A Machine Learning Classification Model for Detecting DDoS Attacks using Decision Tree Gini Entropy and Random Forest

Thai Son Chu
School of Computer, Data & Mathematical Sciences
Western Sydney University
Sydney, NSW Australia
15845085@student.westernsydney.edu.au

Weisheng Si
School of Computer, Data & Mathematical Sciences
Western Sydney University
Sydney, NSW Australia
w.si@westernsydney.edu.au

Simeon Simoff
School of Computer, Data & Mathematical Sciences
Western Sydney University
Sydney, NSW Australia
s.simoff@westernsydney.edu.au

Quang Vinh Nguyen
School of Computer, Data & Mathematical Sciences and MARCS Institute
Western Sydney University
Sydney, NSW Australia
q.nguyen@westernsydney.edu.au

ABSTRACT

Distributed Denial of Service (DDoS) attacks are menaces to network security that causes exhausting the target networks with malicious traffic. With simple but powerful attack mechanisms, it introduces an immense threat to the current Internet community. The infected networks are controlled remotely by hackers or Trojans, which are programmed to launch packet floods. We propose a novel model based on an entropy random forest algorithm, combining with decision tree Gini index and significantly improving DDoS attack detection. In comparison with existing benchmarking models, the experimental results indicate that our machine learning model can detect attacks and provide great confidence to secure networks that may be disrupted in the future. In this study, we use the newly released dataset CICDDoS2019, which contains a comprehensive variety of DDoS attacks, including a new classification based on the flows network.

CCS CONCEPTS

• DDoS • Random Forest • Gini • Entropy.

1 Introduction

Distributed Denial of Service (DDoS) attacks target websites and online services. The purpose of this attack is to jam the network or server with overwhelming traffic. It achieves its goal by utilizing many compromised systems as sources to attack a network. There are different sub-types of DDoS attacks based on the layer of the network connection they attempt to attack concerning the Open Systems Interconnection (OSI) model. Some sub-types of DDoS attacks that we classified through our research are User Diagram Protocol Flood (UDP) [1], Synchronize Flood (SYN) [2] [3], Internet Control Message Protocol (ICMP) [1], Hypertext Transfer Protocol (HTTP) [4], and Slow Post (POST) [5] [1].

These days, with the backing of machine learning, applications become more accurate at decision-making and predicting outcomes. We are utilizing this method to solve problems in various domains with accuracy close to human performance in detecting threats in the field of cybersecurity.

The main motivation of DDoS attacks is to severely slow or shut down a specific resource and one way of operation is by exploiting a system flaw and causing a processing failure or exhaustion of system resources. Another way of attacking the victim system is by flooding [6] and monopolizing the network, prohibiting anyone else from using it [7] [8] [9]. While the absence of datasets was the very focal point at which this study was conducted, it can also be seen as a limitation on its own given the fact that potentially results that are more accurate would have been obtained on the comparison between the datasets [10] [11].

Considering numerous threats to the vulnerable networks [12] that have categorized into two types of threads, including insider attacks and outsider attacks. An integrated and cooperative NIDS framework is deployed at the front end on the controller and back end on every processing server to identify both classifications of intrusions. All the NIDS placed on the servers work cooperatively to update their signature database by receiving alerts stored in the central log. This makes it possible for correlation in the central registry, and hence, detection of unknown attacks is possible. The cognitive module in this design uses Snort to classify an attack by detecting intrusions based on the misuse detection database. Snort tries to determine the nature of the attack and transmits the information to the Alert System, and the packet will be refused. This technique allows the researchers to easily update the misuse database without any alteration of the existing rules.

A protocol-based network intrusion detection system is designed [13] [14] to detect DDoS attacks in networks. In this system, Incoming packets are distributed according to the protocol and queued for additional processing. Relevant features will be extracted, and protocol-specific classifiers are applied to each packet to generate alerts and thus update the attack signature database.

Singh et al. [15] have designed a framework using Snort as a rule-based attack detection system and have installed NIDS in the virtual bridge to monitor network traffic and to form low-level intrusion alerts. The correlation section in this design converts these low-level intrusion alerts to high-level intrusions. They selected a best attribute based on maximum information gain and make the root node of the tree to use this attribute. The branches of this node are the distinct values of the selected attributes. All profiles belong to a same class label the leaf node with that class label is created, if not, another attribute of categorical data values is selected to create an internal node like root node. There are no attribute with categorical data values remaining or the information gain of best attribute chosen is less than the threshold a model is created using SVM for the continuous values which is a popular classifier for data in high-dimensional space, i.e., data with large number of features [16].

