Detecting DDoS Attacks in Blockchain-enabled IoT Networks using Distributed Intrusion Detection System and Machine learning

CS658A PROJECT REPORT

Team Members:

Manu Shukla-21111040-manushukla21@iitk.ac.in

Divyansh Bisht-21111027-dbisht21@iitk.ac.in

Manthan Kojage-21111039-manthank21@iitk.ac.in

Hrugved Wath-180301-hrugved@iitk.ac.in

Kartavya-180343-kartvya@iitk.ac.in

Under the guidance of

Dr. Sandeep Kumar Shukla

Indian Institute of Technology Kanpur (IIT Kanpur)

April 29, 2022

Keywords: Intrusion Detection System(IDS), IoT enabled blockchain networks, Machine Learning(ML), Distributed Denial of Service(DDoS) attack, XGBoost algorithm, BoT-IoT data

1 Problem being solved

This project aims at building a Machine Learning(ML) based intrusion detection system for blockhain enabled IoT networks for Distributed Denial of Service(DDoS) attacks mainly. Conventional methods are not very effective for the same purpose. So, we try different ML models and perform their comparative analysis to generate an effective model that provides sufficient security against DDoS attacks. Also, a frontend UI is created that displays warning to the user whenever our ML model predicts a DDoS attack.

2 Introduction(Goals and achievements)

IoT and blockhain technology are very popular these days. There are several security problems related to conventional IoT systems that work in a centralized architecture. As a solution to it, the use of blockchain enabled IoT networks has emerged successfully. It helps to decentralize the overall architecture mainly. But, recently these blockchain enabled IoT networks have been prone to various network attacks, specifically DDoS. Several conventional techniques have been developed and tested to avoid it but, none of them are able to completely secure these networks. So, the aim of this project is to develop an Intrusion Detection system for blockchain enabled IoT devices that helps preventing various attacks on these networks(specifically DDoS). The ML models were trained on a common BoT-IoT dataset generated by UNSW. The results were compared and XGBoost algorithm was found out to be the best among all models. An interactive frontend User Interface was built using this XGBoost algorithm. To test our model on realtime data, we used the 95% BoT-IoT dataset that the model was not trained on. Since, the BoT-IoT dataset was a huge dataset that could not be trained on our laptops due to low computational power, we trained our model on only 5% of the data. Rest data was used for realtime testing. We kept some data for testing from that 5% dataset only, upon which we achieved 95.77% accuracy. Our model achieved a great accuracy on the randomly generated real time data too from the 95% dataset. For 100 random inputs, our model was able to predict correctly for 91 entries, thus giving a 91% accuracy on the real time data. Hence, we were conclusive about the fact that our experiments led to successful achievements of our goals.

3 Dataset

The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of the UNSW Canberra. The environment incorporates a combination of normal and botnet traffic. The dataset's source files are provided in different formats, including the original pcap files, the generated argus files and csv files. The files were separated, based on attack category and subcategory, to better assist in labeling process. Further details about the dataset will be provided as the presentation progresses as many different version of the dataset are there now. Bot IoT dataset was generated with the help of Ostinato Tool. Ostinato tool generates realistic data with the help of a cloud server consisting of Virtual Machines and Kali machines. In the Kali machines, DNS, SSH, FTP, HTTP and various other services were deployed. Node red tool was used for the simulation of IoT devices and MQTT protocol was used for communication between IoT devices. Five IoT scenarios were implemented in cloud server to collect the dataset: A weather station, A smart fridge, Motion activated lights, A remotely activated garbage door, A smart thermostat. We used this dataset because this version was mainly built of the Dos/DDos attacks which originally was the motive to solve. We used two datasets from UNSW where one was a completely imbalanced dataset, and the other was a balanced one. The links to the same are provided below:
Dataset-1(imbalanced), Dataset-2(balanced)

4 Brief description of the problem

The centralized nature of IoT networks provide various difficulties such as privacy, security, and single point of failure. So, to overcome it, blockchain enabled IoT networks were combined. But, due to so many nodes, there are chances of gaining access of the less secure IoT nodes and performing a DDoS attack on the server using it. Such cases have rose recently and it heavily targets the security levels of blockchain enabled IoT network. A need for some effective intrusion detection system is there that can provide some good security against DDoS attacks mainly. The conventional methods do not provide very secure solutions. The use of Machine Learning can be performed here as the BoT-IoT network dataset is very huge. ML models can easily work on this data to distinguish between attacks and non-attacks. So, a need for some good ML model is necessary to build a good intrusion detection system.

