision)	78% (ID3) (Precision)
Shieh et. al.	89.80%(BI-LSTM) (Accuracy)	86.8%(BI-LSTM-GMM) (Accuracy)
Our Work	94.8% (Accuracy)	96.1% (Precision)

Comparison. From the experiment, it was found that XgBoost, when used with the ADASYN sampling method, outperforms other models under consideration with respect to accuracy and F1 score, where as the XgBoost-SMOTE method has better precision over other models. Further, the results obtained are compared with some of the existing work, such as [24], [14] and [22]. The comparison is demonstrated in Table 5.5.

6. Conclusion and Future Work. The behaviour of the DDoS dataset is extensively studied, and then a few state-of-the-art classification techniques are investigated. It may be suggested that not only the classification technique but also the sampling method play a great role in obtaining better results. From the overall observation, the conclusion obtained is that the XgBoost-ADASYN combination helps in obtaining the best results in this scenario.

Future research will consider other datasets related to various protocol packets. Some other techniques for identifying malicious packets may also be considered for identifying proper attributes, and then normalizing and sampling, as well as new techniques for classification, may be considered to improve efficiency.

REFERENCES

[1] Y. Al-Hadhrami and F. K. Hussain, *Ddos attacks in iot networks: a comprehensive systematic literature review*, World Wide Web, 24 (2021), pp. 971–1001.
[2] G. E. Batista, R. C. Prati, and M. C. Monard, *A study of the behavior of several methods for balancing machine learning training data*, ACM SIGKDD explorations newsletter, 6 (2004), pp. 20–29.
[3] E. Bursztein, *Inside the infamous mirai iot botnet: A retrospective analysis*, Cloudflare Blog, Aug, (2020).
[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *Smote: synthetic minority over-sampling technique*, Journal of artificial intelligence research, 16 (2002), pp. 321–357.
[5] T. Chen and C. Guestrin, *Xgboost: A scalable tree boosting system*, in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[6] D. Giusto, *A. lera, g*, Morabito, l. Atzori (Eds.) The Internet of Things, (2010), p. 11.
[7] H. He, Y. Bai, E. A. Garcia, and S. Li, *Adasyn: Adaptive synthetic sampling approach for imbalanced learning*, in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), Ieee, 2008, pp. 1322–1328.
[8] T. K. Ho, *Random decision forests*, in Proceedings of 3rd international conference on document analysis and recognition, vol. 1, IEEE, 1995, pp. 278–282.
[9] A. K. Jain et al., *Ddos detection using machine learning ensemble*, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12 (2021), pp. 1647–1655.
[10] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, *Flowguard: An intelligent edge defense mechanism against iot ddos attacks*, IEEE Internet of Things Journal, 7 (2020), pp. 9552–9562.
[11] G. Kesavaraj and S. Sukumaran, *A study on classification techniques in data mining*, in 2013 fourth international conference on computing, communications and networking technologies (ICCCNT), IEEE, 2013, pp. 1–7.
[12] M. Khari, G. Shrivastava, S. Gupta, and R. Gupta, *Role of cyber security in today's scenario*, in Detecting and mitigating robotic cyber security risks, IGI Global, 2017, pp. 177–191.
[13] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, *Ddos in the iot: Mirai and other botnets*, Computer, 50 (2017), pp. 80–84.
[14] A. Kumar and T. J. Lim, *Edima: Early detection of iot malware network activity using machine learning techniques*, in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), IEEE, 2019, pp. 289–294.
[15] R. Latif, H. Abbas, S. Latif, A. Masood, et al., *Evfdt: an enhanced very fast decision tree algorithm for detecting distributed denial of service attack in cloud-assisted wireless body area network*, Mobile Information Systems, 2015 (2015).
[16] M. Marvi, A. Arfeen, and R. Uddin, *A generalized machine learning-based model for the detection of ddos attacks*, International Journal of Network Management, 31 (2021), p. e2152.
[17] A. Natekin and A. Knoll, *Gradient boosting machines, a tutorial*, Frontiers in neurorobotics, 7 (2013), p. 21.

[18] D. Panda, B. K. Mishra, and K. Sharma, *A taxonomy on man-in-the-middle attack in iot network*, in 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), IEEE, 2022, pp. 1907–1912.
[19] R. R Core Team et al., *R: A language and environment for statistical computing*, 2013.
[20] M. Şahin and E. Aybek, *Jamovi: an easy to use statistical software for the social scientists*, International Journal of Assessment Tools in Education, 6 (2019), pp. 670–692.
[21] P. S. Saini, S. Behal, and S. Bhatia, *Detection of ddos attacks using machine learning algorithms*, in 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, 2020, pp. 16–21.
[22] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, *Developing realistic distributed denial of service (ddos) attack dataset and taxonomy*, in 2019 international carnahan conference on security technology (ICCST), IEEE, 2019, pp. 1–8.
[23] K. Sharma and B. B. Gupta, *Towards privacy risk analysis in android applications using machine learning approaches*, International Journal of E-Services and Mobile Applications (IJESMA), 11 (2019), pp. 1–21.
[24] C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, *Detection of unknown ddos attacks with deep learning and gaussian mixture model*, Applied Sciences, 11 (2021), p. 5213.
[25] G. Shrivastava and P. Kumar, *Sensdroid: analysis for malicious activity risk of android application*, Multimedia Tools and Applications, 78 (2019), pp. 35713–35731.
[26] S. G. H. Soumyalatha, *Study of iot: understanding iot architecture, applications, issues and challenges*, in 1st International Conference on Innovations in Computing & Net-working (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications, vol. 478, 2016.
[27] M. Suresh and R. Anitha, *Evaluating machine learning algorithms for detecting ddos attacks*, in Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15-17, 2011 4, Springer, 2011, pp. 441–452.
[28] M. Wang, Y. Lu, and J. Qin, *A dynamic mlp-based ddos attack detection method using feature selection and feedback*, Computers & Security, 88 (2020), p. 101645.
[29] E. Zdravevski, P. Lameski, A. Kulakov, B. Jakimovski, S. Filiposka, and D. Trajanov, *Feature ranking based on information gain for large classification problems with mapreduce*, in 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 2, IEEE, 2015, pp. 186–191.

Edited by: Manish Gupta
Special issue on: Recent Advancements in Machine Intelligence and Smart Systems
Received: May 29, 2024
Accepted: Aug 4, 2024