IoT Intrusion Detection System Using Machine Learning

DSCI599 - Machine Learning for a Secure Internet

Emerson Jin

https://github.com/emersonjin01/DSCI599_project

Abstract

As more and more devices connect to the Internet, more and more vulnerabilities will become present. This includes Internet of Things devices as they become more prevalent in our everyday lives. In order to defend against malicious attacks, many mechanisms have been created and implemented such as firewalls and intrusion detection systems. This paper aims to evaluate machine learning models and their effectiveness as intrusion detection systems in order to defend against potential attacks.

Introduction

Internet of Things (IoT) devices often refer to hardware such as sensors or cameras that are connected to the Internet. Common examples include security cameras, doorbell cameras, Blue-tooth connected ovens, and smart locks. Although very convenient for everyday life, IoT devices are vulnerable to malicious traffic as a result of being a part of the Internet. For example, there have been reports of baby cameras being remotely controlled by hackers, allowing them to see the camera's live footage and communicate using them. IoT devices can also be infected with malware to be used to conduct attacks as a part of a botnet.

In order to deal with malicious traffic that can exploit vulnerabilities in these types of devices, mechanisms like firewalls and intrusion detection systems have been proposed and

implemented. Firewalls are a common type of defense for Internet connected devices that filter packets based on sets of rules. Although firewalls can be effective in many scenarios, it is not entirely comprehensive. As a result, it is often implemented alongside other mechanisms such as intrusion detection systems.

Intrusion detection systems help detect any unwanted and malicious traffic by monitoring inbounding traffic and flagging it if it seems suspicious. A common type of IDS you might see in organizations and devices is a network based IDS. This type looks at network traffic such as packets and flows between devices and analyzes the characteristics of this data to detect it for anything potentially harmful. Some systems analyze network traffic that matches specific signatures reflective of past malicious activity and block it as a result. Another type of IDS utilizes machine learning to learn from past malicious and benign traffic and categories new traffic based on that data.

Machine learning based intrusion detection systems can be very effective due to the use of models that can learn from past network attacks and generalize it to similarly new attacks. The more data machine learning models get, the more it can learn regarding benign and harmful traffic. Especially since network traffic is abundant, machine learning can be very effective in cybersecurity. This paper aims to find which machine learning model seems to be the most effective in intrusion detection systems and does so in the context of IoT devices.

Dataset

In order to test the effectiveness of different machine learning models for an IoT IDS, I used the Bot-IoT dataset for training and testing. This dataset contains normal and attack network traffic produced in a research testbed utilizing virtual machines to simulate IoT devices

and cybersecurity attacks. Some examples of these simulated devices include a weather station, smart fridge, motion activated lights, garage door, and a smart thermostat. A botnet was utilized to simulate attacks expected against IoT devices, and these included probing attacks, denial of service attacks, and information theft. After the packet data was collected, it was summarized into network flows by the Argus security tool.

In total, there were 43 different features. Some important features of this flow data include protocol, total number of packets and bytes, and flow duration. Many aggregated and Argus-generated flow features were included as well such as standard deviation of aggregated records, source-to-destination packets per second, total number of packets per destination port, and total number of packets per source IP. The label to be predicted was “attack” which represented a 0 for normal traffic and a 1 for attack traffic.

Data Cleaning

In order to have appropriate data available for training, data cleaning had to be done. Seven features that were invalid or irrelevant were removed. Some examples included the numerical representation of feature flags, row identifier, source IP address, and Argus sequence number. The traffic category and subcategory for attacks were also removed.

The source and destination port number had to be converted since they had high cardinality as categorical variables. Each port number was categorized as well-known, registered, or dynamic/private. The respective ranges are 0-1023, 1024-49151, and 49152-65535.

Flows involving the ARP, ICMP, and IPV6-ICMP protocols were removed. Each of these protocols’ flows either had only attacks or non-attacks associated with them. This is problematic

since the models will not be able to differentiate attack and non-attack flows from the features for these categories.

Methodology

The three models considered were logistic regression, XGBoost, and support vector machines. Logistic regression models the probability for a given input using the logistic function and classifies it based on that probability. XGBoost utilizes a bunch of decision trees and bases its predictions on the average results of those trees. Support vector machines find some sort of hyperplane that separates the classes based on the feature space. These three models are very common in machine learning and were selected due to their ability to do binary classification.