5 Methodology to obtain required solution

We performed feature engineering on the dataset like, we kept one feature from a set of highly correlated features. Then, we applied CLAMPING which is pruning the extreme values to reduce the skewness of some distributions. Then we applied log function to nearly all numeric values, since they are all mostly skewed to the right. Then we reduced the cardinality of the categorical features to less than or equal to 6 since all of them had mostly less than 6 classes that formed the majority. Also the dimensionality will not explode when encoding is done later. Then we trained our model on 2 datasets(balanced and unbalanced). Better dataset was balanced one so, we used that one only to design our UI later. Several models were tried and tested. The best accuracy was obtained by using XGBost algorithm. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. Finally we saved the model using pickle file and used it later in our UI frontend app. We tried testing on realtime unknown data that the model was not trained on by testing it on the remaining 95% UNSW dataset.

6 Architecture:

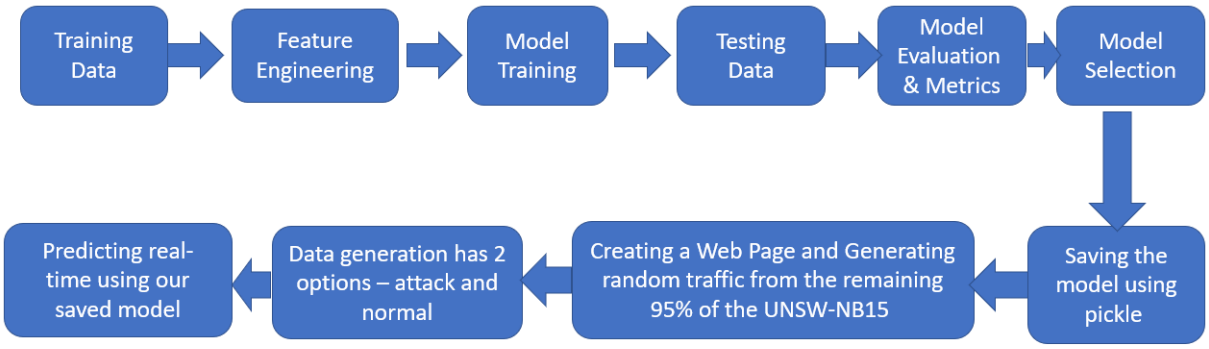

Figure 1: Web App Architecture

7 Experimental Setup

The entire model was trained using Google Collab, the frontend was developed using streamlit and the client server app was built using React. Actual BoT-IoT devices were not available, so we used the

BoT-IoT dataset that was already generated from real IoT devices. Everything was designed on an 8 GB RAM laptop without any other physical resources.

7.1 Client-Server Application Architecture

The architecture we built as an Experimental Setup was aimed to mimic the Blockchain-enabled IoT network itself where, the clients are corresponding to the IoT nodes generating traffic (transaction requests) in the network and server is corresponding to the mining pools residing at the edge of the network(in which transactions waits to get verified). The clients generates network traffic (packets) which before reaching to the server goes through ML model(IDS) which we integrated at the back-end using nodejs. This model classifies the traffic as attack/normal. Further, It also classifies the attacks into categories and sub-categories. The clients are logged in the Database using MYSQL. Upon classifying the traffic, the UI displays warning whenever the ML model predicts an attack. Also, The counters for the categories and sub-categories of the attacks are increased on encountering the corresponding attack in the UI.

