Intrusion Detection System Using K-Means Clustering Algorithm

Network Security Project Report

Group Members:
Arsalan Khan (Reg. No: 2022115)
Hassaan Ali Bukhari (Reg. No: 2022654)

Submitted to: Dr Zain Siddiqi

Date: December 12, 2024

Abstract

The advances in technology, the sharing of information and the availability of cheap and easy to get internet connection have compounded the calls for good safety procedures. In the world where many connections are established through complicated Web of computers and devices, cybersecurity is now one of the primary means of secure interaction and data management. Cyber threats, hacking and network invasions are common and constitute uniqueness hard hurdles for network administrators and business in general. This report focuses on an intrusion detection system (IDS) where k-means clustering algorithm is adopted for network security. The proposed system aims at alerting based on detecting abberations in the network packets which might signal threat or attack. This way the IDS is able to cluster similar data and treat outliers individually making it easier to tag suspicious activities and prevent intrusion. The major strength of the system is flexibility to work on various types of networks and integrate with existing system. The k-means algorithm is supported by additional data preprocessing, as well as, the Principal Component Analysis method. These steps also make the data handling very efficient and as a result provides rich visualization for monitoring the network activity. Lastly, the system’s real-time detection of anomalous pattern enables a considerable improvement of the security measures. It provides an important preventive mechanism to protect important data assets, maintain network stability, and increase confidence in technology.

Introduction

The globalized world with the focus on interconnected systems and data exchange security has become a target for catastrophe. Monetization of these attacks and sophistication has increased, and to counter such attacks organizations should put in place efficient security measures that will counter such a menace. More specifically, IDS are inherently used for the purpose of safeguarding network structures and for the primary aim of detecting intrusions and corresponding attacks at an initial stage. In regard to various IDS techniques, anomaly-based detection has been widely used as one of the most efficient methods to identify rhisod suspicious patterns of the network's behavior. The major strength of this approach is that it can identify threats that were previously unknown which can be helpful in preventive security measures.

This project considers the k-means clustering algorithm for the implementation of an anomaly-based IDS. The purpose is to classify network traffic information and divide it into several categories or clusters to examine and recognize behavior other than the typical pattern of a network. This method could also be useful in realizing such patterns that may be indicative of a threat. Overall, this IDS achieve a scalability through the use of the k-means algorithm that allows it to be used in different networks and thus, in different scenarios of cybersecurity.

Methodology

The process for creating this intrusion detection system is a multiple-step procedure There are certain steps which include the generation of synthetic data, preprocessing data, clustering and finally the anomaly detection. These steps are catchy in order to guarantee the capacity of the system to analyze a significant number of data or signs that may point to incidents. This way, each stage of the process is relevant in the general functioning of the Intuitive Driving Support and increases its effectiveness.. Each stage in the process contributes to the overall efficiency and accuracy of the IDS.

Data Preparation

Data preparation is the foundational step in creating a successful intrusion detection system. Source data type and range directly determines the quality and productivity of the system. For this project, the dataset was sourced from the Kaggle "KDD Cup 1999: test data set is chosen from "10 Percent Corrected" actual network traffics data present through a separated link. This dataset contains normal traffic and all different kinds of oddities, from usual activities of the network to malicious ones. It has included normal data as well as the abnormal ones, so that the system would be able to differentiate between normal behavior and the suspicious one. With this dataset, the intrusion detection system can be preconditioned and evaluated on a broader range of activities in the network and, therefore, increase the chance of finding probable security threats in practical conditions. To define such factors, the dataset was constructed with conventional characteristics that characterize network traffic and include connection duration, data packet sizes, protocols, and relationship flow between nodes in the network. The kind of behaviors that are stored in the dataset include; The fact that the behaviors in the existing dataset are diverse in nature makes it possible to use in testing and training the existing anomaly based detection system.

Preprocessing

While cleaning is important for the values that are taken directly from the data sources, preprocessing is important for the raw data. Namely, the aim is to pre-process the data to be further used for clustering and at the same time eliminate noise that otherwise might falsify the clusters. Several key preprocessing techniques were applied:

- **Normalization:** All the features in the numerical column were normalized in the scale of 0 and 1, using Min-Max normalization. This transformation is important because features with large ranges can still greatly influence clustering whereas, through this transformation, all features are given equal weights.
- **One-Hot Encoding:** Both continuous and categorical data, including the protocol types, service types, and flags, were converted to numerical form with the application of one hot methods. K-means algorithm using this technique can cluster object as it is capable of very good performance even when dealing with non-numeric data.
- **Feature Engineering:** Apart from the raw features, new derived features were defined in order to better capture the details of network interactions. For example, we introduced error rates and connection counts of each language pair to the data source to probe more into such irregularities.