The dataset was initially divided into the training and testing sets with a 7:3 ratio. The split was stratified in order for the attack and non-attack flow ratio to be the same in both the training and testing datasets.

The dataset also has a heavy class imbalance. Only 422 of the flows are non-attack flows while 3,658,827 flows are considered attacks. To mitigate the issues that class imbalance might bring, the minority class was upsampled to 1000 flows and the majority class was downsampled to 1000 flows.

With this data, three models were configured and trained. The logistic regression model was implemented with a L1 penalty and 5-fold cross validation was used. XGBoost also was configured with 5-fold cross validation that tested multiple different alpha values. The SVM model also tested different hyperparameter values with 5-fold cross validation.

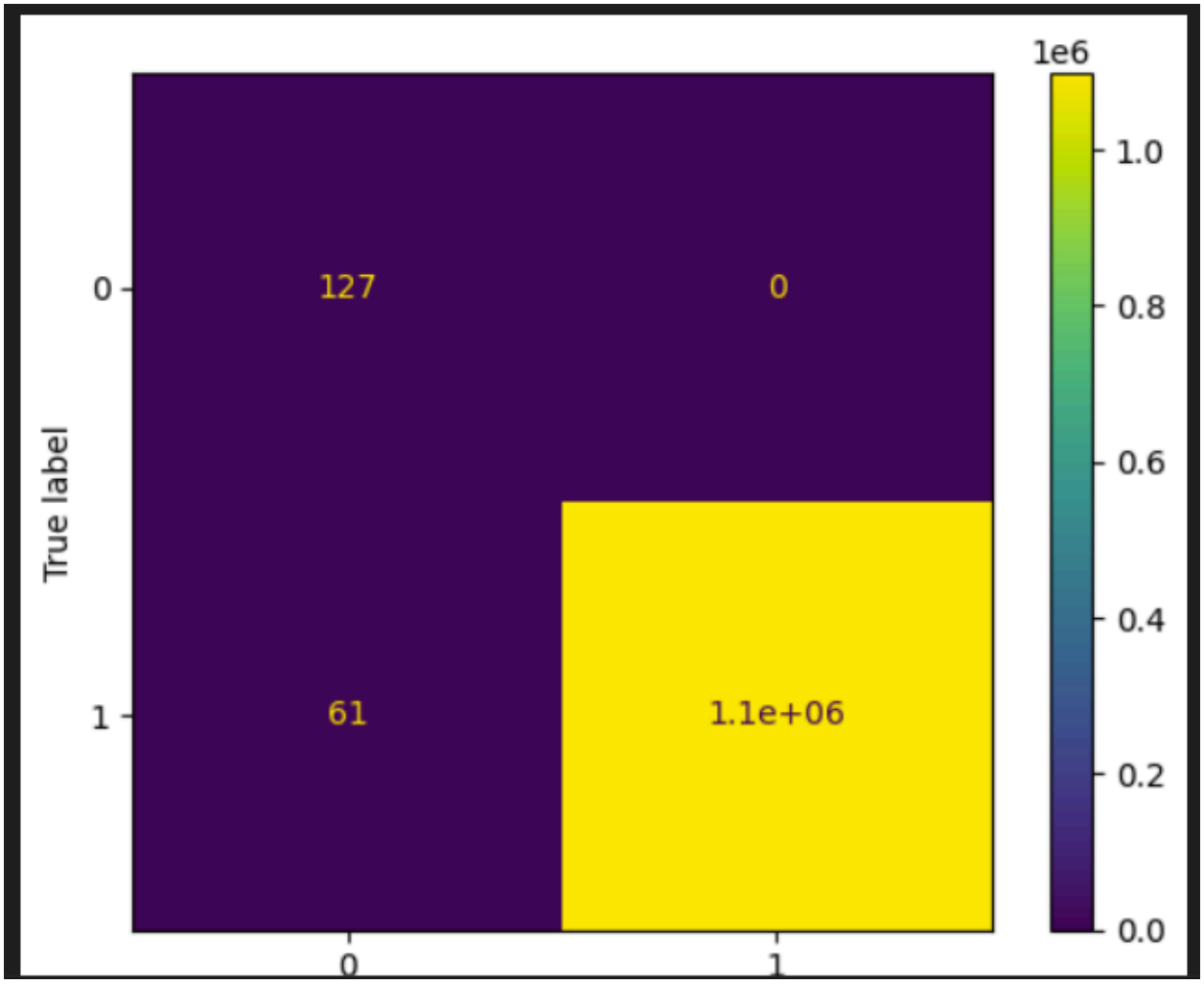
Results

Logistic Regression

```
Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       0.68      1.00      0.81       127
           1       1.00      1.00      1.00   1097648

    accuracy                           1.00   1097775
   macro avg       0.84      1.00      0.90   1097775
weighted avg       1.00      1.00      1.00   1097775
```



