Anomaly Detection in Networks: An Analysis of UNSW-NB 15 Dataset

Shruti Mishra | 3742137 | GitHub

1. Problem Statement

a. Value of Solving the Problem

The significance of addressing cybersecurity threats has never been more critical, given the increasing frequency and sophistication of cyber-attacks. The UNSW-NB 15 dataset provides a valuable opportunity to develop a machine learning solution for detecting and classifying various types of cyber threats. The successful implementation of such a solution can contribute to enhancing the overall security posture of networks, protecting sensitive information, and minimizing the potential impact of cyber-attacks.

b. End-User and Potential Impact

The end-users of this solution would include organizations and enterprises that rely on secure network infrastructures. The impact of cyber-attacks on these entities can range from financial losses and reputational damage to the compromise of sensitive data. By deploying an effective machine learning model, these end-users can proactively identify and mitigate potential threats, thereby safeguarding their digital assets.

c. Constraints and Requirements

- **Computational Resources:** The algorithm should be designed to operate efficiently, considering the potential deployment in resource-constrained environments.
- **Real-time Processing:** Depending on the network's nature, real-time or near-real-time processing might be a requirement for timely threat detection.
- **Scalability:** The solution should be scalable to handle varying network sizes and complexities.

d. Type of ML Problem

The problem at hand is a supervised classification problem. Given the features extracted from network traffic, the goal is to train a model capable of accurately classifying instances into one of the nine predefined attack categories.

e. Success Metrics

The success of the machine learning model will be evaluated based on several metrics, including:

- **Accuracy:** Overall correctness of the model's predictions.
- **Precision:** Proportion of true positive predictions out of the total predicted positives.
- **Recall:** Proportion of true positive predictions out of the total actual positives.
- **F1 Score:** Harmonic mean of precision and recall, providing a balanced metric.
- **Feature Importance Analysis:** Understanding and leveraging the most impactful features for intrusion detection.

2. Solution Design

a. Literature Review

i. Background and Related Works

The challenge of anomaly detection and cyber threat classification is well-established in the field of cybersecurity. Various datasets, such as KDD Cup 99 and NSL-KDD, have been employed for similar purposes. However, the unique characteristics of the UNSW-NB 15 dataset, including its hybrid nature of real and synthetic data, make it a distinct and challenging problem.

KDDCUP99 Data Set:

The KDDCUP99 dataset was created through a simulation on a military network environment (U.S. Air Force LAN) and involved nine weeks of raw tcpdump files. The simulation generated training data of approximately four GBs, which was then processed into about five million connection records. The dataset included 22 attack types in the training set and 15 attack types in the test set.

How it was solved:

- The dataset featured 41 features for each connection, along with a class label, derived using the Bro-IDS tool.
- Various researchers utilized this dataset for intrusion detection system (IDS) evaluation due to its public availability.

Limitations:

- **TTL Discrepancy:** One criticism was that the time-to-live (TTL) values of attack data packets (126 or 253) were different from the TTL values in the training records of the attack. This raised concerns about the dataset's representativeness.
- **Distribution Discrepancy:** The probability distribution of the testing set was reported to differ from that of the training set due to the addition of new attack records, potentially biasing classification methods.
- **Lack of Representation:** Another drawback was that the dataset did not comprehensively represent recently reported low-footprint attack projections.

State-of-the-Art at the Time: While the KDDCUP99 dataset was widely used, the techniques employed at that time, including traditional machine learning algorithms like decision trees and SVMs, can now be considered as precursors to more advanced approaches. Ensemble learning and deep learning were not as prevalent during the era of KDDCUP99.

NSL-KDD Data Set:

To address some limitations of KDDCUP99, the NSL-KDD dataset was created with the goals of removing duplication, selecting diverse records, and balancing the number of records in the training and testing phases.

How it was solved:

- Duplication of records in the training and test sets was removed to eliminate classifier bias towards repeated records.

- A variety of records from different parts of the original KDD dataset were selected to achieve reliable results from classifier systems.
- Unbalance among the number of records in the training and testing phases was addressed to decrease False Alarm Rates (FARs).

Limitation: The major criticism of NSL-KDD was that it did not represent modern low-footprint attack scenarios.