Tang et al. [17] have also proposed a framework to detect DDoS attacks using Snort, a signature-based tool. The overall architecture adapts a module called the correlation unit. The component deployed over the network is a correlation unit so that all hosts can share the signatures in real-time. Snort itself has a detection engine that can match the packet with rules for any correlation. The Snort signature will be generated only when the major alert factor reaches a pre-set threshold. Kumar et al. [18] also described a signature-based DDoS using Snort. They stated intrusion signature system generates the rules used in the detection engine, including rule headers and rule options. Cisco has more than 2500 rule bases in the database. Further, users also can change the rules up to what they need. However, the model is not effective for the spoofed IP from valid address range. It is challenging to obtain range of expected IP for complex topology and the tunneling required for mobile IP users.

ROC graphs are a very useful tool for visualizing and evaluating classifiers [19]. They are able to provide a richer measure of classification performance than scalar measures such as accuracy, error rate or error cost. Because they decouple classifier performance from class skew and error costs, they have advantages over other evaluation measures such as precision-recall graphs and lift curves. However, as with any evaluation metric, using them wisely requires knowing their characteristics and limitations.

DDoS attack is a kind of distributed, cooperative large-scale attack. It has the same working principles as DoS, but compared with the traditional DoS whose attack is originated from a single attacker point, the realization of DDoS comes from hundreds or even thousands of PC attackers which have been installed Daemon, and it is a group-based attack behavior. Li et al. [8] proposed DDoS detection methods based on entropy computing. They stated that the DDoS detection algorithms based on entropy monitoring, and proposed two improved entropy detection approaches, cumulative entropy and time-based methods [8]. We could see that our cumulative entropy detection method has good detection capability.

The literature review has addressed research the gap in previous works that even though researchers strived to detect DDoS attacks with machine learning methods, there was a shortage of strategic level framework to apply such methods in a systematic manner and the comprehensive evaluation possible, as well as slow detection speed, less detection accuracy rate and slow convergence speed.

This paper presents the state-of-art architecture design and implementation of a DDoS detection system based on the hybrid of machine learning decision tree and random forest to cover the limitation from other research. The model's performance in binary classification is studied in terms of accuracy, precision, valid positive rate and f1_score [11]. In this research, we also thoroughly analyzed the logs generated during a DDOS attack, used supervised and unsupervised techniques for detection of threat, and finally used machine learning random forest to achieve over excellent accuracy rate for classifying different types of DDoS threats along with the safe connection. The experimental results on CICDDOS2019 dataset confirms the capability of the proposed model in detecting five types of attacks reaching to 97% of accuracy rate which contributes to the network security field in near future.

2 Proposed Model

The proposed model aims to detect the DDoS attack in data exchanged by vulnerable network servers at the transport layer of the vulnerable network. To do so, the model requires a process design, which would accept data of network traffic as input, then process the input data and generate a two-fold classification: “Attack” or “Benign”.

Figure 1 displays our model which has three main components involved: (1) Preprocessing, (2) Training and (3) Detection and Exploration is to give the output of whether it is an attack or benign. In the first component (preprocessing) of the proposed model, we preprocessed the data using the python language [20] for cleaning data, feature selection and normalization of the data. In the second component (training), we trained and validated the data before applying machine random forest machine learning for training and testing of the data. In the third component of our proposed model, we tested data and applied decision making to find out the attacks then visualized them in three-dimensional diagrams.

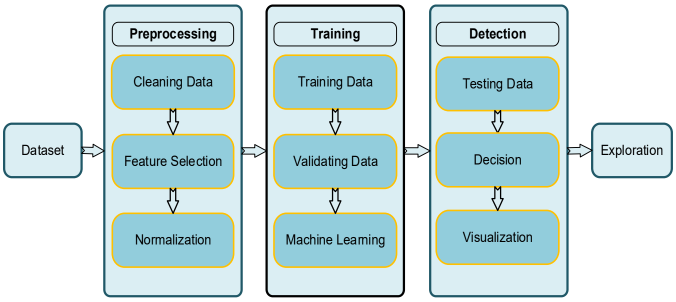


Figure 1: Our DDoS Machine Learning Detection Model

2.1 Preprocessing

Data preprocessing is the most efficient technique within data machine learning, making the raw data easy, clean, and understandable. Real-time data is inconsistent and incomplete but preprocessing is the most helpful technique through which less valuable data can be converted into valuable data.