The logistic regression model achieved a near perfect accuracy, classifying almost all flows as either benign or attack correctly. However, the model resulted in a precision score of 0.68, which means that there were a significant number of false negatives. This makes sense given that the model predicted a decent amount of benign flows as attack flows.

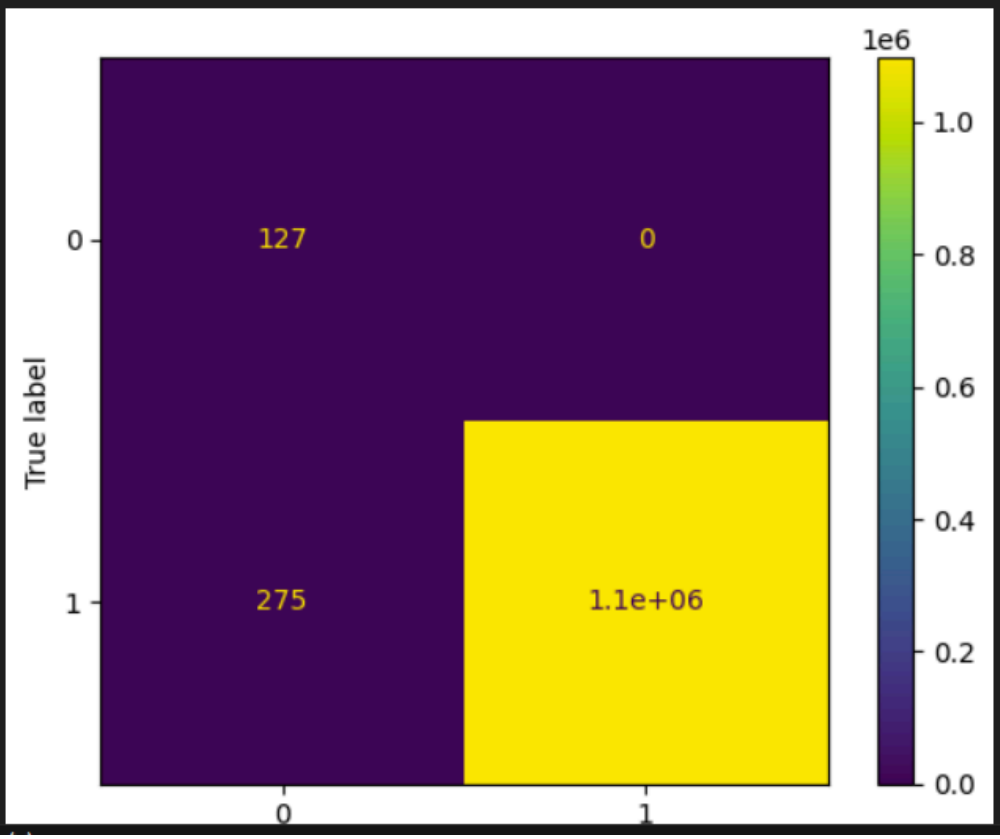
XGBoost

```
Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       0.32      1.00      0.48       127
           1       1.00      1.00      1.00   1097648

    accuracy                           1.00   1097775
   macro avg       0.66      1.00      0.74   1097775
weighted avg       1.00      1.00      1.00   1097775


<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x281b75dce10>
```



XGBoost also achieved a near perfect overall accuracy. However, when compared to logistic regression, its precision was worse.