State-of-the-Art at the Time: While NSL-KDD introduced improvements over KDDCUP99, the techniques used were still rooted in traditional machine learning. State-of-the-art approaches involving ensemble learning, deep learning, and adaptive learning mechanisms have emerged in more recent years, pushing the boundaries of intrusion detection systems. The methodologies applied to KDDCUP99 and NSL-KDD can be seen as foundational steps in the evolution toward more sophisticated techniques employed in contemporary intrusion detection research.

ii. Proposed Novelty

Building on the existing landscape of intrusion detection, this project introduces novel elements to enhance the efficacy of cyber threat classification on the UNSW-NB 15 dataset.

1. **Ensemble Learning Fusion:**
    - A novel approach involves fusing the outputs of diverse machine learning models, including Random Forests, Decision Trees and Gaussian Bayes networks.
    - By leveraging the strengths of each model, the ensemble learning fusion aims to create a more robust and accurate intrusion detection system.
2. **Dynamic Feature Engineering:**
    - Feature engineering plays a pivotal role in extracting meaningful information from the dataset.
    - The project introduces dynamic feature engineering techniques that adapt to the hybrid nature of the UNSW-NB 15 dataset, capturing both real-world and synthetic attack behaviors.

iii. References of Previous Related Work

1. Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.
2. Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset." *Information Security Journal: A Global Perspective* (2016): 1-14.
3. Moustafa, Nour, et al. "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks." *IEEE Transactions on Big Data (2017).*
4. Moustafa, Nour, et al. "Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models." *Data Analytics and Decision Support for Cybersecurity. Springer, Cham, 2017.* 127-156.
5. Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings* (p. 117). Springer Nature.

b. From perspective of ML workflow

i. Dataset selection

The dataset chosen for this study is the UNSW-NB15 dataset, a widely recognized cybersecurity dataset for network anomaly/intrusion detection. This dataset is a compilation of diverse network traffic scenarios, encompassing both normal and malicious activities. The raw network packets of the UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours.

Tcpdump tool is utilised to capture 100 GB of the raw traffic (e.g., Pcap files). Pcap refers to packet capture, which contains an Application Programming Interface (API) for saving network data. The UNIX operating systems execute the pcap format using the libpcap library while the Windows operating systems utilise a port of libpcap, called WinPcap.

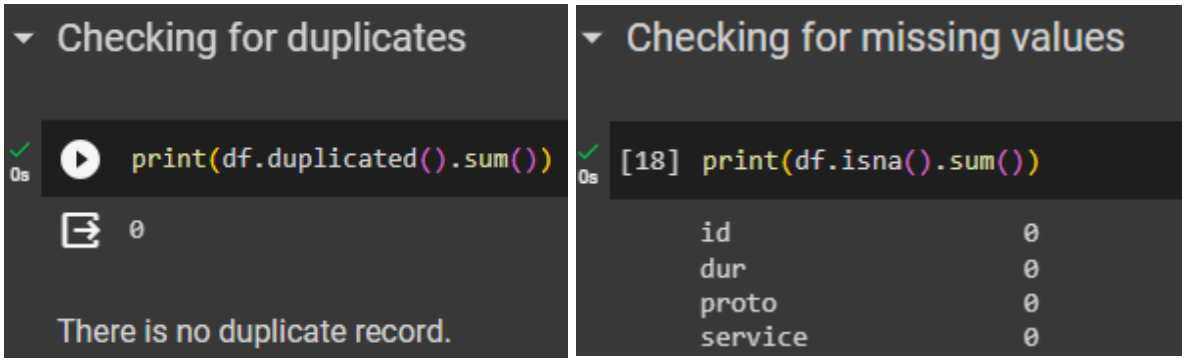
This dataset has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The Argus, Bro-IDS tools are used and twelve algorithms are developed to generate totally 49 features with the class label.

ii. Data pre-processing

Given the raw network packets captured by the IXIA PerfectStorm tool, data pre-processing is a critical step.

1. Data Quality Assurance:

- The initial steps of the data preprocessing involved checking for duplicate records and ensuring there were no missing values. The absence of duplicates ensures the integrity of the dataset, while the absence of missing values indicates that the dataset is complete and ready for further analysis.