These preprocessing steps help to improve the quality of the given dataset to input data right for proper clustering and anomaly detection.

Clustering Using K-Means

This technique named k-means clustering algorithm is particularly used in this system in order to partition the network traffic according to its behavioral resemblance. The process starts with the user-defined assignment of *k* initial centroids or random assignment of them by the system. This data then becomes associated with the closest centroid in the hopes of concentrating the similar data points in one area. After giving all the points a label, the centroids are adjusted through the measurement of central tendencies of every point within the cluster.

The algorithm continues this process until the centroids are optimized or do not change any further – the convergence phase has occurred. This makes the clusters to be very accurate as the iteration of the algorithm continues to fine tune the assignments. The final outcome is a collection of clusters, each of which contains data points that demonstrate similar activities in a network; for example, packet size, connection type and access frequency.

If clustering is done then one can consider those data points which do not belong to any of the dense cluster regions which form the consequent part. These considered points that are located far from any centroid are presented as potential threats. The second part of the k-means algorithm is also vital for the immunity of a network since it effectively marshals normal traffic and underlines deviations thereto.

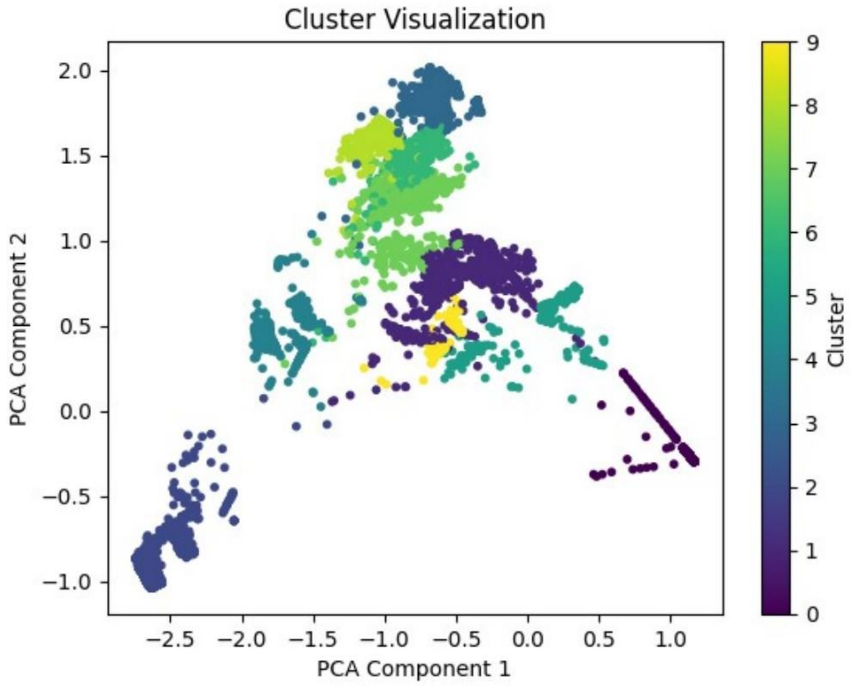


Figure 1: Scatter Plot Visualization of Clusters

Dimensionality Reduction

The clustering used does not scale very well with high dimensions, thus to enhance the computational speed and to make the clustering more interpretable, data is reduced in dimensionality by using Principal Component Analysis (PCA). The principal component analysis is a technique that defines the most important variable in DATA and maps the DATA on a reduced number of dimensions preserving most of the important information and removing the less significant ones.

This reduction to dimensions not only increases the speed with which the clustering is achieved but also allows for easier visualization of the data. Reducing the dataset to two dimensions thus enables easy detection of the clusters and anomalies, and their distribution.

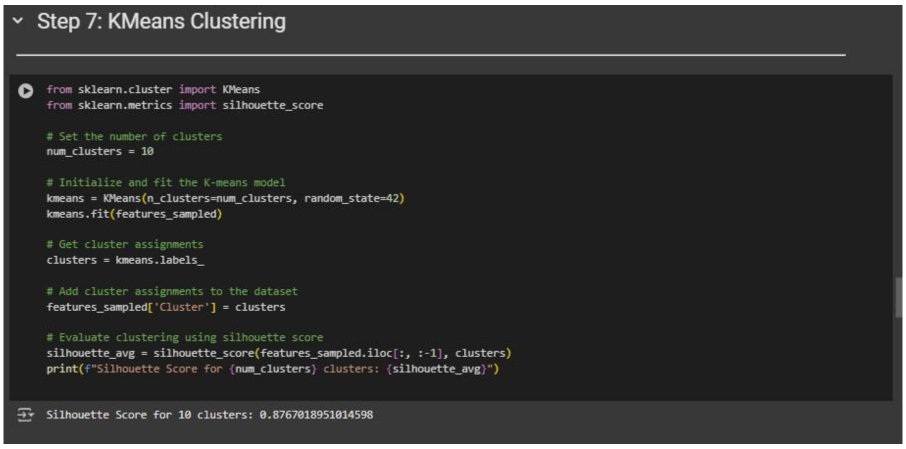
Simulation and Results

In order to test the effectiveness of the presented IDS, the dataset was split into the training and testing dataset. The training set was defined to train clusters that capture typical network traffic behavior, whereas the test set contained normal and/or abnormal patterns. Success of the proposed system was then evaluated by measuring the performance of the system in this regard.