- Data Cleaning: The first step in preparing data is removing any data that is not usable, such as empty data, and special characters [21]. The original data contains a large amount of missing and noise values [10]. We removed all these values from the data.

- Feature Selection: This is an essential step in the pattern recognition process and consists of defining the most miniature possible set of variables capable of efficiently describing a set of classes [22]. Several techniques for variable selection are available in the literature and implemented in software libraries like Scikit-Learn. In this work, special features parameters from the collected network traffic and assign each feature to the first column [16], and this will be used as a vector in the new dataset.
- Normalization: Required when researchers apply machine learning techniques to data that have different scales on attributes. The feature data have different numerical values. Training the model directly with the original data can cause classification error, then the model takes much time during its training [10]. We normalize the data where the minimum value is zero, and the maximum is one.

2.2 Training

- Training Data: The type of data builds up the machine learning algorithm [23]. We feed the data into the algorithm which corresponds to an expected output. The proposed model evaluates the data repeatedly to learn about the data's behavior and then adjusts itself to serve its intended purpose. Select features parameters from the collected network traffic and assign each feature to the first column, and this will be used as a vector in the new dataset.
- Validating Data: During training, validation data infuses new data into the model that it has not evaluated before. Validation data provides the first test against unseen data, allowing data scientists to evaluate how well the model makes predictions based on the new data. Not all data scientists use validating data, but it can provide some helpful information to optimize hyper-parameters, which influence how the model assesses data.
- Machine Learning: The datasets prepared for machine learning experiments in the previous section are used to derive various machine learning models and subsequent performance evaluations [16]. Our research idea is having a combination of learning models to increase the result in overall, simply using random forest to build multiple decision trees then merge them together to get more accuracy and reliable prediction.

2.3 Detection

- Testing Data: After the model is built, testing data once again validates that it can make accurate predictions. If training and validation data include labels to monitor performance metrics of the model, the testing data should be unlabeled. Test data provides a final, real-world check of an unseen dataset to confirm that the machine learning algorithm was trained effectively [16].
- Visualization: The visualization of the classifier performance [3]. Visualization techniques have advantage of the visual system's ability to process large amounts of information in order to efficiently represent network characteristics. Visualization in 3D allows n^2 more information to be visualized than its 2D counterparts, reduces clutter, results in clearer representations, and provides a more precise and accurate global view of the data's structure [23].

3 Case Study

3.1 Dataset

In this research, the dataset employed was CICDDOS2019 from the Canadian Institute for Cybersecurity that is the most current and comprehensive dataset available. Along with good-natured and the latest DDoS attacks, which are similar to real data (PCAP), CICDDoS2019 include the result of network traffic analyses [1]. Network traffic analyses use CICFlowMeter-V3 that has labelled traffic justified to attack (CSV documents), time stamping, originator and target IPs, originator and target ports, protocols. There are some different types of DDoS attacks, such as UDP, SYN, ICMP, HTTP and POST. The dataset are publicly available at https://www.unb.ca/cic/datasets/ddos-2019.html [13].

We used five different attacks csv files of UDP, SYN, ICMP, HTTP and POST, then concatenated into one single variable dataset, which was loaded into our python program then preprocessing techniques were applied to select the data for training and testing phases. We have 25 features including the class label as the most significant features according to the configured value of correlation coefficient [16].

3.2 Data preprocessing

As an initial experiment, reduced dataset samples are randomly selected from the whole training set and placed in a new .csv Microsoft Excel file. We used machine learning libraries for the numeric operations on the data manipulation and importing from the directory.

After the dataset has loaded, it must go through the data preparation process. Firstly, data cleaning process removes data that are not usable, such as open data and special characters. Table 1 shows the total number of records for all types of attacks after the data preparation process.

Table 1: Attack types in test-set along with number of samples used for classification

#	Attack Type	Number of Records
1	UDP	58,683
2	SYN	53,254
3	ICMP	56,427
4	HTTP	51,527
5	POST	51,286

Next, we combined all the attacks into single variable data and utilized the machine learning algorithms for the training and testing in detection of five types of attacks including UPD, SYN, ICMP, HTTP and POST.

For features selection in this research, we used 25 best features used to predict DDoS attacks [13], which are extracted from both malicious and normal traffic. These features consist of network flow information, including their statistical properties as well. The final dataset is in CSV format for evaluation.

Secondly, feature selection is used to select the relevant feature for DDoS attack detection. It can be chosen by using our algorithm or reviewing literature from proper research to minimize irrelevant attack detection features. Ultimately, feature engineering transforms the data into a form that deep learning can use in the training process. Each instance represents a snapshot of the network traffic at a given point in time. These instances are labelled according to the type of attacks. Classification is binary, where regular traffic is labelled UDP, SYN, ICMP, HTTP or POST.