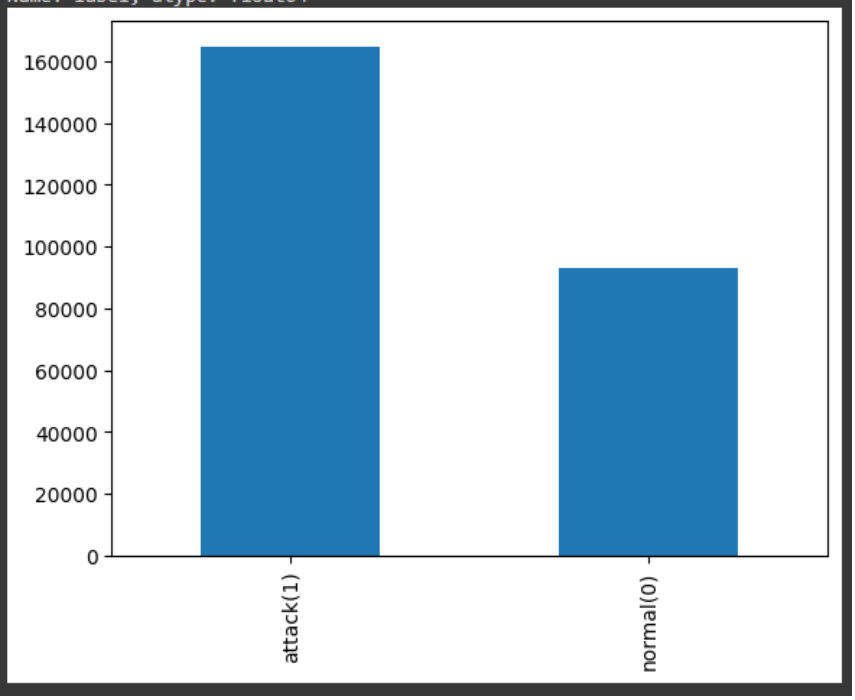
2. Class Distribution Analysis:

- An essential aspect of understanding the dataset's nature was to analyze the distribution of classes, specifically the balance between benign and attack data. The visualization of class distribution highlighted a slight imbalance, with approximately 64% of instances being

attacks and 36% being normal instances. The decision not to address this slight imbalance was supported by a thoughtful analysis of its impact on the machine learning model. Check for class imbalances in the 'attack_cat' column. If certain attack categories are significantly underrepresented, consider applying techniques such as oversampling, undersampling, or using synthetic data generation methods.

```
df['label'] = df['label'].map({0: 'normal(0)', 1: 'attack(1)'})
df['label'].value_counts().plot.bar()
df['label'].value_counts(normalize=True)
```

```
attack(1)    0.639077
normal(0)    0.360923
Name: label, dtype: float64
```



3. Feature Engineering:

- Streamlining the dataset for optimal model performance involved feature engineering. Unnecessary columns, such as 'id' and 'attack_cat', were rightfully dropped, focusing the analysis on the key features relevant to the binary classification task.

Dropping the column id. This is just for identification, so we can remove this column.

```
[19] df = df.drop(columns=['id'])
```

0s

```
[ ]  df = df.drop(columns=['attack_cat'])
```

4. Categorical Feature Encoding:

The categorical features ('proto', 'service', 'state') were appropriately encoded using the LabelEncoder, ensuring that these non-numeric variables were transformed into a format compatible with machine learning models.

```
[ ] for col in ['proto', 'service', 'state']:
        df[col] = df[col].astype('category').cat.codes

    df['attack_cat'] = df['attack_cat'].astype('category')
    df.head()
```

	dur	proto	service	state	spkts	dpkts	sbytes
0	0.000011	119	0	5	2	0	496
1	0.000008	119	0	5	2	0	1762
2	0.000005	119	0	5	2	0	1068
3	0.000006	119	0	5	2	0	900
4	0.000010	119	0	5	2	0	2126

5 rows × 44 columns

The data is now prepared for further modeling steps. The context provided by these data preprocessing steps establishes a foundation for subsequent stages of the machine learning workflow, including model selection, training, and evaluation.

iii. Model selection

In the pursuit of effective anomaly detection, a variety of machine learning models were considered for their suitability. The models encompassed for evaluation were:

1. **Decision Tree Classifier:** Known for simplicity and interpretability, decision trees are effective in capturing non-linear patterns in data.
2. **Random Forest Classifier:** An ensemble of decision trees, Random Forest excels in handling complex relationships, reducing overfitting, and providing robust predictions.
3. **Gaussian Naive Bayes:** Built on Bayes' theorem, this model assumes conditional independence of features, making it suitable for well-behaved datasets.

```
[54] df_models
```

	Training score	Accuracy	Precision	Recall	Training time
Decision Tree Classifier	0.997755	0.937259	0.950677	0.951254	4.735999
Random Forest Classifier	0.997749	0.950325	0.962493	0.959729	47.798756
Gaussian Naive Bayes	0.818990	0.817443	0.856704	0.858091	0.162169

