

Network Anomaly Detection

Naveen Ajay karasu

Abstract

In today's world the cyber-attacks are becoming more and more complicated and hard to identify. The attacks targeting large networks have become more and more frequent which use advanced methods. Because of this, Traditional cybersecurity systems can no longer be useful in detecting these dynamic attacks in real-time due to the large amount of data generated by modern networks [1]. Therefore, for my project I was planning to explore how different big data analytics and machine learning techniques can be used to detect network anomalies in real-time to provide stronger security for the organizations and their network. The main objective for this project will be trying to use the latest technologies to identify different malicious activities like Distributed Denial of Service (DDoS) attacks, botnets and other network intrusions that can severely damage both the performance of the network and its security.

Introduction

In today's age of cybersecurity the detection of network anomalies is very important to prevent cyberattacks and to ensure the integrity of the network. As part of this course, in this project we are trying to focus on the development of an anomaly detection model using machine learning [3]. By using AI, we aim to automate the process of detecting the anomalies within network traffic. Which can help in providing early warning for the signs of cyber intrusion. The main goal is to enhance the capabilities of analysts by creating a robust detection system that can be used to classify normal and anomalous network activities.

The rate at which organizations have been experiencing cyber threats make anomaly detection crucial in protecting networks. The key threats that imply break into the network integrity and unauthorized access are presented by the Network anomalies like DDoS attacks, unauthorized access & data exfiltration. Standard knowledge-based offense identification techniques are generally not enough effective as most contemporary cyber threats are non-stationary and cannot be predicted, therefore specifying the requirement in AI-based strategies [6]. In our case our motivation is to use machine learning in order to automate the process of looking for anomaly behaviors, which in turn ascertain the early signs of suspicious activities that cybersecurity specialists can help to prevent.

Area of application

One of the greatest benefits of network anomaly detection through AI is that they can be updated over time to account for new cyber threats. Present dangers are resilient and change as often as possible and one can easily devise a new approach to implement as opposed to others. While static solutions cannot be used to detect them. The AI models can discern even a slight anomalies in traffic flow and expand their database of attack types. Compared to large amounts of usual network data, AI-powered models can find changes that are usually slight, providing notifications for the cybersecurity analyst to investigate. Target 'Preemptive' capability comes in handy within a context where attacks employ new strategies to gain access into the networks.

Dataset

For developing our anomaly detection system, the data set used was known as the "Network Anomaly Detection" and can be downloaded from Kaggle which is a contribution of Anushonkar and CTU-13 dataset[5]. This dataset consists of thousands of records of the network traffic and each record is either normal or anomalous. This dataset enables devising the machine learning based algorithm to differentiate between regular and malicious actors. For all experiments, we split the data into a training set (70%), validation set (15%) and the testing set (15%). Such division lets us train the model carefully, and check it on new data, which would enhance its reliability when used in practice. NetFlow[6], Which is a protocol system that can be used to collect and display information about network traffic as it flows in or out of an interface details like source and destination addresses, packet counts, and data volumes. This data can be essential for understanding both normal and abnormal network patterns. As it's hard to collect any abnormal network over a personal router. We can find publicly available datasets like CTU-13, which is a collection of botnet traffic data [5] and some other datasets from Kaggle related to network anomalies. We can use these readily available dataset for intrusion detection analysis. We can use platforms like Azure HDInsight, Apache Hadoop, spark and Splunk to process and analyze large-scale, real-time traffic data [1]. These datasets will allow us to study network anomalies which covers a wide range of scenarios like common and rare attack patterns [2].

Data Preprocessing

We preprocess the CTU-13 botnet dataset for analysis and model training [5]. The dataset is first obtained from Kaggle which comprised several parquet files containing the network flow records of various different botnet families with each file capturing features such as duration ('dur'), protocol ('proto'), direction ('dir'), state, packet counts, byte counts, and labels. Due to the dataset's complexity and large size. We need to have a structured approach for maintaining and to have high data quality.

Packet Count	Protocol Type	Connection Duration	Label
45	TCP	0.2 seconds	Normal
67	UDP	0.4 seconds	Anomalous
128	TCP	1.2 seconds	Normal
34	ICMP	0.1 seconds	Anomalous

Table 1: Sample subset of data from the dataset

So, we will have consistency throughout preprocessing and dataset. The first step involves the combining individual files into a single DataFrame which helps us to ensure the feature and data type consistency across all records. This consolidation resulted in a dataset with over 10.5 million records. We preserved all relevant features from the original files for a comprehensive analysis across botnet families.

Model Evaluation

In this project, I evaluate three distinct machine learning models—K-Means, DBSCAN, and Isolation Forest—to analyze botnet traffic patterns within the CTU-13 dataset. Each model serves a unique role in clustering similar data points and detecting potential anomalies. After training the models, I integrate them into a Flask-based web application, which allows me to test each model's predictions interactively with new data points. I focus my evaluation on each model's strengths and metrics that help measure their clustering or anomaly detection effectiveness.

