MODC 2324

Grupo10

Henrique Catarino – 56278

João Osório – 56353

Vasco Maria – 56374

Dataset Characterization and ML Goals

Dataset Topic: Cybersecurity Attacks

HDD Size: 17.86 MB

Number of Features: 25

Number of Instances: 40,000

URL: Kaggle - https://www.kaggle.com/datasets/teamincribo/cyber-security-attacks?resource=download&select=cybersecurity_attacks.csv

Access Date: 25/05/2024

Dataset Description

The dataset is about cyberattacks, containing detailed information about various network traffic features and their associated attack types. Each instance represents a unique network event with features such as source and destination IP addresses, ports, protocol types, packet lengths, and more, where we will be looking to aim at identifying and categorizing the type of cyberattack.

Machine Learning Goals

The primary objective is to utilize machine learning techniques to predict the "Attack Type" based on the other existing features in the dataset. By constructing a robust predictive model, the goal is to enhance the ability to identify and classify different types of cybersecurity attacks accurately.

Methodology and Expected Outcomes

Methodology: Exploratory Data Analysis (EDA), Handling Missing Data, Handling Categorical Variables, Normalization of Data, Feature Selection, Model Building and Evaluation

Expected Outcomes: While the dataset is not the largest, limiting the potential for achieving very high accuracy (or other metrics), it serves as an excellent foundation for demonstrating key concepts and workflows in machine learning. This project will showcase the process of building and evaluating a predictive model for cybersecurity attack classification. The insights and methods derived from this exercise are scalable and can be applied to larger datasets, which would likely yield improved performance and more significant results. By understanding and implementing these foundational techniques, we set the stage for more complex and high-performing models in larger, real-world datasets.

Exploratory Data Analysis (EDA) Results

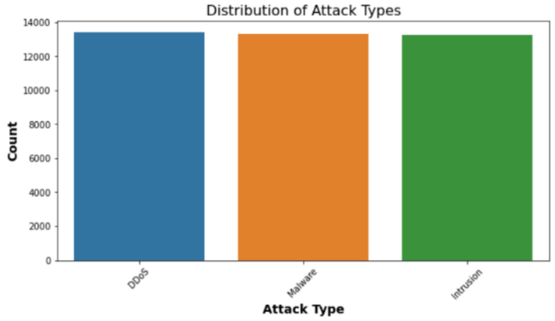
In the exploratory data analysis phase, we examined the dataset to understand its structure, the types of data it contains, and any potential issues such as missing values. The dataset consists of 40,000 instances and 25 features, including various network traffic attributes such as source and destination IP addresses, ports, protocol types, packet lengths, and more.

The dataset contains a mix of numerical and categorical variables, with the following breakdown:

Numerical features: Source Port, Destination Port, Packet Length, Anomaly Scores

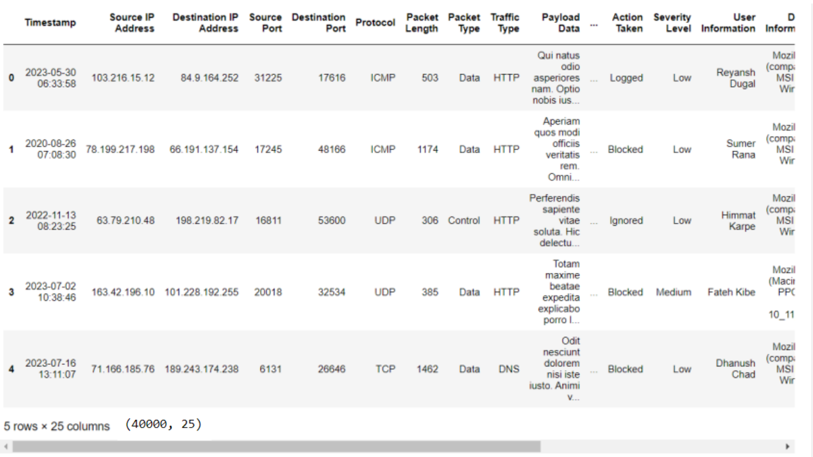
Categorical features: Timestamp, Source IP Address, Destination IP Address, Protocol, Packet Type, Traffic Type, Payload Data, Malware Indicators, Alerts/Warnings, Attack Type, Attack Signature, Action Taken, Severity Level, User Information, Device Information, Network Segment, Geo-location Data, Proxy Information, Firewall Logs, IDS/IPS Alerts, Log Source

