Exploring Cybersecurity Datasets for Data Mining and Machine Learning

CICIDS2017 Dataset

K-Nearest Neighbors

MTH 410 – 20/05/2024

GROUP- 11

Aslı Gizem Ulusoy

Berkay Alpay

Harun Emre Yıldız

https://github.com/asliulusoy/MTH-410-Midterm-Project

Istanbul Bilgi University

1. Introduction

At the present time, the increase in the number of cyber security threats opens a road to developing efficient and mandatory defense mechanisms. In this context, defining these threats and preventing them will be the most significant point. [1] In this work, it will be seen that by using the CIC-IDS 2017 dataset, various types of attacks will be detected by applying some machine learning algorithms and data mining techniques. This dataset contains some attack scenarios and also simulates benign web traffic in detail.

The primary objective of this report is to demonstrate how data preprocessing, exploratory data analysis, and classification techniques can be employed to detect cyber threats and develop a predictive model. The report will cover the entire process, from initial data preprocessing to the final application of a chosen data mining technique, with the goal of effectively identifying and mitigating cyber threats.

1.1 About CICIDS2017 Dataset

In today's digital era, cybersecurity has emerged as a critical challenge. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) play vital roles in defending against the escalating threat of web attacks. However, the efficacy of these defense mechanisms relies heavily on robust and thoroughly tested datasets. Addressing this need, the Canadian Institute for Cybersecurity (CIC) has developed the CIC-IDS 2017 dataset, which serves as a comprehensive resource for cybersecurity research. This dataset, collected from 03.07.2023 at 9 AM to Friday at 5 PM, mirrors real-life web traffic by incorporating benign and newsworthy attack types. The primary goal of the CIC-IDS 2017 dataset is to facilitate the detection of cyber intrusions and foster the development of effective solutions for the shortcomings in Intrusion Detection and Prevention Systems. [2]

2. Data Preprocessing

In this part of the report, the procedure of cleaning and getting the dataset ready for analysis will be covered. Missing values are identified, feature selection is performed and if necessary, data transformation is carried out. In this work these steps are applied in order, first loading the dataset and listing the files to ensure to be in correct path. After importing the essential libraries, in order to comprehend the extent of the dataset the amount of the rows and columns are determined in Figure 1.

After merging, it was discovered that there are duplicate header fields throughout the CICIDS 2017 CSVs. Duplicate rows are identified and removed to avoid redundant information that could skew the analysis in Figure 2.

Data samples were excluded if they contained duplicate records or implausible features, following the approach outlined by Reis et al. [3] and Rosay et al. [4]. They argued that such data cleansing is essential for ensuring accuracy in real-world applications.

```
# Print the number of rows and columns in the data
rows, cols = data.shape
print("Number of rows:", rows)
print("Number of columns:", cols)
```

```
Number of rows: 2830743
Number of columns: 79
```

Figure 1: Determining the number of rows and columns.

```
# Check the number of duplicated values
duplicates = data[data.duplicated()]
print("Number of duplicates:", len(duplicates))
```

```
Number of duplicates: 308381
```

Figure 2: Obtaining the number of duplicates.

Afterwards, preceding and following whitespace are removed from column names to guarantee naming convention compatibility. Columns with missing values are identified (Figure 3), and infinite values are converted to NaN before dropping any rows containing NaN values to maintain data integrity (Figure 4).

```
# Identify columns that contain missing values.
data.columns[data.isnull().any()]
```

```
Index(['Flow Bytes/s'], dtype='object')
```

Figure 3: Identifying the colums that contain missing values.

Moreover, columns where all values are the same are dropped because they do not provide any useful information for the analysis (Figure 4). Labels are separated from the features to prepare for model training. By performing these steps at data preprocessing part, the dataset would become more organized and cleaned from unnecessary information to be more suitable to apply exploratory data analysis and machine learning applications. Important parts for the data preprocessing can be found at the below visually. (Figure 4)

```
# Finding infinite values in the data and converting them to NaN
data = data.replace([np.inf, -np.inf], np.nan)
# Dropping NaN values
data.dropna(inplace=True)
```

```
# Drop columns where all values are the same (constant columns)
for col in data.columns:
    if data[col].nunique() == 1:
        data.drop(col, inplace=True, axis=1)
```

Figure 4: Converting and dropping values

```
data.shape
```

```
(2520798, 71)
```

```
data.reset_index(drop=True, inplace=True)
```

Figure 5: Final shape of the data, resetting the index

3. Exploratory Data Analysis

Exploratory data analysis has a significant role in processing and interpreting the dataset. It uses various visualization techniques together with statistics. The aim of the EDA is to gain insights into the data, detect possible anomalies and identify relevant features for the modeling stage.