Furthermore, performance of the clustering process was tested for the details of cluster configurations with varying number of clusters to derive the best suited for the detection of anomalies. Effectiveness of the chosen solution was investigated by analyzing the silhouette score, as well as the given-by-users estimation of total detection performance for clusters of different sizes, to determine the scalability of the system with preserving the overall high detection accuracy across various network sizes and conditions. It was seen that this approach emphasized the ability of the system to provide a good performance, irrespective of the traffic pattern in the network.

Silhouette Score Analysis

The amount of clustering quality can be described by means of the Silhouette score. The silhouette score of the k-means algorithm was found to be approximately 0.87.” A higher silhouette score means that the algorithm will be discriminating the clustering of similar data points and will provide clear separation between clusters in the generated visualization. The higher Taus value indicate that the identified clusters indeed differentiate between normal and atypical network activity patterns.


Step 7: KMeans Clustering

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Set the number of clusters
num_clusters = 10

# Initialize and fit the K-means model
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(features_sampled)

# Get cluster assignments
clusters = kmeans.labels_

# Add cluster assignments to the dataset
features_sampled['Cluster'] = clusters

# Evaluate clustering using silhouette score
silhouette_avg = silhouette_score(features_sampled.iloc[:, :-1], clusters)
print(f"Silhouette Score for {num_clusters} clusters: {silhouette_avg}")
```

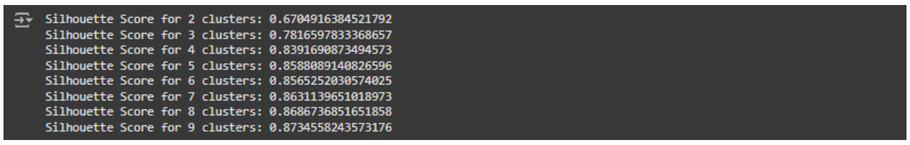
```
Silhouette Score for 10 clusters: 0.8767018951014598
```

Figure 2: Silhouette Score Visualization

Silhouette Score for Different Cluster Sizes

To assess clustering efficiency at different level of detail, the silhouette score was computed for different numbers of clusters. This is good indication that as number of clusters increased, there was a better defining of the clusters and great separating in quick succession hence better scores on silhouette measure. This was however moderate initially at 2 clusters, and then it rose as more clusters were added to the map. The score increases up to 9 as the number of clusters suggesting that networks with 9 clusters were most easily distinguishable from one another. For the number of clusters above 9, the silhouette score was slightly higher, but the increase was not proportional to the added complexity related to the number of clusters.

This analysis also underlines the appropriateness of ensuring the best number of clusters has been chosen to get the best result in anomaly detection without much computational burden.