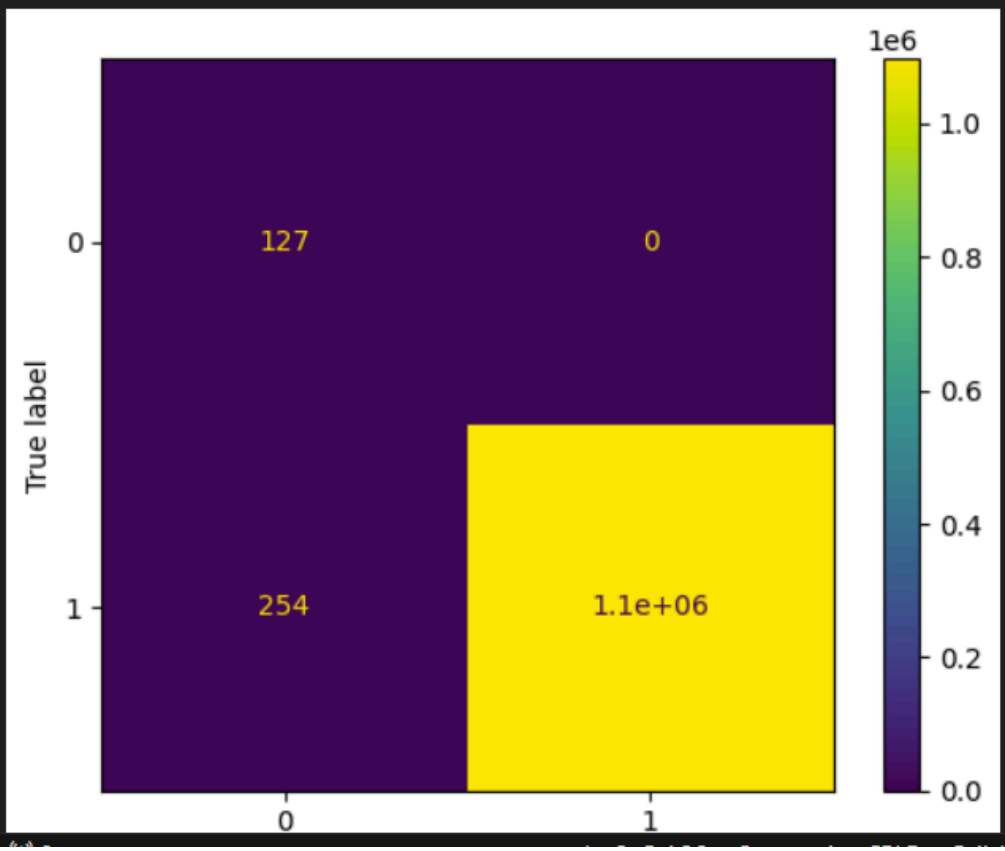
Support Vector Machine

```
Accuracy: 1.00
Classification Report:
              precision    recall  f1-score   support

           0       0.33      1.00      0.50       127
           1       1.00      1.00      1.00   1097648

    accuracy                           1.00   1097775
   macro avg       0.67      1.00      0.75   1097775
weighted avg       1.00      1.00      1.00   1097775


<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x281b528a250>
```



The support vector machine model performed nearly the same as the XGBoost model. Despite achieving a high overall accuracy, there were a significant number of false negatives when compared to true negatives.

Discussion

All the models performed very well in terms of overall accuracy. When it comes to precision, logistic regression out performed the other models by a significant margin. However, in general, all of the models performed poorly when it comes to classifying benign attacks. A large reason for this is the extreme class imbalance in the dataset when it comes to attack and non-attack flows. There are also a very limited number of benign flows as well, making it difficult for the models to learn and generalize what is normal traffic behavior.

Despite upsampling and downsampling being used, the models still did not perform well when it came to classifying false negatives. Important considerations to keep in mind in the future would be to consider datasets in the future that are more reflective of actual traffic behavior. Unfortunately, Bot-IoT consists mostly of attack traffic which usually is not reflective of typical IoT device traffic.

Conclusion

Although all models have very high overall accuracy and true positive rates in classifying attacks, logistic regression performed the best in classifying benign flows. Despite this result, the model still performed relatively poorly in classifying benign flows which would be problematic as an actual intrusion detection system. More reflective and high quality datasets reflecting real world traffic should be considered for better results.

References

Khraisat, A., Gondal, I., Vamplew, P. *et al.* Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur* 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7

Koroniotis, Nickolaos, et al. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." *Future Generation Computer Systems* 100 (2019): 779-796.

Li, Jundong, et al. "Feature selection: A data perspective." *ACM computing surveys (CSUR)* 50.6 (2017): 1-45.

Peterson, Jared M., Joffrey L. Leevy, and Taghi M. Khoshgoftaar. "A review and analysis of the bot-iot dataset." *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2021.