**Abstract**

Network anomaly detection is an essential part of network security, which means detection of intrusions or malicious activities from the traffic. New attacks often go unnoticed by conventional approaches to security when other approaches are applied, which is why using machine learning methods is necessary. In this paper, we provide a machine learning heuristic for identifying network anomalies via clustering methods including K-Means and DBSCAN. We chose the “Network Anomaly Detection” dataset available from Kaggle, which has network traffic data with labeled anomalies. The dataset was normalized, the features optimal for the model were selected, and transformed to improve the results. The system is designed to detect, in near real time, high level anomalous behavior and inform the cybersecurity analyst. The performance was measure through accuracy, recall and F1-score as we assessed the proficiency of the model in regard to classification of network events. The findings show that the proposed model can identify both existing and unrealized anomalies and can be a cost-effective and easy solution for cybersecurity experts. Future work may refine other approaches on how best to eliminate false positive results and improve on the scalability of this proposed system.

**Introduction**

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. In our case our motivation is to use machine learning in order to automate the process of looking for anomaly behaviours, which in turn asertain the early signs of suspicious activities that cybersecurity specialists can help to prevent.

This work employs a publicly available dataset called “Network Anomaly Detection” from Kaggle for a model construction, which is based on labeled network events. In the case of the input, structured records of network traffic will inculcate characteristics such as packet count, the given protocol, and connection time to perpetrate a given duration out of the time. In the case of output, a binary classification, the model will either label the network traffic as normal or anomalous. This project concerns one of the most pressing issues called AI in Cybersecurity, and focuses on the clustering technique related to anomaly detection within networks by utilizing K-Means and DBSCAN algorithms linked to Scikit-learn in Python. To this end, the proposed system aims for time sensitive parameters for real time detection and develop a comprehensive architecture for cybersecurity applications.

**Area of application, Dataset and/or Features**

Our project is in the cybersecurity field and particular concentrates on network anomaly detection using artificial intelligence approaches for improved network security and possible cyber threats. That is why network anomalies, including Distributed Denial of Service (DDoS) attacks, attempts to gain unauthorized access, and other unauthorized access attempts are so dangerous, as they cannot be detected using most of the traditional rule-based systems that are based on two-sigma thresholds or patterns of known attacks. Now that the nature of modern threats is toward continuously adapting and changing its methods of attack, AI can and should be used to model an always-evolving network of systems that can constantly search for disturbance patterns in the network traffic and provide alerts for cybersecurity analysts to analyze.

For constructing our anomaly detection system, the data set used was known as the “Network Anomaly Detection” and can be downloaded from Kaggle, contribution of Anushonkar. This dataset consists of thousands of records of the network traffic and each record is either normal or anomalous. This dataset enables devising of the machine learning based algorithm to differentiate between regular and malicious actors. For all experiments, we split the data into training set (70%), validation set (15%) and the testing set (15%). Such division let use train the model carefully, and check it on new data, which would enhance its reliability when using in practice.

Data Preprocessing

To prepare the dataset for model training and improve the model's ability to generalize, we performed several key preprocessing steps:

**Normalization:** Other parameters, including packet numbers and connection time, were converted to the same scale as a range of 0-1. This step is important in order to avoid having some feature dominating the modelDestination than is necessary and is beneficial to enhancing the functionality of the clustering algorithms.

**Feature Selection:** To help us hone in on the features for anomaly detection, we decided to look at which features were important and kept the data simple in order to train faster and more efficiently.

**Transformation:** All the data were preprocessed to bring some formats into the same standard and any instances of missing values dealt with in a way that caused no undue influence on the training of the models or the incorporation of errors.

Input Data and Features

The input data consists of a range of attributes describing each network connection. Key features used in our model include:

Packet Count: Describes the number of packets in a particular network connection value and used to gauge the amount of information transferred.

Protocol Type: Here, it denotes the type of protocol used, where different TYpes, means different forms of messages or attacks may be taken place.

Connection Duration: Records the amount of time spent on each of the networks. Abnormal periods of time may be defined as excessively long or too short.

Examples of records from the dataset:

| 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 |

The availability of the labeled data also allow the usage of supervised learning, which would allow the test and training of the model that will be required in the classification of the network events. The selected features enable us to quantify the extent of packet transmission and time related properties as well as qualify other properties such as protocol types that are critical in modeling network activities.

**Methods**