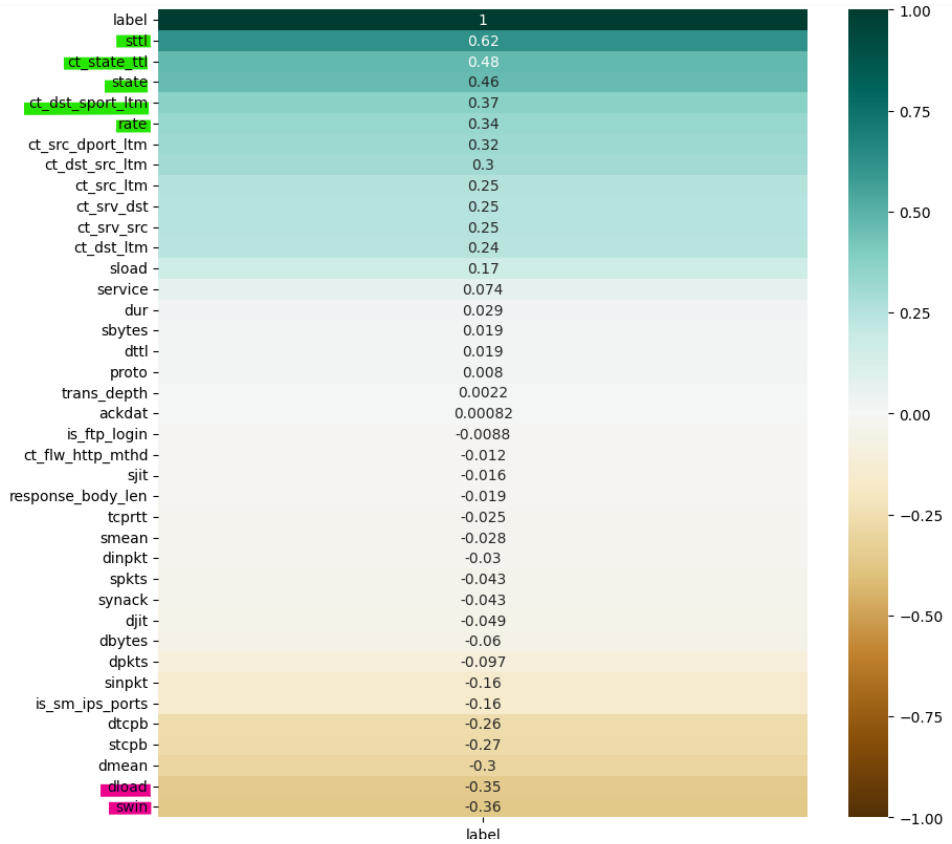
iv. Training/ Fine-tuning

- The dataset, comprising 180,371 instances in the training set and 77,302 instances in the testing set, underwent meticulous preprocessing to ensure data quality and consistency. A crucial step involved standard scaling, which normalized the features and put them on a comparable scale, preventing any feature from dominating the model training process.
- Subsequently, the three selected models – Decision Tree Classifier, Random Forest Classifier, and Gaussian Naive Bayes – were trained on the preprocessed data. Each model's training performance was assessed, providing valuable insights into their initial capabilities.
- Notably, the Random Forest Classifier emerged as a promising candidate for further optimization due to its ensemble nature and ability to handle complex relationships within the data.

v. Hyperparameter Tuning Strategy:

- Before delving into the hyperparameter tuning strategy, an essential preliminary step involved understanding the correlation of variables with cyber-attacks. A heatmap was generated to visualize the relationships between different features and the target variable, 'label,' signifying cyber-attacks. The heatmap revealed both positive and negative correlations between certain features and cyber-attacks:

  Positively Correlated Features: sttl, ct_state_ttl, state, ct_dst_sport_ltm, rate.
  Negatively Correlated Features: swin, dload.



- A rule-based system was then employed to filter data for potential attacks based on specific conditions related to features like 'sttl'. The rule system effectively filtered out 23% of network traffic, showcasing its efficacy in detecting non-threatening network activity.