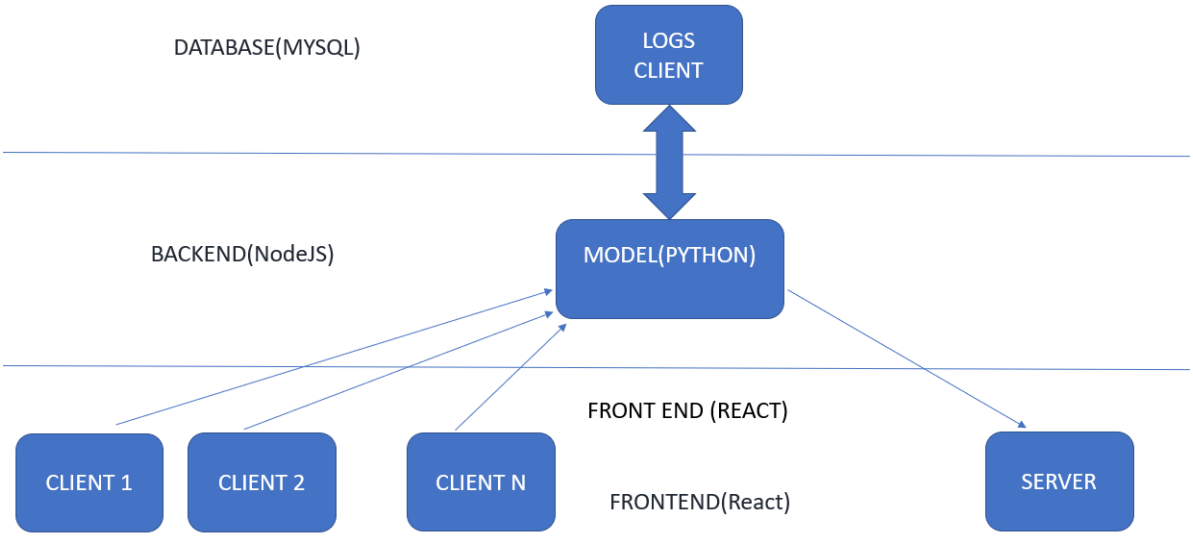

Figure 2: Client-Server Application Architecture

The working of this client-server application (which we built through React) is demonstrated and the link of the video is provided below: Video of working of the client-server app

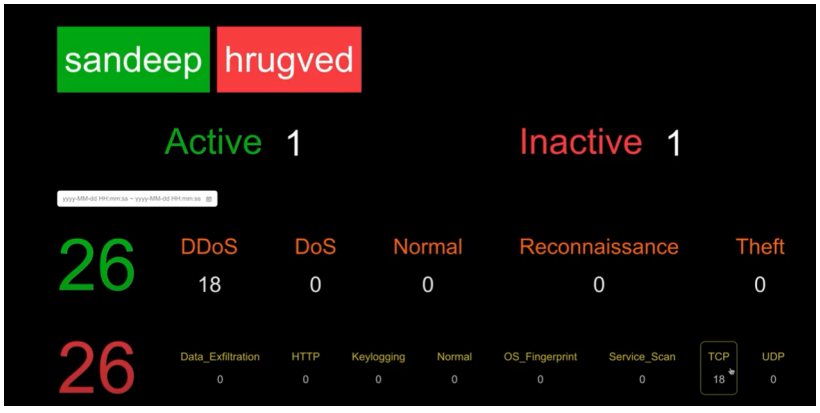

Figure 3: Snippet of the demonstration of the client-server app

The frontend we developed using streamlit is demonstrated below: Video of the frontend(streamlit)