Table 2: The feature set used in the proposed model

#	Feature	Description
1	Fwd Packet Length Max	Maximum packet size in the forward direction
2	Fwd Packet Length Min	Minimum packet size in the forward direction
3	Min Packet Length	Minimum length of a packet
4	Max Packet Length	Maximum length of a packet
5	Average Packet Size	Average size of a packet
6	Label	Attack type name
7	Protocol	TCP or UDP for data transmission
8	Fwd Packets	Number of forward packets per second
9	Fwd Header Length	Header length of a forwarded packet
10	Fwd Header Length 1	No. of bytes in a header in the forward direction
11	Min Seg Size Forward	Minimum segment size in the forward direction
12	Total Length of Fwd Packet	Packet size in the forward direction
13	Fwd Packet Length Std	Stand. deviation of a packet in the forward direction
14	Flow IAT Min	Min time between two packets in the flow
15	Subflow Fwd Bytes	Avg no. of bytes in a subflow in the fwd direction
16	Destination Port	Address to receive TCP or UDP packets
17	Packet Length Std	Standard deviation of the packet length
18	Flow Duration	Duration of the flow in µs
19	Fwd IAT Total	Total time between 2 packets in the fwd direction
20	ACK Flag Count	Count Number of packets with ACK
21	Init Win Bytes Forward	No. of bytes in initial window in the fwd direction
22	Flow IAT Mean	Mean time between two packets in the flow
23	Flow IAT Max	Max time between two packets in the flow
24	Fwd IAT Mean	Mean time between 2 packets in the fwd direction
25	Fwd IAT Max	Max time between 2 packets in the fwd direction

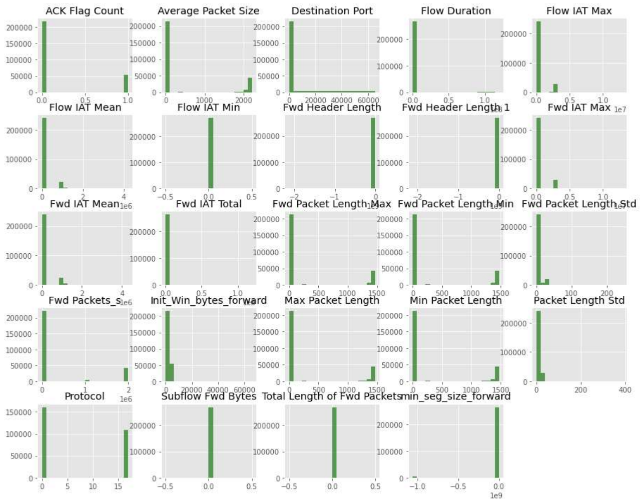


Figure 2: Distribution of values in each feature.

There are the visual representations of the features that each column have different values as we can see in the figure 2 that each graph is showing each feature and we employed [24] to calculate the importance of each feature in the dataset.

Machine learning can satisfy a high-performance rate by finding the correlation on raw data automatically. Therefore, with the proposed model, the accuracy in detecting DDoS attacks have further improved.

Figure 3 displays a correlation. The heatmap is a plot that takes in the correlation of the features and plots based on the values. The colour stripe on the side of the head map signifies the range of values of the correlation term with the help of colours. It means the crimson colour has a correlation of 1.0, and the deep blue colour has a correlation of -1.0.