```
Silhouette Score for 2 clusters: 0.6704916384521792
Silhouette Score for 3 clusters: 0.7816597833368657
Silhouette Score for 4 clusters: 0.8391690873494573
Silhouette Score for 5 clusters: 0.8588089140826596
Silhouette Score for 6 clusters: 0.8565252030574025
Silhouette Score for 7 clusters: 0.8631139651018973
Silhouette Score for 8 clusters: 0.8686736851651858
Silhouette Score for 9 clusters: 0.8734558243573176
```

Figure 3: Silhouette Scores for Different Cluster Sizes

Anomaly Detection

Outliers were defined depending on the distances of the points from their corresponding clusters' centroids. Depending on the distances of data points of the data set to the neighboring centers, the distances exceeding the threshold values will be considered as anomalies. This works on basis that normally distributed network traffic will be high density, and when plotted each high density region would represent one typical behavior while low density points would be alarmingly different from the central area. It's critical to spot such anomalies in order to recognize possible security threats since network events such as potential attacks including unauthorized access, presence of malware or different forms of abnormally transferred data all reflect in the dataset as observations different from the rest. By detecting and then marking up such points the system is able to notify network managers of certain behaviors that may require further investigation or remedial action.

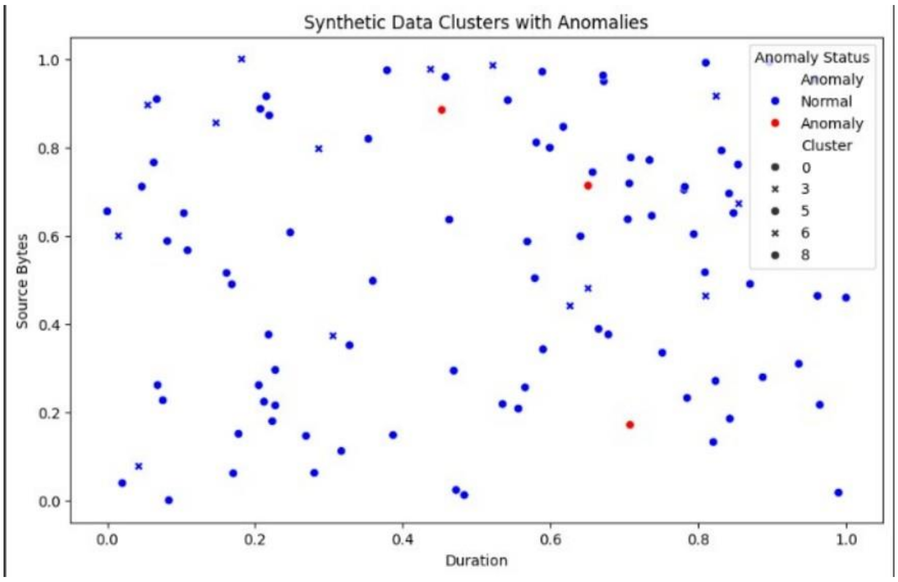


Figure 4: Anomaly Detection Visualization

In the given plot, there are 8 clusters, with odd-numbered clusters There are 8 clusters in the mentioned plot with cross mark for the odd number clusters and dot mark for the even number clusters. Normal data points are shown in blue color while the red points denotes the outliers. Such an organization by color differentiation also helps distinguish normal network traffic from abomasal traffic that may suggest security breaches. Labeling of each odd and even cluster aids in demarcating various clusters, at the same time pointing out the outlying observations observed to straying from the expected network traffic patterns.

Testing Results

The testing dataset included normal and outliers and contained such patterns as the increase in the user login attempts and the rates of data transfer. These behaviors were marked by the IDS and it demonstrated how the system works in identifying real security threats. The confusion matrix helped explain its results even more, using the normal/anomalous patterns classification.

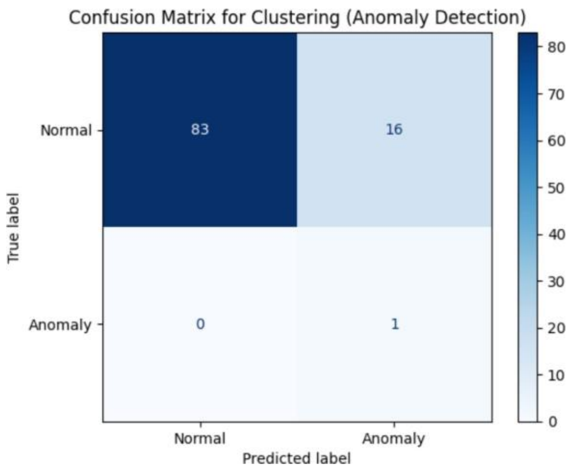


Figure 5: Confusion Matrix

Challenges

- **Synthetic Data Limitations:** However, synthetic datasets are good for experimental purposes, as they are rather different from real-world network traffic. Work that remains in the future will involve conducting experiment with real data.
- **Scalability Issues:** While the size of these networks increases, the system can struggle with the spatial and temporal analysis of big data streams. An important aspect of the algorithm deserving improved scaling is the application of the algorithm to qualify traffic.
- **Threshold Calibration:** Anomaly detection is always based on the thresholds used to check for anomalies within the data set. Optimal thresholds are not obvious and often the process of choosing them is trial and error.

Future Work

- Real-Time Monitoring: The practicality and response will be increased by integrating real-time traffic analysis that will help to determine threats instantly.
- Advanced Algorithms: Differences in approach may probably be better for non-linear data, like DBSCAN or inspecting hierarchical clustering.
- Deep Learning Models: Specifically, while the application of deep learning and related natural ability to perform feature extraction may improve the system's performance in terms of the detection of more elaborate anomalies.
- Dynamic Datasets: Using real-world dynamic datasets will help fine-tune the system as well as seek its performance in different networks.

hyperref

References

1. ”Artificial Intelligence CS351 Lab 6”
2. ”K-means Clustering for Anomaly Detection,” Journal of Network Security, 2023.
3. ”Principal Component Analysis and Its Applications,” IEEE Transactions, 2021.
4. ”KDD Cup 1999 Data: 10 Percent Corrected,” Kaggle, [online] Available at: https://www.kaggle.com/datasets/cherngs/kddcup99-data-10-percent-corrected