3.1. Correlation Heatmap:

A correlation heatmap is used to visualize the relationships between features in a dataset. Correlation coefficients are calculated to identify positive and negative relationships between features and displayed as a heatmap. Coefficients of the correlation maps are represented between –1 and +1. +1 is full positive and as one attribute increases, the other increases at the same rate. 0 is no correlation at all and –1 is full negative correlation, as one attribute increases, the other one decreases at the same rate. The heatmap (Figure 5) is below:

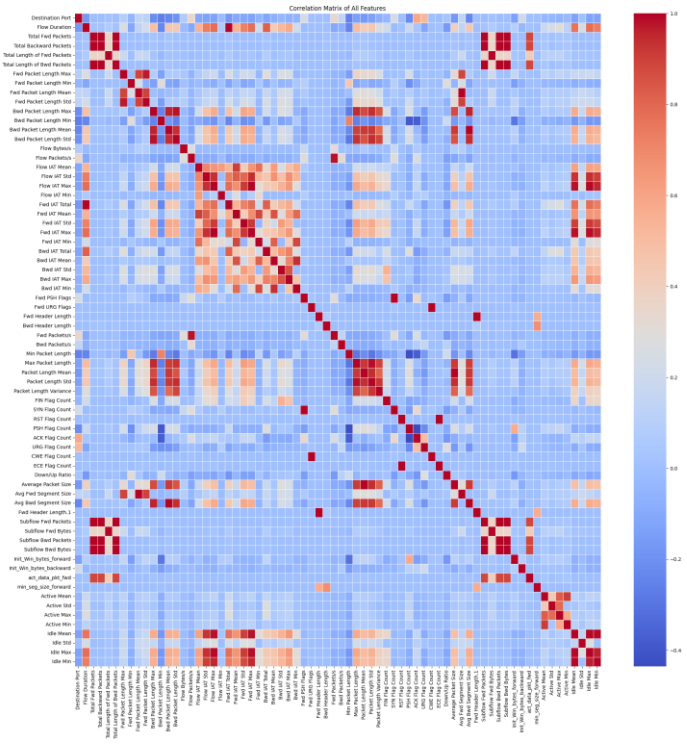

Figure 5: Heatmap plot.

3.2 Implications of the Correlation Map:

High Positive Correlation:

“Bwd Packet Length Max and Bwd Packet Length Mean”: There is a high positive correlation between these two attributes. This indicates that as the maximum value of backpack length increases, the average value of backpack length also increases. These two properties reflect the general behavior of the backpack length. Using one characteristic might be enough because they represent similar information.

“Fwd Packet Length Max and Fwd Packet Length Mean”: There is a high positive correlation between these two characteristics. As the maximum value of forward packet length increases, the average value of forward packet length also increases. Forward packet length characteristics reflect the general behavior of forward packets. Using one characteristic might be enough because it can increase the simplicity of the model.

High Negative Correlation:

“Fwd Packet Length Max and Bwd Packet Length Max”: There is a significant negative correlation between these two features. As the forward packet length maximum value increases, the backward packet length maximum value decreases. This would suggest that the direction of network traffic affects the length of packets. A specific kind of assault might, for instance, generate smaller packets in the opposite way while sending larger packets in the forward route. These connections may be essential for classifying and detecting attacks.

“Flow Duration and Total Fwd Packets": There may be a weak or negative correlation between flow time and the total number of forward packets. This indicates that as the flow time increases, the total number of forward packets may decrease. Long duration flows may generally contain fewer large packets, while short duration flows may contain a larger number of small packets. This relationship can be helpful in understanding the nature of network traffic and distinguishing between different types of traffic.