We also checked for imbalance in our target variable (Figura 1) and found that the classes were quite balanced. This balance in the target variable helps ensure that the machine learning models trained on this data can potentially achieve better performance and generalize well to unseen data.



```
DDoS         13428
Malware      13307
Intrusion    13265
Name: Attack Type, dtype: int64
```

Figura 1 – EDA - Checking for class imbalance in target variable



	Timestamp	Source IP Address	Destination IP Address	Source Port	Destination Port	Protocol	Packet Length	Packet Type	Traffic Type	Payload Data	...	Action Taken	Severity Level	User Information	D Inform
0	2023-05-30 06:33:58	103.216.15.12	84.9.164.252	31225	17616	ICMP	503	Data	HTTP	Qui natus odio asperiores nam. Optio nobis ius...	...	Logged	Low	Reyansh Dugal	Mozil (comp MSI Wir
1	2020-08-26 07:08:30	78.199.217.198	66.191.137.154	17245	48166	ICMP	1174	Data	HTTP	Aperiam quos modi officiis veritatis rem. Omni...	...	Blocked	Low	Sumer Rana	Mozil (comp MSI Wir
2	2022-11-13 08:23:25	63.79.210.48	198.219.82.17	16811	53600	UDP	306	Control	HTTP	Perferendis sapiente vitae soluta. Hic delectu...	...	Ignored	Low	Himmat Karpe	Mozil (comp MSI Wir
3	2023-07-02 10:38:46	163.42.196.10	101.228.192.255	20018	32534	UDP	385	Data	HTTP	Totam maxime beatae expedita explicabo porro l...	...	Blocked	Medium	Fateh Kibe	Mozil (Macir PPC 10_11
4	2023-07-16 13:11:07	71.166.185.76	189.243.174.238	6131	26646	TCP	1462	Data	DNS	Odit nesciunt dolorem nisi iste iusto. Animi v...	...	Blocked	Low	Dhanush Chad	Mozil (comp MSI Wir

5 rows × 25 columns (40000, 25)

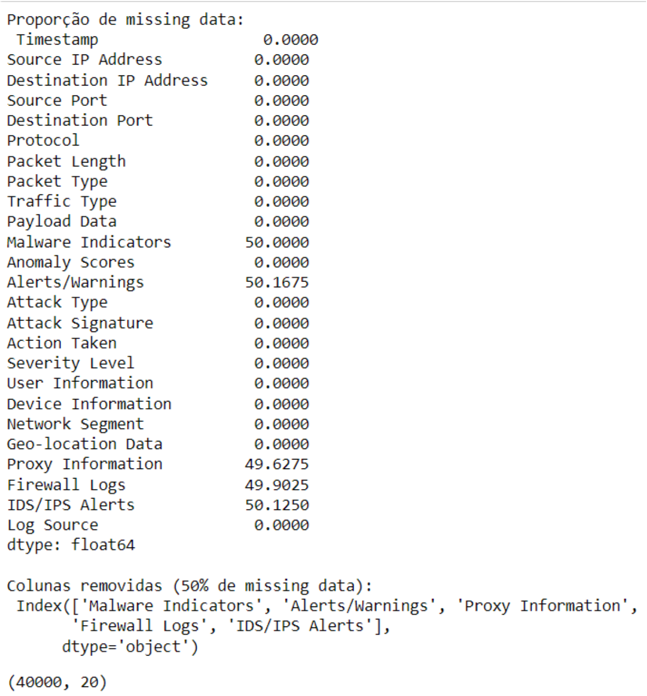
Figura 2 - EDA - Checking some data

Missing Data Analysis

A significant aspect of our analysis was identifying missing data within the dataset. Several features have a high percentage of missing values (Figura 3), such as Malware Indicators (50%), Alerts/Warnings (50.17%), Proxy Information (49.63%), Firewall Logs (49.90%), and IDS/IPS Alerts (50.13%).

We decided to exclude the features Malware Indicators, Alerts/Warnings, Proxy Information, Firewall Logs, and IDS/IPS Alerts due to their high percentage of missing data, which is approximately half (50%). Given the context of the dataset and the substantial proportion of missing values, we determined that imputing these values would not be meaningful. Instead, excluding these features will help maintain the integrity and

