Anomaly Detection in Computer Networks using Machine Learning Classifiers

Vatsalkumar Patel*, Shreya Karia†, Digant Patel‡ and Samarth Chauhan§

***Abstract*—Our project aims to train supervised learning models to detect anomalies in computer networks to identify possible security threats. We selected an appropriate dataset and cleaned and pre-processed the data. We did exploratory data analysis to choose appropriate features to train our classifiers. We reduced the dimension of our dataset using PCA and trained a kNN and a Random Forest classifier to select the better model.**

***Index Terms*—Anomaly detection in computer networks, PCA, kNN, Random Forest**

I. Introduction

The process of detecting anomalies is crucial in network defense as it provides security administrators with advanced warnings of potentially harmful actions such as attacks, malware, and intrusions. In computer networks, anomalies have dynamic nature, and thus they cannot be identified easily with any rule-based approach. Machine learning techniques are fit for detecting such threats. We have selected an IDS dataset, namely CICIDS2017[1], which covers all the eleven necessary criteria with common updated attacks such as DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port scan, and Botnet. However, this report only focuses on four types of attacks: DoS Hulk, DDoS, PortScan, and DoS GoldenEye. We implement various machine learning classifiers on our dataset and analyze their performance to classify the attacks.

II. Literature Survey

There have been several datasets and papers on intrusion detection over the years. The most commonly used dataset is the KDD dataset, but these datasets are extremely outdated. Our dataset is CICIDS2017, which considers the existing datasets' limitations while constructing its own. Extensive research has already been carried out on the same dataset. A paper[2] analyzes the generated dataset to select the best feature sets to detect different attacks and also executes seven common machine learning algorithms to evaluate the dataset. The paper finds the best short feature set to detect each attack family using the RandomForestRegressor algorithm. Afterward, it examines the performance and accuracy of the selected features with seven common machine learning algorithms: kNN, RF, ID3, Adaboost, MLP, Naiye-Bayes, and QDA. Finally, it compares the quality of the generated dataset by searching for common mistakes and criticisms of other synthetically created datasets.

III. Implementation

A. Data Cleaning and Pre-processing

Data cleaning is a critical step in data preprocessing, where data quality issues are addressed to improve the accuracy and reliability of data analysis. There were 308381 duplicate values and 353 null values in the dataset. Therefore we removed unnecessary duplicate values and null values from the data. Then we reduced the classes to focus on the 4 classes and Benign data. We removed data for other classes as the number of rows for those classes was less than 10,000 which could have caused problems while balancing the dataset. Memory usage was then optimized by downcasting.

B. Exploratory Data Analysis

Highly correlated features may affect the performance and interpretability of a machine learning model. Removing highly correlated features can have a positive impact on the performance of an algorithm. We achieved this objective by finding the features that have a correlation of more than 0.85 and then removing them.

C. Balancing Dataset

We used undersampling[3] to balance the dataset. It is important to balance the data before training the model to make sure there is no bias. We undersampled data for all classes to have 10,286 rows. Undersampling was a better choice compared to oversampling because in oversampling, we would have to create artificial data to have an equal number of rows of each class which would make the dataset less reliable. Also, the dataset is large enough to have enough rows for each class left after undersampling.

D. Dimensionality Reduction

Principal Component Analysis (PCA) is a dimensionality reduction technique wherein the most important information of the dataset is retained. PCA works by finding a set of variables that are linear combinations of the original features such that they capture as much variance in the data as possible. PCA is the suitable option in this case as the features are linearly correlated and of high dimensionality.

E. Model Training

We split the data into two parts, the training set (70%) and the test set (30%). We used two models, kNN and Random Forest Classification, on the data.

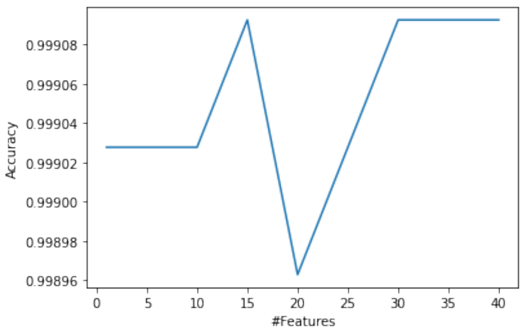
1) k-Nearest Neighbour: In KNN classification, the algorithm tries to predict the class of a new instance based on the majority class of its k-nearest neighbors in the training set. The distance between instances can be measured using the Minkowski distance.

2) Random Forest: Random Forest works by randomly selecting a subset of features at each decision tree split, which helps reduce the model's variance and overfitting. Each decision tree in the forest is built independently using a different subset of the training data, which helps increase the model's diversity and accuracy. The final prediction of the Random Forest algorithm is made by aggregating the predictions of all the decision trees in the forest. For classification problems, the most common class predicted by the decision trees is selected as the final prediction.

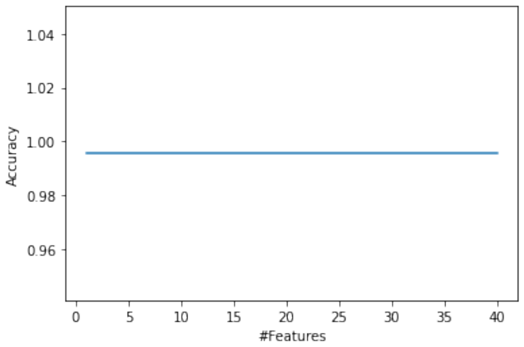
We used PCA to transform the data to 11 dimensions and trained the model using RF and kNN each time.

IV. Results

The line chart for the number of features vs. model accuracy on the test data for Random Forest algorithm:



The line chart for the number of features vs. model accuracy on the test data for kNN algorithm:



The accuracy of k-Nearest Neighbors (kNN) does not change with different numbers of features after performing Principal Component Analysis (PCA); it could be due to the following reason:

- The first few principal components may capture most of the variability in the data: PCA aims to reduce the dimensionality of the data while retaining most of the information. If the first few principal components already capture most of the variability in the data, reducing the number of features may not significantly impact the accuracy of kNN.
- kNN is a non-parametric algorithm that works by finding the k nearest neighbors to a given point. It does not make any assumptions about the data distribution, and the number of features may not significantly impact the algorithm's accuracy.

V. Conclusion

After using both models and analyzing the results, one can infer that the top few features capture the most data variability. Because of that reason, there is not much difference in the accuracy of the model when changing the dimension of features from 1 to 40 in the case of random forest. In the case of kNN, no change is observed in accuracy when the dimensionality is varied. RF is a very efficient algorithm for handling high-dimensional data compared to kNN, which can be computationally expensive in the case of large datasets as it requires calculating the distance between each pair of data points. RF is also more accurate as it chooses the most informative features to make decisions, whereas kNN considers all features equally important, which may lead to decreased performance when some features may be irrelevant.

References

[1] CICDataset, "CICIDS2017," Kaggle, 03-Jan-2020. [Online]. Available: https://www.kaggle.com/datasets/cicdataset/cicids2017. [Accessed: 11-Mar-2023].

[2] "Toward generating a new intrusion detection dataset and intrusion TRAFC ..." [Online]. Available: https://rb.gy/9lirab. [Accessed: 11-Mar-2023].

[3] "A review on imbalanced data handling using undersampling and oversampling technique," International Journal of Recent Trends in Engineering and Research, vol. 3, no. 4, pp. 444–449, 2017.