Zero Correlation:

Zero correlation indicates that there is no relationship between two attributes. The value of one attribute does not affect the value of the other attribute.

For instance, between total bwd packets and flow duration there may be a weak or near zero correlation between these characteristics. In Addition, between fwd header length and bwd header length has weak correlations. Individual features can improve overall performance by providing different information in the model.

These inferences help us understand the overall structure of the dataset and make more informed decisions during the modeling phase. The correlation map provides a solid foundation for visualizing important relationships in the dataset and for the application of data mining techniques.

3.3. Scatter Plots:

Attack Type Distribution: In this plot, the distribution of the attack types has been visualized. This helps us to understand which types of attacks are, how common they are in the dataset. This bar chart shows only the number of attack types, excluding data points labeled 'BENIGN'.

According to this graph, the most common attack types in the dataset are 'DoS Hulk', 'DDoS', and 'PortScan'. These attack types have a higher number of instances compared to the others. For example, the 'DoS Hulk' attack is one of the most common attack types in the dataset and is represented in a fairly high number. Similarly, 'DDoS' and 'PortScan' attacks also have a significant presence in the dataset.

On the other hand, some attack types are represented by fewer instances. Attack types such as 'Infiltration' and 'Heartbleed' are less common in the dataset. The lower number of such attacks may make it difficult for the model to classify them correctly and may require special attention in the modeling process.

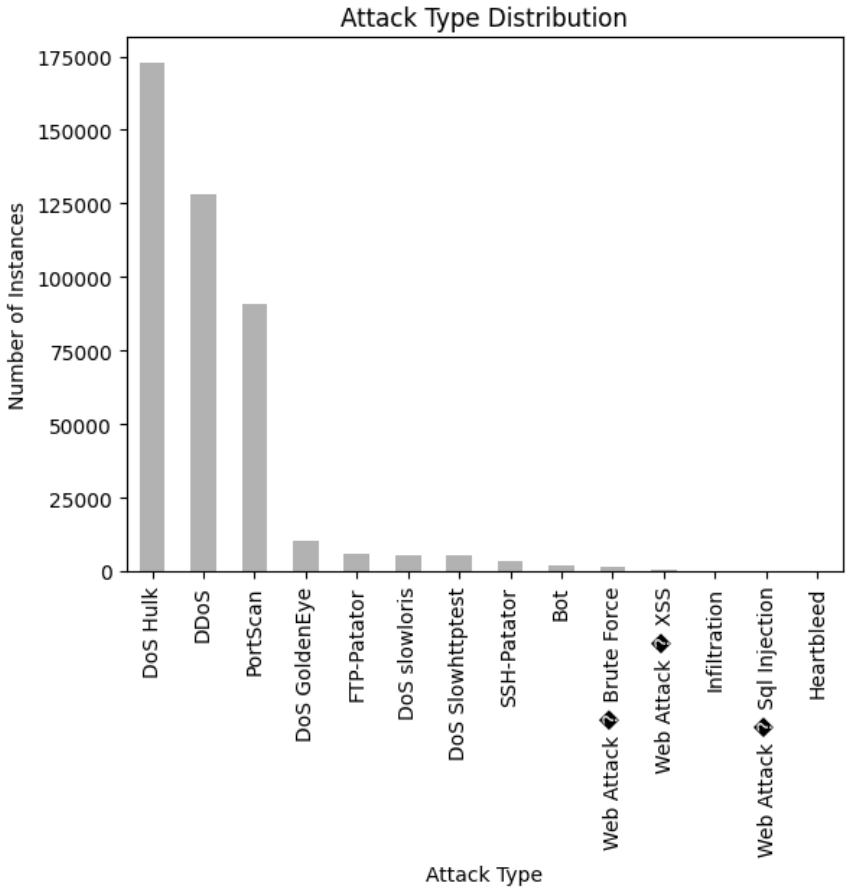

Figure 6: Attack Type Distribution

Plot of Flow Packets and Fwd Packets:

As it seems above with the plot, Normal traffic ('BENIGN') is represented by low 'Flow Packets/s' and 'Fwd Packets/s' values in the lower left corner of the graph. This indicates that normal network traffic has lower packet rates. The types of attacks are grouped in various locations, usually with higher values than normal traffic. DDoS attacks in particular are represented by a large number of points along the 'Flow Packets/s' axis that reach high values, which is an indicator of intensive packet flow. The areas where the attack types (for example, DDoS, PortScan, Bot) are concentrated reflect the Although attack types like "Web Attack & SQL Injection" and "Web Attack & XSS" may have lower packet rates, their content is more significant than their quantity because these attacks are more tailored and target oriented. characteristic features of such attacks. While DDoS attacks and Bot activities are associated with high 'Flow Packets/s' values, they show similar trends on the 'Fwd Packets/s' scale.

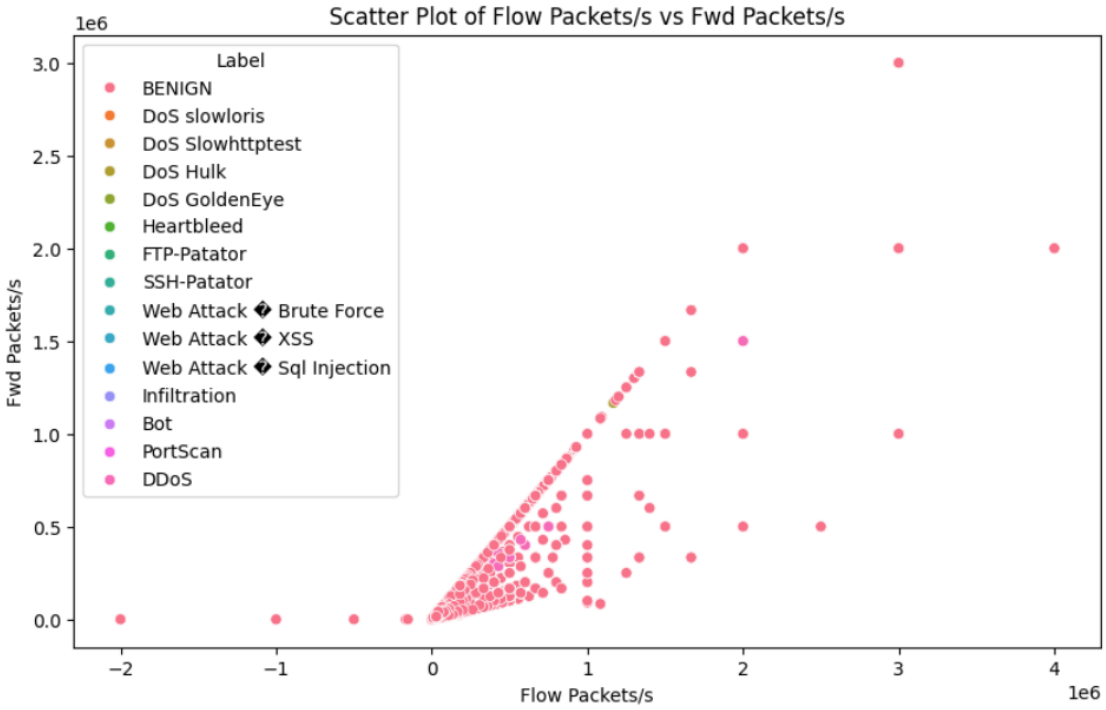

Figure 7: Scatter Plot of Flow Packets/s vs Fwd Packets/s

Feature Importance Analysis with Random Forest Classifier:

The analysis was carried out on a 10% random sample of the data set. n_estimators=50, max_depth=10, random_state=42 and n_jobs=-1 were used as model parameters. These parameters are critical factors that affect the learning process of the model and determine the prediction success. It is seen that characteristics such as Bwd Packet Length Std, Bwd Packet Length Mean and Packet Length Variation are of the highest importance. These features are particularly effective in capturing observed anomalies in network traffic during attacks. Especially high-scoring features should be considered as a priority when monitoring network traffic. These features increase the capacity to detect potential attacks and security threats at an early stage, which allows security teams to respond faster and more effectively.