relevance of the remaining data for our analysis.
As for the rest of the features, there we no missing values to be handled.



```
Proporção de missing data:
 Timestamp                 0.0000
Source IP Address         0.0000
Destination IP Address    0.0000
Source Port               0.0000
Destination Port          0.0000
Protocol                  0.0000
Packet Length             0.0000
Packet Type               0.0000
Traffic Type              0.0000
Payload Data              0.0000
Malware Indicators       50.0000
Anomaly Scores            0.0000
Alerts/Warnings          50.1675
Attack Type               0.0000
Attack Signature          0.0000
Action Taken              0.0000
Severity Level            0.0000
User Information          0.0000
Device Information        0.0000
Network Segment           0.0000
Geo-location Data         0.0000
Proxy Information        49.6275
Firewall Logs            49.9025
IDS/IPS Alerts           50.1250
Log Source                0.0000
dtype: float64

Colunas removidas (50% de missing data):
 Index(['Malware Indicators', 'Alerts/Warnings', 'Proxy Information',
       'Firewall Logs', 'IDS/IPS Alerts'],
      dtype='object')

(40000, 20)
```

Figura 3 - Missing data and removal of columns

Categorical Variables

To prepare the categorical variables for machine learning models, we employed ordinal encoding. This technique converts categorical values into numerical values, allowing models to process them effectively. Initially, we identified the categorical and numerical columns in the dataset. The categorical columns included features such as Timestamp, Source IP Address, Destination IP Address, Protocol, Packet Type, Traffic Type, Payload Data, and others. We excluded the target variable Attack Type from the encoding process to maintain its integrity for prediction purposes. Using the OrdinalEncoder from scikit-learn, we transformed the categorical variables into numerical values. This encoding assigns an integer value to each category within a feature, maintaining the categorical distinctions while enabling their use in numerical computations. The encoded dataset retains its original shape but now consists of numeric values suitable for machine learning algorithms.

Normalization of Data

Normalization is an essential step in data preprocessing for machine learning as it ensures that all features contribute equally to the model, preventing features with larger scales from dominating the learning process. This step is particularly important when using algorithms that rely on distance calculations, such as k-nearest neighbors or support vector machines. Normalization transforms the data to a common scale, which can improve the convergence rate and performance of the models.
In our approach, we tested different normalization techniques to identify the most effective method for our dataset. The techniques evaluated included StandardScaler,

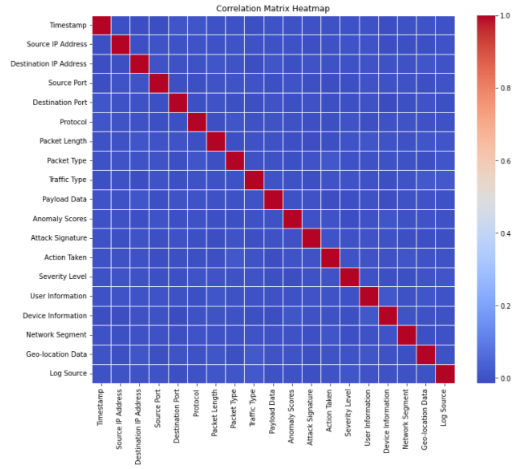
which standardizes features by removing the mean and scaling to unit variance; MinMaxScaler, which scales features to a specified range (typically 0 to 1); RobustScaler, which scales features using statistics that are robust to outliers by removing the median and scaling according to the interquartile range; and Normalizer, which scales individual samples to have unit norm.
We implemented a function to evaluate each normalization method using a RandomForestClassifier and measured the performance using the F1-Score with 80-20 train-test split. The results indicated that while all normalization techniques provided similar F1-Scores, the Normalizer achieved the highest score by a small margin. This suggests that scaling individual samples to unit norm slightly improved the model's ability to classify attack types. Despite the close performance among the methods, the normalization process demonstrated its value by ensuring consistent and comparable feature contributions, ultimately leading to more reliable model performance.

Feature Selection

Feature selection is a crucial step in the machine learning workflow as it aims to select the most relevant features from the dataset, thereby improving the model's performance and reducing computational complexity. In our analysis, we performed feature selection using correlation analysis and a Random Forest model to identify and retain the most significant features.