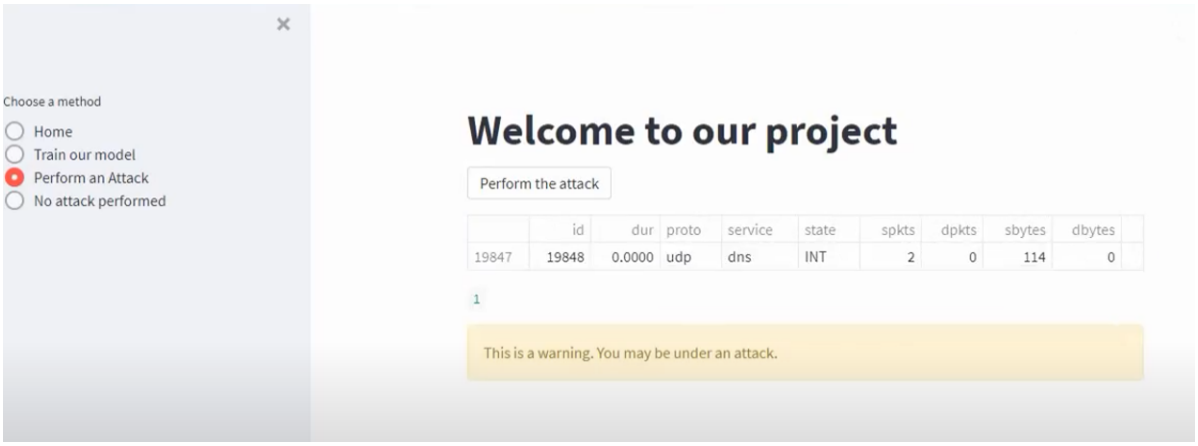

Figure 4: Snippet of the demonstration of the frontend

8 Results

On evaluating the performance of different ML models on the Dataset-1(imbalanced), We observed that

	Model	Accuracy	Recall	Precision	F1-Score
0	Random Forest	99.99%	67.76%	100.00%	76.20%
1	Gaussian Naive Bayes	99.62%	94.67%	51.67%	53.13%
2	Decision Tree(Information Gain)	99.99%	68.69%	97.61%	76.84%
3	Decision Tree(Gini Index)	99.99%	68.22%	96.42%	76.17%
4	Gradient Boost	100.00%	96.73%	99.50%	98.08%
5	Multi Layer Perceptron	99.99%	97.66%	86.17%	91.13%
6	Support Vector Machine	99.99%	53.74%	99.99%	56.95%
7	Logistic Regression	99.99%	54.20%	74.99%	57.20%

Figure 5: Accuracy on unbalanced BoT-IoT UNSW dataset

the accuracies for the models were very high whereas the F1-Scores(macro) for the same were on low. This was clearly indicating that the models where not performing good for one of the labels. Also, very high accuracies also meant that the models were performing good on the majority of the data. This evaluations was hinting of the dataset being biased. On observation, It was found that there were lacs of attack entries in the dataset whereas, the normal entries were in hundreds. We then chose a new dataset which was also an Bot-Iot dataset(Dataset-2(balanced)) for the performace evaluation for our models.

The evaluation done on this dataset (which through feature engineering we found was unbiased) gave us high accuracies and high F1-Scores for almost every model. XGBoost was giving the highest accuracy,precision,recall and F1-Scores for this dataset. We proceeded with XGBoost as our ML model for classifying the network traffic as attack or normal.

	Accuracy	Recall	Precision	F1-Score	Time to train	Time to predict	Total time
Logistic Regression	93.75%	93.75%	93.89%	93.62%	16.3	0.0	16.4
kNN	94.60%	94.60%	94.58%	94.59%	0.0	102.3	102.3
Random Forest	92.17%	92.17%	92.98%	91.84%	8.8	0.2	9.0
Gaussian Naive Bayes	83.77%	83.77%	84.41%	83.98%	0.2	0.1	0.2
Decision Tree (Information Gain)	92.99%	92.99%	93.43%	92.77%	1.2	0.0	1.2
Decision Tree (Gini Index)	93.47%	93.47%	93.69%	93.31%	1.0	0.0	1.0
SVM	93.63%	93.63%	93.89%	93.48%	22.9	0.0	23.0
XGBoost	95.77%	95.77%	95.76%	95.76%	15.0	0.1	15.1
Multi-Layer Perceptron	94.53%	94.53%	94.50%	94.50%	53.8	0.0	53.8

Figure 6: Accuracy on balanced BoT-IoT UNSW dataset

9 Conclusion

XGBoost gave the best accuracy(95.77%) and also the latency(0.1s) was low too. Our model performed well for the BoT-IoT dataset generated by UNSW. The only shortcoming that this model could face is that as new techniques keep developing in cyber security related field, so, the model must be kept updated from time to time. Also the model needs to be integrated with fog computing architecture of IoT devices so that the block-chain traffic coming at the fog nodes is able to identify the attack and only the normal traffic gets sent to the sender.

10 Artifacts

The links to the artifacts used are provided below:

- Research Paper: A distributed intrusion detection system to detect ddos attacks in blockchain-enabled iot network[1]
- BoT-IoT Dataset-1(imbalanced)
- BoT-IoTDataset-2(balanced)
- Models Trained(GitHub)
- Client-Server App(GitHub)
- Web App(GitHub)

References

[1] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect ddos attacks in blockchain-enabled iot network," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 55–68, 2022.