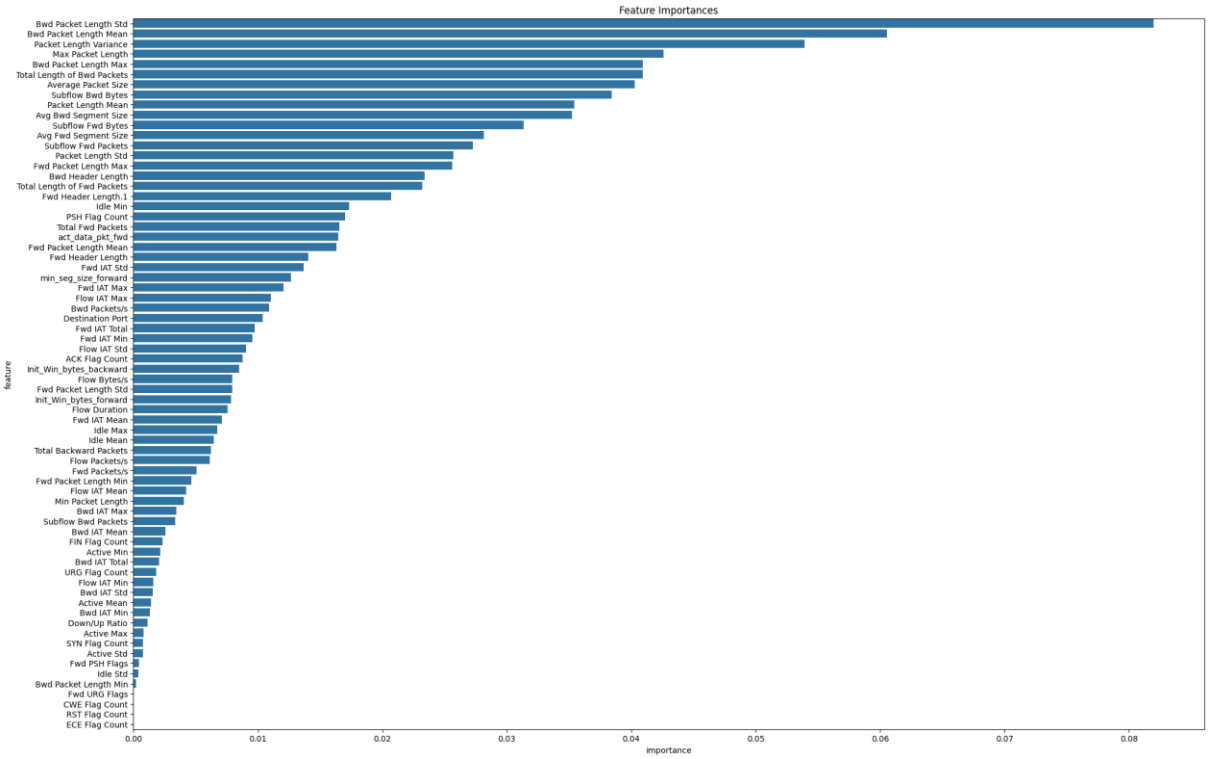

Figure 8: Plot for Feature Importances

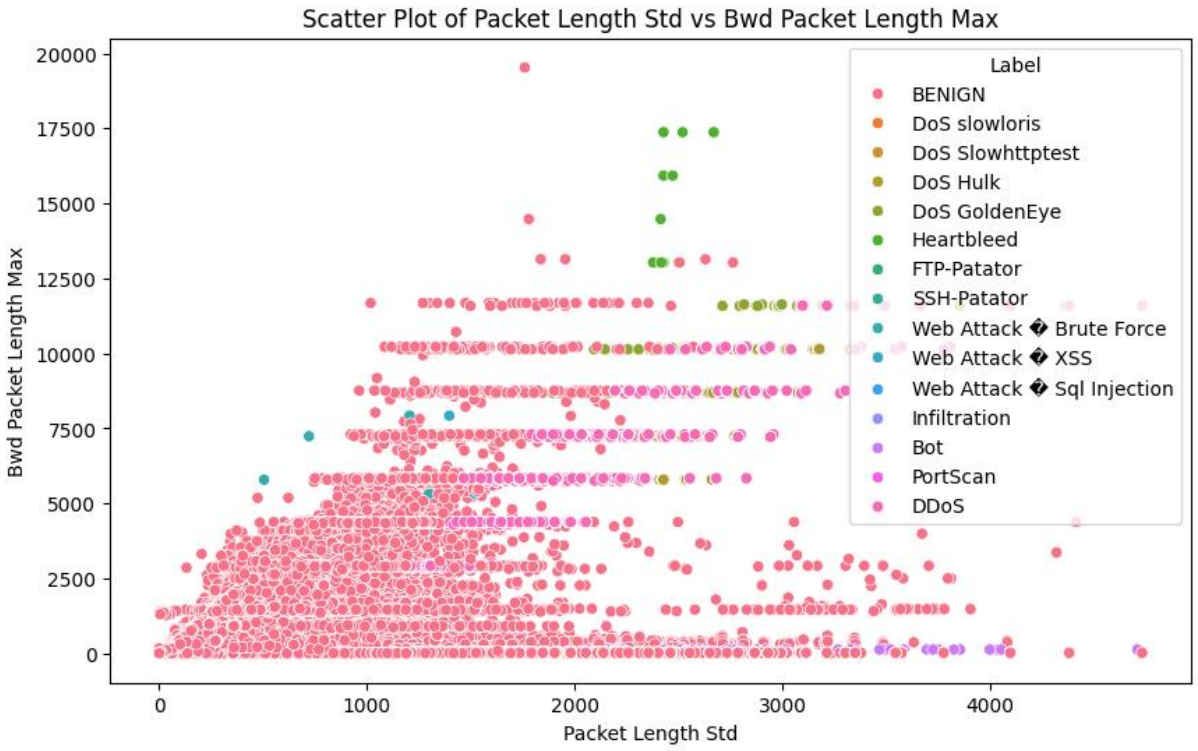

Figure 9: Scatter Plot of Packet Length Std vs Bwd Packet Length Max

Looking at the graph, it seems that most types of attacks and normal traffic are clearly separated. Normal traffic usually has lower 'Packet Length Std' and 'Bwd Packet Length Max' values, while attack types have higher values on these two measurements. In particular, the 'DoS Hulk' and 'DDoS' attack types are notable for having high maximum packet-back lengths. This indicates that such attacks create a high volume of data flow on the network.

Web-based attacks such as 'Web Attack & Brute Force', 'Web Attack & XSS' and 'Web Attack & SQL Injection' are grouped in an area where the standard deviation of packet lengths is usually moderate, but the maximum back packet lengths remain lower. This indicates that such attacks are more specialized and target-oriented, but instead of sending large data packets, they are aimed at causing harm by other means.

Attacks such as 'SSH-Patator' and 'FTP-Patator' have relatively low 'Packet Length Std' values, which indicates that these attacks are carried out using more regular data packets.

4. Applying Data Mining Techniques:

A great instance of how data mining techniques can be applied successfully in the area of cybersecurity is the K-Nearest Neighbors (KNN) algorithm. This section focuses on evaluating the model's performance and detecting attacks using the KNN method.

The data are divided into training and test sets, where the test set is selected to make up 20% of the data set. This division was made in order to fairly evaluate the generalization ability of the model.

The features are scaled using Standard Scaler. This step is critical to ensure that each feature has equal weight on the model and to ensure that the algorithm works faster and more efficiently. The KNN model was established with three neighbors. This parameter selection provides a balanced compromise Decoupling between the complexity of the model and the calculation time. The model training was carried out on scaled training data, and then predictions were made on the test set. These steps are important for observing how the model performs in practice.

The accuracy of the model has been calculated at a very high rate, such as 99.34%. However, low precision and recall values were observed for some classes in the classification report. In particular, the performance of the model is poor in some minority classes (such as label 13 and 14), which reveals the weaknesses of the model on unstable data sets.