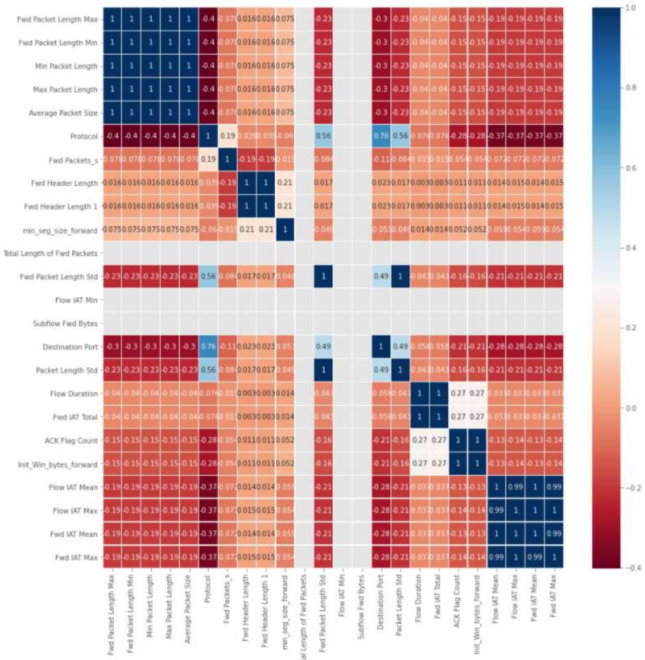


Figure 3: Correlations of the features

The training dataset is manually manipulated using the approach of Sharafaldin [25]. The benefit of using the method is that the input dataset would become competent for intrusion detection and would fit the purpose of the research. Moreover, the approach helps to obtain labelled intrusion datasets for the network at a low cost.

3.3 Performance metrics

We evaluate the performance of our proposed model based on the following metrics:

- Precision (P): Defined as the % ratio of the number of true positives (TP) records divided by sum of true positives (TP) and false positives (FP) classified records.
- Recall (R): Defined as the % ratio of the number of true positives records divided by sum of true positives (TP) and false negatives (FN) classified records.
- F-Score (F): Defined as the harmonic mean of precision and recall and represents a balance between them.

- Accuracy: Defined as the percentage of correctly classified records over the total number of rows.

Metrics used for the proposed model are defined as follows [3]

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (3)$$

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \quad (4)$$

Where True Positive (TP) and True Negative (TN) indicate the values that are predicted correctly. In contrast, False Positive (FP) and False Negative (FN) represent misclassified events [27] [28].

3.4 Classification

We used different evaluation measures to evaluate the performance of the trained algorithms [22] that how to detect the attacks by giving the test data without attacks. We calculated the accuracy of how accurate attacks were detected. Confusion matrix library [26] used for getting the count of true positive which is the detected attack by the algorithm is same as an original attack or false positive. The detected attack by the algorithm is not same but detected as an original attack, false negative which is the detected attack by the algorithm is not same as an original attack but assigned other one, and true negative which is the detected attack by the algorithm is same as the original attack of other class.

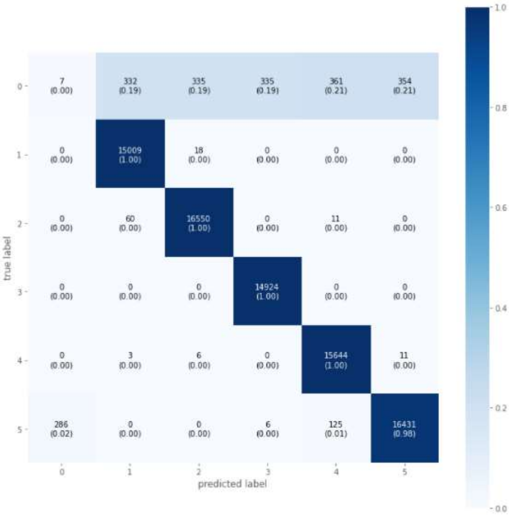


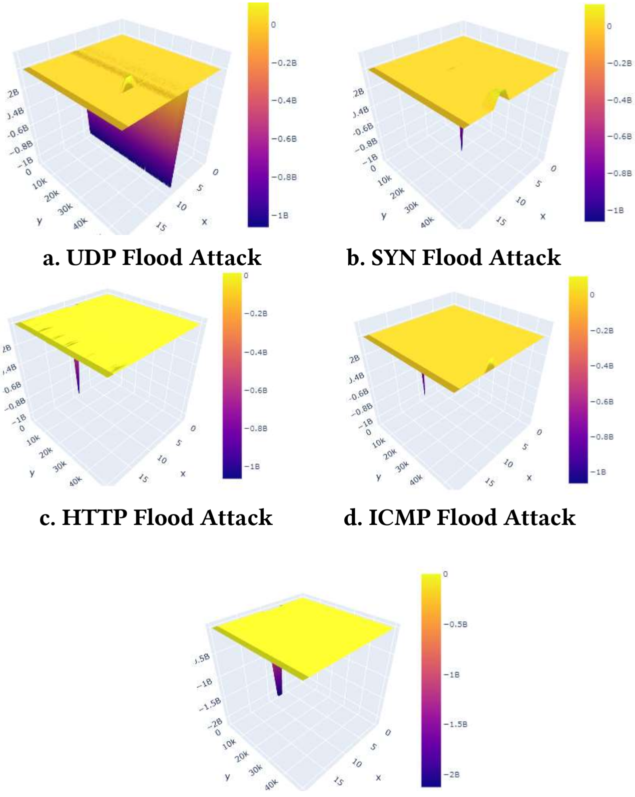
Figure 4: Confusion Matrix

The confusion matrix (CM) [1] is used to characterize the performance of our proposed model during testing. The CM is a 2x2 dimensional matrix representing the relationship between the predicted and actual value as shown in figure 4.

