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 ANOVA. The processed data was used to train a kNN, a Random Forest, a Gaussian Naive Bayes, and a Quadratic Discriminant Analysis classifier to select the better model.**

***Index Terms*—Anomaly detection in computer networks, PCA, ANOVA, kNN, Random Forest, Gaussian Naive Bayes, Quadratic Discriminant Analysis**

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 The most commonly used dataset is the KDD dataset, but these datasets are extremely outdated.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[**?**] 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, Naive-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[2] 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. We did not use Synthetic Minority Oversampling Technique (SMOTE) because it creates duplicate data for the minority classes and this would make the data unreliable because the difference in number of rows for each class was too high.

D. Dimensionality Reduction

1) PCA: 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.

2) ANOVA: Analysis of variance (ANOVA) is one of the statistical tests that is used to examine the correlation between the features. The study of the correlation of the features is made through the F-test for ANOVA. Features will be ranked in accordance with the F-statistic's results. The top set of features from the data can be chosen based on the rankings of the scores.

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.

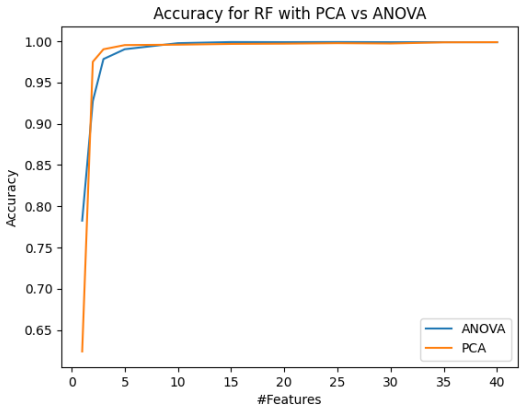
3) Gaussian Naive Bayes: Gaussian Naive Bayes is a type of Naive Bayes Algorithm that adds to the assumption that all features are independent of each other and also assumes that the features follow a Gaussian distribution. It calculates the class conditional probabilities for all the classes and chooses and classifies a new data point in the class which has the highest value of the class conditional probability.

4) Quadratic Discriminant Analysis: Quadratic Discriminant Analysis (QDA) is a supervised machine learning algorithm that assumes that the data points follow a multivariate Gaussian Distribution. When given an input, it calculates the probability distribution for the input given to each class separately. The obtained values are used to calculate the posterior probabilities. This leads to a non-linear decision boundary, making the model flexible.

IV. Results

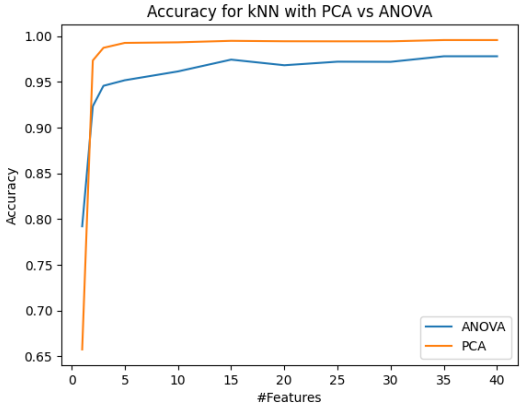
A.

Random forest algorithm is the slowest but highly accurate model. When using PCA or ANOVA, the accuracy is close to 99% for more than 5 features. For less than 5 features, the accuracy reduces significantly. The line chart for the number of features vs. model accuracy on the test data for the Random Forest algorithm with ANOVA and PCA is as follows:



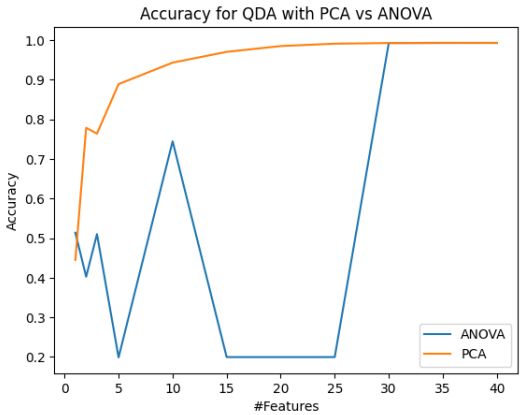
B.

kNN is as accurate as Random Forest but a faster algorithm. A similar trend is seen in the plot for accuracy vs #features for kNN as in RF. The line chart for the number of features vs. model accuracy on the test data for the kNN algorithm with ANOVA and PCA is as follows:



C.