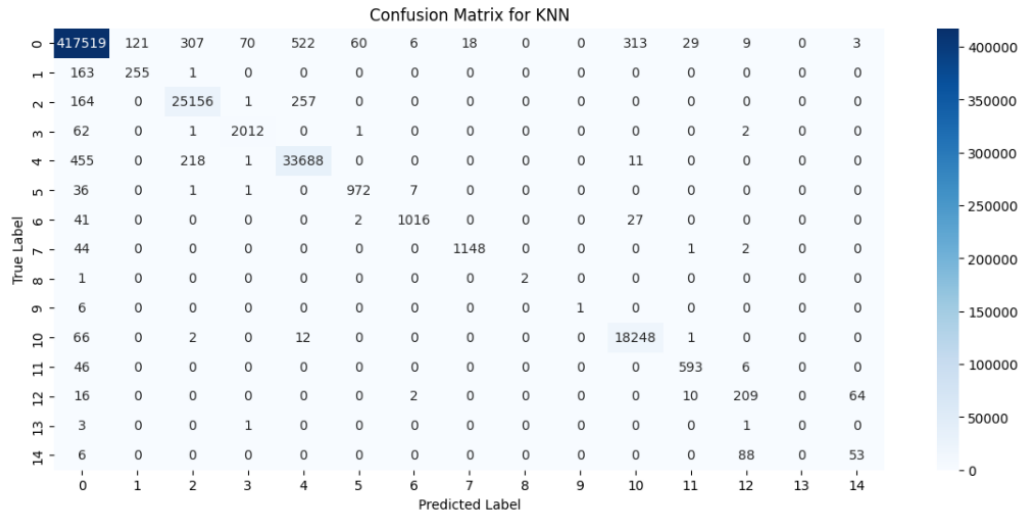

Figure 10: Confusion Matrix for KNN

5. Conclusion

In conclusion, the analysis conducted on the CIC-IDS 2017 dataset using various data preprocessing, exploratory data analysis (EDA), and data mining techniques has provided valuable insights into cyber threat detection and mitigation. Through careful data preprocessing steps such as handling missing values, removing duplicates, and identifying and dropping columns with constant values, the dataset was refined for further analysis. Exploratory data analysis, including correlation heatmap and scatter plots, revealed significant relationships between features and provided a deeper understanding of the dataset's structure. Additionally, feature importance analysis using the Random Forest classifier highlighted crucial attributes for detecting anomalies in network traffic.

The application of the K-Nearest Neighbors (KNN) algorithm demonstrated its effectiveness in classifying cyber threats, achieving a remarkable accuracy score of 99.34%. This high accuracy, coupled with KNN's simplicity and transparency, makes it a suitable choice for cybersecurity tasks. Moreover, principal component analysis (PCA) further enhanced KNN's performance by reducing the dataset's dimensionality, facilitating efficient handling of high-dimensional data and multi-class classification.

Comparatively, the Random Forest model exhibited robustness with a respectable accuracy of 94%, showcasing its capability to handle complex datasets. Despite its lower accuracy of 13%, Gaussian Naive Bayes (GNB) remains relevant for its simplicity and speed in certain contexts.

In summary, the combination of these models offers a comprehensive approach to analyzing network traffic for intrusion detection and anomaly recognition. Among them, KNN emerges as the most accurate and interpretable model, emphasizing its significance in cybersecurity applications.

5.1. Literature Survey Findings

Adaptive Machine Learning Based Network Intrusion Detection [5]

The model classifier tuning began with identifying optimal features from the CICIDS 2017 validation set. Classifier base models were then fine-tuned based on their respective performance indices (PI) using a 69-feature set. PI and PCA were compared to determine the most effective feature reduction method. This step proved crucial for RF, DT, KNN, and MLP classifiers. PCA generally outperformed other extraction methods for dimensionality reduction. Hyperparameters were selected based on the bias/variance tradeoff, and the best models were evaluated to demonstrate their efficacy with limited data. Multi-class classification models were trained for all classes in the CICIDS 2017 dataset. Table 1 presents the best results achieved by the NIDS models on limited data, with 26PI-RF achieving the highest accuracy.