By using the CM effectively, we have calculated values of accuracy along with the ratio of TP, TN, FP and FN. The dark blue coloured boxes in the diagram are the correct prediction with count, percentage and other boxes are the wrong count, predictions as class level. We can see the numbers at y axis and x axis, there are the encoded numbers of attacks.

In the classification report, we used precision which is the percentage of correct detection with false positive, recall is the percentage of correct detection of attacks with the false negative, and F1 measure, which is the mean of precision and recall [26]. Random Forest classifier works with a Meta estimator way and fits number of the decision trees on subset of the data and it uses averaging technique to improve the prediction of class [8]. Random Forest is a machine-learning algorithm combining two decision tree ideas and ensemble learning. Based on different features, the decision trees may manipulate differently on the dataset.

It is possibly the decision trees in the random forest give different prediction on the same data. The forest contains many decision trees that use randomly picked data attributes as their input. It has a collection of trees with controlled variance. Finally, the result of classification can be decided by majority voting or weighted voting. The advantage of random forest [24] is that the variance of the model decreases as the number of trees in the forest increases, while the bias remains the same. In addition, random forests have many other advantages, such as a low number of parameters and resistance to over-fitting. For example, five out of nine decision trees give label Benign to specific traffic, and the forest can predict the traffic is benign.



a. UDP Flood Attack

b. SYN Flood Attack

c. HTTP Flood Attack

d. ICMP Flood Attack

e. POST Attack

Figure 5: Five types of attack visualization used for proposed model in 3D surface plots that displays the attacks in dark blue colour.
a-UDP Flood Attack; b-SYN Flood Attack; c-HTTP Flood Attack; d-ICMP Flood Attack; e-POST Attack

We defined the low variables columns function and calculated the low variance features with threshold of 90%. When we applied this function, there was 3 low variance features named as 'Total Length of Fwd Packets', 'Flow IAT Min', 'Subflow Fwd Bytes'. We removed these features from the data and selected the other features for the training and testing. To visualize each of the attack in three-dimensions, we used the plotly library of the python with offline version of pretty printing to show on Jupyter Notebook. We separated each attack data and plot with 3D. The plot is showing x, y and z coordinates with values of the three features above respectively associated with each attack.

Figure 6 shows the pseudo code for the proposed model using decision tree Gini combines with entropy to produce the high performance in detection.

```
Gini Entropy Pseudocode

collection of different attacks data from web server.
perform combining all attacks for data pre-processing.
perform exploratory data analysis.
perform feature selection          $s^2 = \frac{\Sigma_{i=1}^{n}((x_i - x)^2}{(n-1)}$
perform hybrid decision tree and random forest model
    compute Gini        $\text{Gini}(K) = \Sigma_{K=1}^{N} P_{i,K}(1 - P_{i,K}) = 1 - \Sigma_{K=1}^{N} P^2{}_{i,K}$
    compute Entropy     $H(X) = -p \log_2(p) - q \log_2(q)$
compute                 $\text{Accuracy} = \frac{(TP+TN)}{(TN+TP+FN+FP)}$
compute                 $\text{Precision} = \frac{TP}{(TP+FP)}$
compute                 $\text{Recall} = \frac{TP}{(TP+FN)}$
compute                 $F1 = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$
end
```

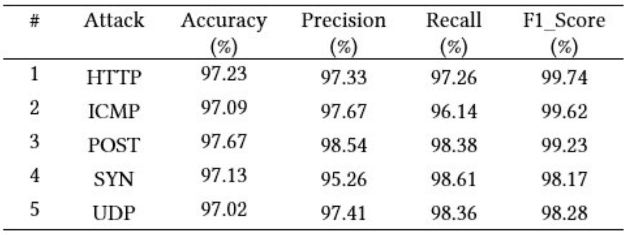
Figure 6: Pseudo code for the proposed model using Gini Index and Entropy [8] [29]

3.5 Results and discussion

We used a machine learning hybrid decision tree and random forest to simulate the proposed model. In the simulations, the proposed model basically performs binary classification where it classifies each input test sample as "benign" or "attack" in the testing phase. The evaluation metrics defined in the previous section, i.e., *accuracy, precision, recall, F1_Score, true positive rate, false-positive rate* and *error rate*, are used to measure the detection performance of the model for classification are used to measure the detection performance of the model.