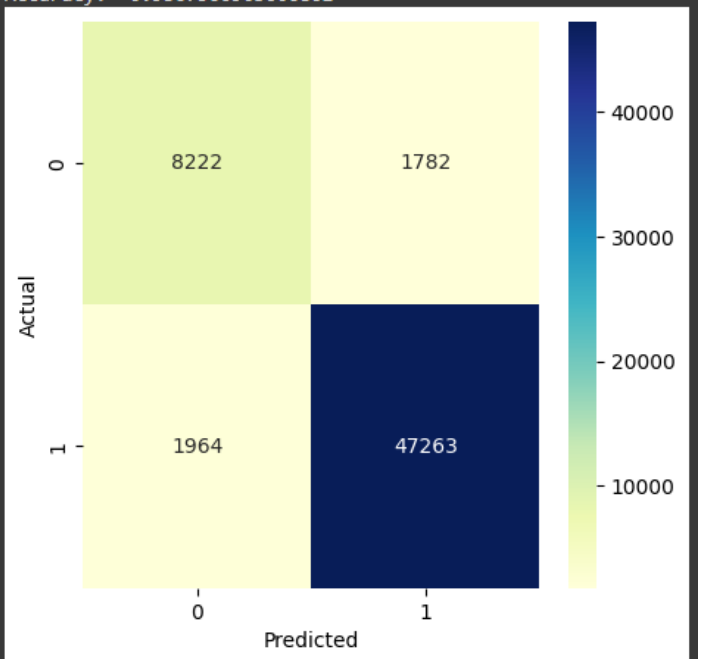
```
[ ]  X_test = X_test.reset_index(drop=True)

     rules= "(sttl <= 61.00 & sinpkt<= 0.00) | (sttl >  61.00 )"

     ind = X_test.query(rules).index

     X_test_2 = X_test.loc[ind,:]
     y_test_2 = y_test[ind]

     print(X_test.shape)
     print(X_test_2.shape)
     print("filtered data" , (1- np.round(X_test_2.shape[0] / X_test.shape[0],2))*100, "%")
```

```
(77302, 38)
(59231, 38)
filtered data 23.0 %
```

- Subsequently, the filtered data underwent evaluation using the Random Forest Model. The recall, precision, and accuracy metrics were reported, demonstrating the model's ability to correctly identify cyber-attacks.

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(random_state=11)

results['Random Forest Model'] = model_evaluation(rf)
```

```
Recall:  0.9601031954008978
Precision:  0.9636660210011214
Accuracy:  0.9367560905606861
```



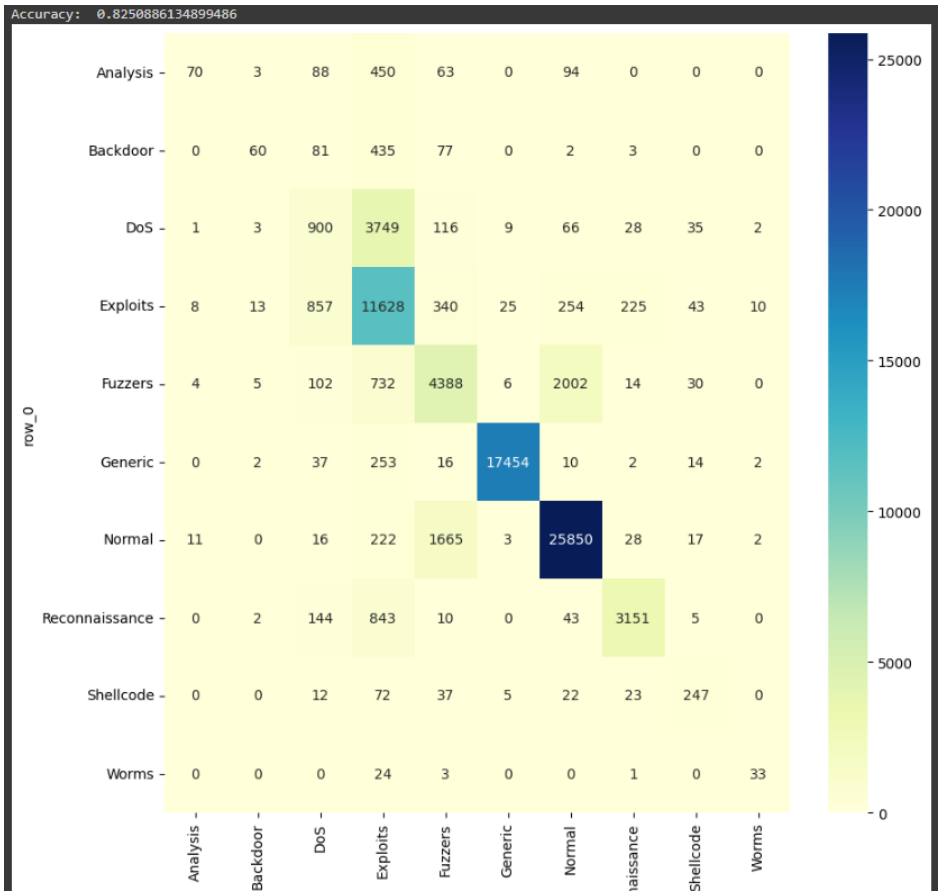
- Feature importance analysis was conducted to identify the top contributing features. The Random Forest Model ranked features such as 'sttl,' 'ct_state_ttl,' and 'rate' as the most influential in predicting cyber-attacks.