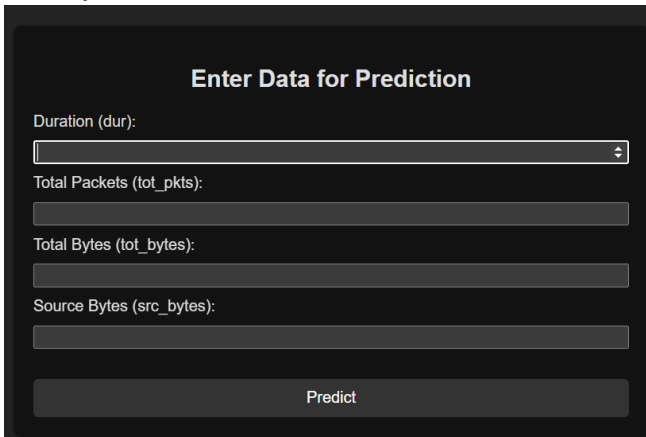


Fig 1: The Form for user to manually input the network information to test the model

We evaluated each model within the Flask application, where the /predict route processes form inputs, makes predictions using each model, and displays results in real-time. This interactive approach allows me to assess the behavior of each model with new data points and validate its predictive capabilities.

Result

The results of my analysis reflect the strengths of each model in clustering and detecting botnet traffic patterns. By using K-Means, DBSCAN, and Isolation Forest, I gain a multi-dimensional understanding of botnet behaviors in the CTU-13 dataset, with each model contributing unique insights into clustering and anomaly detection.

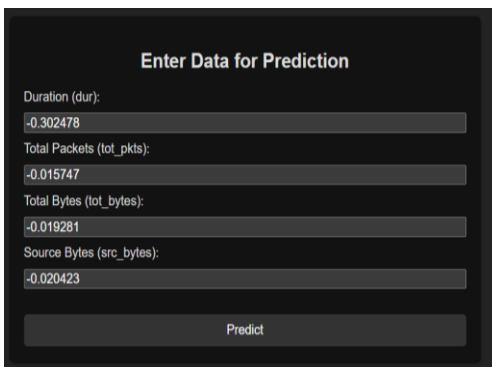


Fig 2: Predict form with Random sample data

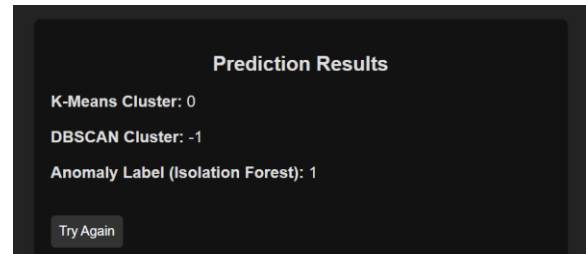


Fig 3: Prediction result from the three trained models

Through the Flask app's /predict route, I interact with each model's predictions on the result.html page, testing various inputs and observing cluster and anomaly labels in real-time. This interactivity helps validate each model's effectiveness and offers a practical view of how they perform with new data points.

Conclusion

This project demonstrates how machine learning can effectively analyze and detect botnet traffic patterns. By preprocessing the CTU-13 dataset and implementing K-Means, DBSCAN, and Isolation Forest, I identify meaningful patterns and anomalies within network traffic. Integrating these models into a Flask-based web application highlights their practical applicability in real-world botnet detection scenarios. The clustering and anomaly detection results align well with known botnet behaviors, emphasizing the contribution of this project to botnet traffic analysis in cybersecurity.

References

- [1]. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19-31. <https://doi.org/10.1016/j.jnca.2015.11.016>
- [2]. Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: A Survey. *International Journal of Information Management*, 45, 289-307. <https://doi.org/10.1016/j.ijinfomgt.2018.08.006>
- [3]. Parwez, M. S., Rawat, D. B., & Garuba, M. (2017). Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics*, 13(4), 2058-2065. <https://doi.org/10.1109/TII.2017.2650206>
- [4]. Terzi, D. S., Terzi, R., & Sagioglu, S. (2017). Big data analytics for network anomaly detection from NetFlow data. *2017 International Conference on Computer Science and Engineering (UBMK)*, 592-597. <https://doi.org/10.1109/UBMK.2017.8093473>
- [5]. Stratosphere Research Laboratory. (2011). The CTU-13 dataset: A labeled dataset with botnet, normal, and background traffic. Retrieved from <https://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html>

[6]. Yuzzi, R. (2021, March 10). The importance of symmetrical vs asymmetrical internet connections [Video]. YouTube. [What is NetFlow and what can it show you?](#)

[7]. Onkar, A. (2021). *Network anomaly detection* [Data set]. Kaggle. <https://www.kaggle.com/datasets/anushonkar/network-anamoly-detection>