The accuracy of all data has been calculated in percentage by the total number of correct data classification among the total number of classification [30]. The accuracy is the measure of the performance of the classifier. The high value of accuracy indicates the good performance of the classifier. In the experiment, the simulated model was trained with a total of 189,824 samples and tested with 81,353 samples. The training samples were 70% from csv file. The model was then tested with 81353 test samples. These test samples were 30% extracted from final csv file. Table 3 shows the results of our experiments with accuracy, precision, recall and F1-Score using the proposed model.

TABLE 3: Reported accuracy, precision, recall and F1_Score

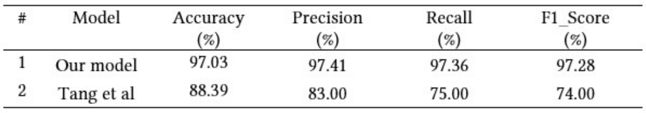


#	Attack	Accuracy (%)	Precision (%)	Recall (%)	F1_Score (%)
1	HTTP	97.23	97.33	97.26	99.74
2	ICMP	97.09	97.67	96.14	99.62
3	POST	97.67	98.54	98.38	99.23
4	SYN	97.13	95.26	98.61	98.17
5	UDP	97.02	97.41	98.36	98.28

(https://github.com/thaisonchu/DDoS-Research)

This experimental results show that the proposed model is able to detect attacks using the reduced CICDDOS2019 dataset, with at least 97% accuracy rate, whilst other deep learning approach Self-taught Learning (STL) using NSL-KDD dataset achieved only the maximum of 88.39% accuracy rate [12]. Even though two experiments implemented in two different datasets, and our method might not superior in their application, however with using our proposed model the result would be positively. Please see the comparison in table 4.

TABLE 4: Compare metrics between our model to Tang's model [17]



#	Model	Accuracy (%)	Precision (%)	Recall (%)	F1_Score (%)
1	Our model	97.03	97.41	97.36	97.28
2	Tang et al	88.39	83.00	75.00	74.00

4 Conclusion and Future Work

In this paper, we presented a hybrid of machine learning decision tree and random forest approach for DDoS detection on vulnerable network. The detection was based on binary classification, thus identifying normal and threat patterns. The proposed model was tested against CICDDOS2019 dataset demonstrating over 97% accuracy. It was able to identify successfully different types of attacks and showed a good performance in terms of true and false positive rates, and impressive precision and recall values. For future developments, the system will be put in a simulated network test environment in order to determine the reliability of our model in real-time situation.

REFERENCES

[1] João Paulo A. Maranhão, João Paulo C.L. da Costa, Elnaz Javidi, César A. Borges de Andrade, and Rafael T. de Sousa, "Tensor based framework for Distributed Denial of Service attack detection," *J. Netw. Comput. Appl.*, vol. 174, 2020, doi: 10.1016/j.jnca.2020.102894.

[2] Bo Hang *et al.*, "An Enhanced SYN Cookie Defence Method for TCP DDoS Attack," *J. Netw.*, vol. 6, 2011.

[3] Vinícius de Miranda Rios, Pedro R.M. Inácio, Damien Magoni, and Mário M. Freire, "Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms," *Comput. Netw.*, vol. 186, 2021, doi: 10.1016/j.comnet.2020.107792.

[4] M.S.P.S. VANI NIDHI, "DETECTION OF ANOMALY BASED APPLICATION LAYER DDoS ATTACKS USING MACHINE LEARNING APPROACHES," 2016.

[5] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut, “DDoSNet: A Deep-Learning Model for Detecting Network Attacks,” 2020.

[6] Aroosh Amjad, Tahir Alyas, Umer Farooq, and Muhammad Tariq, “Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm,” *ICST Trans. Scalable Inf. Syst.*, vol. 0, 2019, doi: 10.4108/eai.29-7-2019.159834.

[7] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, “A survey of distributed denial-of-service attack, prevention, and mitigation techniques,” *Int. J. Distrib. Sens. Netw.*, vol. 13, no. 12, p. 155014771774146, Dec. 2017, doi: 10.1177/1550147717741463.

[8] Liying Li, Jianying Zhou, and Ning Xiao, “DDoS Attack Detection Algorithms Based on Entropy Computing,” *Inf. Commun. Secur. Notes Comput. Sci.*, 2007, doi: 10.1007/978-3-540-77048-0_35.

[9] Christos Douligeris and Aikaterini Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art,” *Comput. Netw.*, vol. 44, 2003, doi: 10.1016/j.comnet.2003.10.003.