Model	Accuracy	Precision	Recall	f1-score (macro)
26PI-RF	≈1	0.90	0.86	0.87
Q-26PI-KNN	≈1	0.80	0.86	0.82
Q-36PI-MLP	≈1	0.83	0.85	0.81
Q-26PI-SVM	≈1	0.82	0.84	0.81
16PI-DT	≈1	0.78	0.83	0.76
Q-69PI-RNN	≈1	0.74	0.82	0.73

Figure 11: Comparison of different models [5]

Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset [6]

This study assesses MLAIDS algorithms' attack detection effectiveness using the CICIDS2017 dataset, benchmarking 10 classification algorithms categorized into supervised (kNN, SVM, DT, RF, ANN, NB, CNN) and unsupervised learning (k-means clustering, EM, SOM). Testing involved 48 models, with 31 models reported here after excluding those with poor results. Results in Table 16 highlight k-NN, DT, and NB as superior in detecting web attacks, especially among supervised algorithms. DT and k-NN perform best when considering training and testing times. EM stands out as the top unsupervised algorithm, regardless of time considerations.

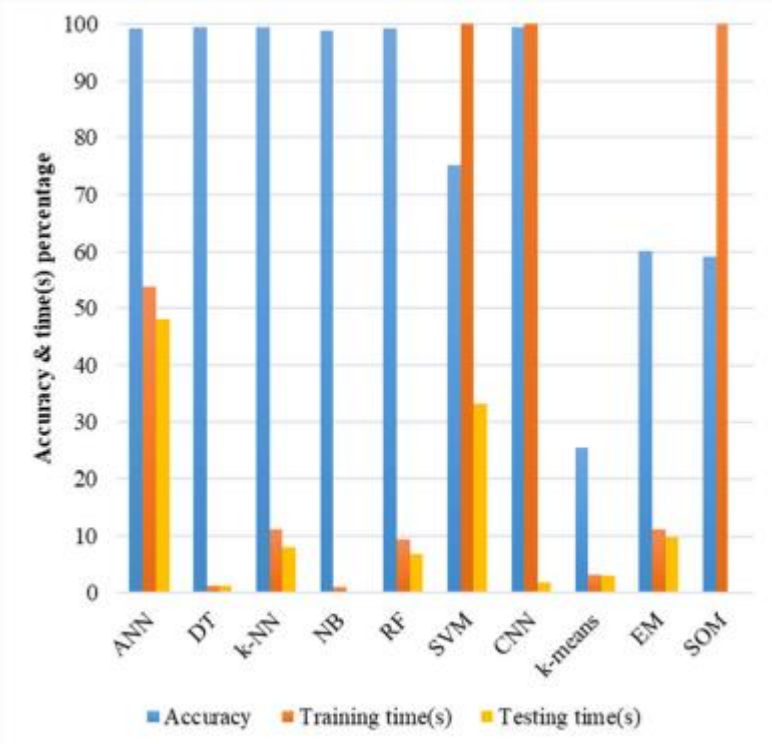

Figure 12: The overall accuracy and runtime performance. [6]

Refrences

[1] A. Kim, M. Park and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," in IEEE Access, vol. 8, pp. 70245-70261, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2986882

[2] Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M. Y., Bamhdi, A. M., & Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access, 8, 132911–132912. https://doi.org/10.1109/access.2020.3009843

[3] Bruno Reis, Eva Maia, and Isabel Praça. 2019. Selection and Performance Analysis of CICIDS2017 Features Importance. In International Symposium on Foundations and Practice of Security. Springer, Springer International Publishing, Cham, 56–71

[4] Arnaud Rosay, Florent Carlier, and Pascal Leroux. 2020. MLP4NIDS: An Efficient MLP-Based Network Intrusion Detection for CICIDS2017 Dataset. In Machine Learning for Networking, Selma Boumerdassi, Éric Renault, and Paul Mühlethaler (Eds.). Springer International Publishing, Cham, 240–254

[5] Hatitye Chindove and Dane Brown. 2021. Adaptive Machine Learning Based Network Intrusion Detection. In Proceedings of the International Conference on Artificial Intelligence and its Applications (icARTi '21). Association for Computing Machinery, New York, NY, USA, Article 15, 1–6. https://doi.org/10.1145/3487923.3487938

[6] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," in IEEE Access, vol. 9, pp. 22351-22370, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3056614