```
feature_imp = pd.DataFrame({'Name':X.columns, 'Importance':rf.feature_importances_})

feature_imp = feature_imp.sort_values('Importance',ascending=False).reset_index(drop=True)

feature_imp[:10].style.background_gradient()
```

	Name	Importance
0	sttl	0.144773
1	ct_state_ttl	0.093865
2	rate	0.056895
3	dload	0.046490
4	sload	0.044910
5	sbytes	0.043058
6	ct_srv_dst	0.038810
7	smean	0.037594
8	dttl	0.036539
9	ct_dst_src_ltm	0.036140

- The final evaluation involved applying the Random Forest Model with optimized hyperparameters to predict the type of cyber-attacks. While achieving an accuracy of

82.43%, the model's performance was detailed through a confusion matrix, providing a granular view of predictions across different attack categories.



- In conclusion, the hyperparameter tuning strategy involved a meticulous exploration of features, rule-based filtering, and model evaluation to enhance the Random Forest Model's effectiveness in identifying and categorizing cyber-attacks. The iterative nature of this process reflects a commitment to refining the model for real-world deployment in cybersecurity scenarios.

vi. Evaluation metrics

The model's performance was evaluated using key intrusion detection metrics:

- **Accuracy:** Overall correctness of predictions.
- **Precision:** Proportion of true positives among predicted positives.
- **Recall:** Proportion of true positives among actual positives.
- **F1 Score:** The harmonic mean of precision and recall, providing a balance between the two metrics.

Additionally, a detailed classification report was generated, offering insights into precision, recall, and F1-score for each class of attacks. The confusion matrix and heatmaps visually depicted the model's ability to correctly classify instances and highlighted areas where improvements could be made.

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
Analysis	0.76	0.09	0.16	768
Backdoor	0.55	0.07	0.13	658
DoS	0.38	0.17	0.23	4909
Exploits	0.63	0.87	0.73	13403
Fuzzers	0.65	0.60	0.62	7283
Generic	1.00	0.98	0.99	17790
Normal	0.91	0.93	0.92	27814
Reconnaissance	0.91	0.75	0.82	4198
Shellcode	0.62	0.58	0.60	418
Worms	0.67	0.51	0.58	61
accuracy			0.82	77302
macro avg	0.71	0.55	0.58	77302
weighted avg	0.82	0.82	0.81	77302

Observation: Generic, Normal, and Exploits attacks have high recalls.

Additional Insights:

- **Feature Importance:** Random Forest's feature importance analysis revealed the top 10 features contributing to the model's decision-making process. This analysis assisted in feature selection, focusing on the most impactful variables for intrusion detection.
- **Rule-based Filtering:** A rule-based system was implemented to filter data for potential attacks, demonstrating the effectiveness of domain-specific knowledge in refining the dataset for analysis.
- **Top 10 Feature Subset:** A subset analysis using only the top 10 features further emphasized the importance of specific variables in achieving a high level of accuracy in intrusion detection.

3. Implementation

Shared in GitHub.

4. Conclusions and Discussions

In conclusion, the implemented Random Forest Classifier demonstrated strong capabilities in anomaly/intrusion detection, achieving high accuracy and effectively identifying cyber threats. The combination of feature importance analysis, rule-based filtering, and focused feature subsets contributed to a refined and efficient model.

Drawbacks:

1. **Imbalanced Data:** The slight imbalance in the dataset may impact model generalization. Addressing this imbalance could enhance performance.
2. **Feature Engineering:** While feature importance analysis guides feature selection, a deeper exploration of feature engineering methods could refine the model's decision boundaries.

Future Scope:

1. **Advanced Models:** Exploring advanced models like deep learning architectures could uncover intricate patterns, potentially enhancing detection capabilities.
2. **Ensemble Techniques:** Combining multiple models through ensemble techniques might further boost accuracy and robustness.
3. **Real-time Implementation:** Adapting the model for real-time intrusion detection could be pivotal for dynamic cybersecurity scenarios.

In conclusion, while the current model excels, continual refinement and exploration of advanced methodologies remain pivotal for enhancing intrusion detection systems' efficacy.