[10] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, “DDoSNet: A Deep-Learning Model for Detecting Network Attacks,” *ArXiv200613981 Cs*, Jun. 2020, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/2006.13981

[11] Stefanos Kiourkoulis, “Use of machine learning to analyse intrusion detection performance,” *Inf. Secur.*, 2020.

[12] Z Chiba, N.Abghour, K.Moussaid, A.El omri, and M.Rida, “A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network,” *Procedia Comput. Sci.*, vol. 83, 2016, doi: 10.1016/j.procs.2016.04.249.

[13] Honours Bachelor of Science (H.B.Sc.), University of Toronto, and 2011 A Report Submitted in Partial Fulfillment of the Requirements for the Degree of MASTER OF ENGINEERING, “Network Intrusion Detection for Distributed Denial-of-Service (DDoS) Attacks using Machine Learning Classification Techniques by Yasar Shahid Hussain,” *Honours Bachelor Sci.*, 2020.

[14] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao, “Survey of network-based defense mechanisms countering the DoS and DDoS problems,” *ACM Comput. Surv.*, vol. 39, 2007, doi: 10.1145/1216370.1216373.

[15] D. Singh, D. R. Patel, B. Borisaniya, and Modi, Chirag, “Collaborative IDS Framework for Cloud,” *Artic. Int. J. Netw. Secur.*, vol. 18, 2015.

[16] Muhammad Aamir and Syed Mustafa Ali Zaidi, “DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation,” *Int. J. Inf. Secur.*, vol. 18, 2019, doi: 10.1007/s10207-019-00434-1.

[17] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, “Deep learning approach for Network Intrusion Detection in Software Defined Networking,” in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, Oct. 2016, pp. 258–263. doi: 10.1109/WINCOM.2016.7777224.

[18] Dileep Kumar G, CV Guru Rao, Manoj Kumar Singh, and Satyanarayana G, “A Survey on Defense Mechanisms countering DDoS Attacks in the Network,” *Article*, 2013.

[19] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

[20] Thapanarath Khempetch and Pongpisit Wuttidittachotti, “DDoS attack detection using deep learning,” *IAES Int. J. Artif. Intell. IJ-AI*, vol. 10, 2021, doi: 10.11591/ijai.v10.i2.pp382-388.

[21] Francisco Sales de Lima Filho, Frederico A. F. Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar, and Luiz F. Silveira, “Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning,” *Secur. Commun. Netw.*, vol. 2019, 2019, doi: 10.1155/2019/1574749.

[22] S. Kiourkoulis, “DDoS datasets: Use of machine learning to analyse intrusion detection performance,” p. 81.

[23] Troy Nunnally, Penyen Chi, Kulsoom Abdullah, A. Selcuk Uluagac, John A. Copeland, and Raheem Beyah, “P3D: A parallel 3D coordinate visualization for advanced network scans,” *2013 IEEE Int. Conf. Commun. ICC*, 2013, doi: 10.1109/icc.2013.6654828.

[24] Adele Cutler, D. Richard Cutler, and John R. Stevens, “Random Forests,” *Ensemble Mach. Learn.*, 2012, doi: 10.1007/978-1-4419-9326-7_5.

[25] 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)*, CHENNAI, India, Oct. 2019, pp. 1–8. doi: 10.1109/CCST.2019.8888419.

[26] Yasar Shahid Hussain, “Network Intrusion Detection for Distributed Denial-of-Service (DDoS) Attacks using Machine Learning Classification Techniques by Yasar Shahid Hussain,” *Honours Bachelor Sci.*, 2020.

[27] Tong Anh Tuan, Hoang Viet Long, Le Hoang Son, Raghvendra Kumar, Ishaani Priyadarshini, and Nguyen Thi Kim Son, “Performance evaluation of Botnet DDoS attack detection using machine learning,” *Evol. Intell.*, vol. 13, 2019, doi: 10.1007/s12065-019-00310-w.

[28] A. M. Irfan Sofi Vibhakar Masotra, “Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks,” *Article*, 2017.

[29] P. Bereziński, B. Jasiul, and M. Szpyrka, “An Entropy-Based Network Anomaly Detection Method,” *Entropy*, vol. 17, no. 4, pp. 2367–2408, Apr. 2015, doi: 10.3390/e17042367.

[30] Manikant Panthi, “Identification of Disturbances in Power System and DDoS Attacks using Machine Learning,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, 2021, doi: 10.1088/1757-899x/1022/1/012096.