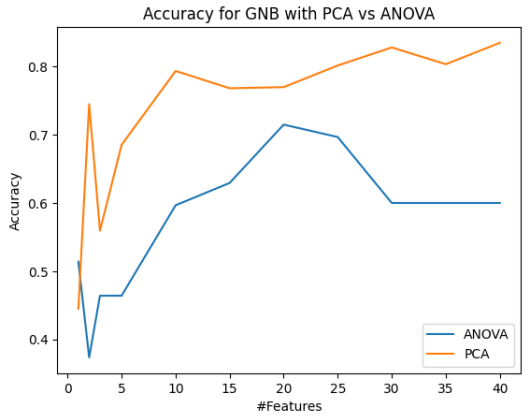
Quadratic Discriminant Analysis performs better with PCA than with ANOVA and is comparatively more accurate than Gaussian Naive Bayes. The line chart for the number of features vs. model accuracy on the test data for the Quadratic Discriminant Analysis algorithm with ANOVA and PCA is as follows:



D.

Gaussian Naive Bayes has the least accuracy of the four models implemented.

The line chart for the number of features vs. model accuracy on the test data for the Gaussian Naive Bayes algorithm with ANOVA and PCA is as follows:



E. QDA vs GNB

While both GNB and QDA are based on the assumption that the features have Gaussian Distribution, one performs extremely well as compared to another. QDA allows for different covariance structures for each class, whereas GNB assumes that all classes share the same covariance structure. QDA can capture the variability more accurately, resulting in better classification performance.

F. PCA vs ANOVA

PCA performs better than ANOVA because, in ANOVA, top features are just selected, whereas, in PCA, features are transformed to a new basis. PCA can therefore capture most information from the data while reducing dimensionality. It even reduces the risk of overfitting. Another reason for the lower performance of ANOVA is that it can only capture linear relationships, while PCA also captures non-linear relationships.

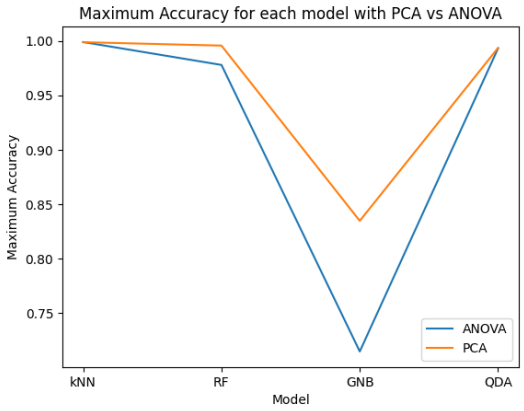
V. Conclusion

- After using the four 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 10 to 40. The accuracy does reduce after reducing the dimension of features to less than 10.

- Increasing the number of neighbors in kNN above 3 leads to overfitting, and the accuracy reduces.

- PCA is better than ANOVA for feature selection as the accuracy with ANOVA is lower compared to the accuracy with PCA.

- RF is a very efficient algorithm for handling high-dimensional data compared to kNN, QDA, and Naive Bayes despite of its high runtime. RF is 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.
  The line chart for the model vs. accuracy on the test data is given below for all four implemented algorithms:



Model Accuracy				
	kNN	RF	GNB	QDA
PCA	0.9987	0.9956	0.8347	0.993
ANOVA	0.9988	0.9778	0.7148	0.9931

- From the plots, we can infer that PCA performs better than ANOVA for all four algorithms, and kNN with PCA is the best choice for Intrusion Detection both in terms of accuracy and efficiency.
- Different classes of intrusions would have quite different characteristics; in such a case, kNN would work best when new activity has to be classified, and the nearest neighbors, having similar characteristics, would be the best metrics.

References

[1] V. Patel, S. Karia, D. Patel, and S. Chauhan, "Anomaly_Detection" GitHub [Online]. Available: https://github.com/Samarth1302/Anomaly_Detection.git

[2] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (n.d.). Toward generating a new intrusion detection dataset and intrusion TRAFC ... Retrieved February 13, 2023, from https://rb.gy/9lirab.

[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.

[4] V. M. Deolindo, B. L. Dalmazo, M. V. B. da Silva, L. R. B. de Oliveira, A. de B. Silva, L. Z. Granville, L. P. Gaspary, and J. C. Nobre, "Using quadratic discriminant analysis by intrusion detection systems for port scan and slowloris attack classification." Retrieved 31-Mar-2023.

[5] Yoshifumimiya, "Feature selection using ANOVA," Kaggle, 18-Oct-2022. [Online]. Retrieved 21-Mar-2023 from https://www.kaggle.com/code/yoshifumimiya/feature-selection-using-anova.