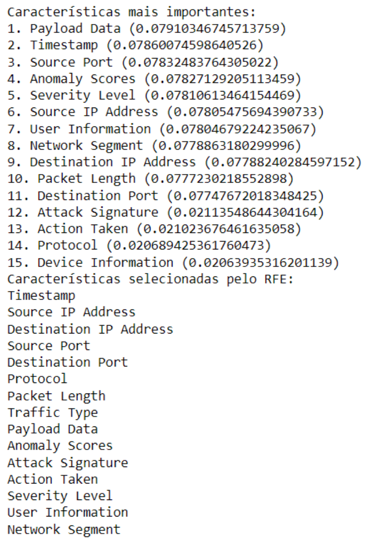
Correlation Analysis

We began the feature selection process with a correlation analysis (Figura 4) to understand the relationships between the features. By calculating the correlation matrix and visualizing it with a heatmap, we were able to identify features that were highly correlated with each other. The heatmap of the correlation matrix provided a clear visual representation of these relationships. The main findings from the correlation analysis are as follows: The highest correlations observed were very low, indicating that there is little to no multicollinearity between the features. The highest absolute correlations include: Traffic Type and Log Source (0.0144); Destination Port and Device Information (0.0134); Traffic Type and Geo-location Data (0.0136)
The low correlation values suggest that each feature contributes uniquely to the model, and no immediate feature elimination was performed based on correlation alone.



```
Maiores correlações absolutas:
 Traffic Type      Log Source                0.014431
Destination Port  Device Information        0.014327
Traffic Type      Geo-location Data         0.013625
Packet Type       Action Taken              0.012971
User Information  Device Information        0.011157
Timestamp         Destination IP Address    0.009840
Packet Length     Attack Signature          0.009614
Protocol          Packet Type               0.008517
Timestamp         Source Port               0.008328
Packet Type       Log Source                0.007782
dtype: float64
```

Figura 4 - Correlation Analysis



```
Características mais importantes:
1. Payload Data (0.07910346745713759)
2. Timestamp (0.07860074598640526)
3. Source Port (0.07832483764305022)
4. Anomaly Scores (0.07827129205113459)
5. Severity Level (0.07810613464154469)
6. Source IP Address (0.07805475694390733)
7. User Information (0.07804679224235067)
8. Network Segment (0.0778863180299996)
9. Destination IP Address (0.07788240284597152)
10. Packet Length (0.0777230218552898)
11. Destination Port (0.07747672018348425)
12. Attack Signature (0.02113548644304164)
13. Action Taken (0.021023676461635058)
14. Protocol (0.020689425361760473)
15. Device Information (0.02063935316201139)
Características selecionadas pelo RFE:
Timestamp
Source IP Address
Destination IP Address
Source Port
Destination Port
Protocol
Packet Length
Traffic Type
Payload Data
Anomaly Scores
Attack Signature
Action Taken
Severity Level
User Information
Network Segment
```

Figura 5 - Feature Selection outputs

Feature Importance using Random Forest

Next, we utilized a Random Forest classifier to determine the importance of each feature. Random Forest is an ensemble learning method that can provide insights into which features are most predictive. We trained the model on the dataset and extracted the feature importances. The top 15 most important features identified by the Random Forest model are (Figura 5):

Payload Data (0.0791); Timestamp (0.0786); Source Port (0.0783); Anomaly Scores (0.0783); Severity Level (0.0781); Source IP Address (0.0781); User Information (0.0780); Network Segment (0.0779); Destination IP Address (0.0779); Packet Length (0.0777); Destination Port (0.0775); Attack Signature (0.0211); Action Taken (0.0210); Protocol (0.0207); Device Information (0.0206)

Recursive Feature Elimination (RFE)

To further refine the feature selection, we applied Recursive Feature Elimination (RFE) using the Random Forest classifier. RFE is a technique that recursively removes less important features and builds the model with the remaining features. We set RFE to select the top 15 features, which are (Figura 5):

Timestamp; Source IP Address; Destination IP Address; Source Port; Destination Port; Protocol; Packet Length; Traffic Type; Payload Data; Anomaly Scores; Attack Signature; Action Taken; Severity Level; User Information; Network Segment

These selected features will be used for further model building and evaluation to ensure that our predictive model is both efficient and effective. The combination of correlation analysis, feature importance from Random Forest, and RFE provides a robust feature selection strategy that helps in enhancing model performance and interpretability.

Model Building and Evaluation

In the model building and evaluation phase, we focused on constructing various machine learning models to predict the "Attack Type" based on the selected features. The models we employed include Decision Tree, Logistic Regression, Random Forest, k-Nearest Neighbors (k-NN), and Naive Bayes classifiers. To ensure a robust evaluation, we implemented k-fold cross-validation, which splits the dataset into k subsets and iteratively trains and tests the model k times. This approach helps in mitigating overfitting and provides a more reliable estimate of model performance.

The results were close, but in the end, the Random Forest classifier showed a slightly better result. The Random Forest classifier is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. This method leverages the wisdom of decision trees to improve performance and reduce the risk of overfitting. Each tree in the forest is trained on a random subset of the data with replacement (bootstrap sampling) and a random subset of features, which enhances the robustness and generalizability of the model.

Next up we decided to further fine-tune the model using Grid Search with Cross-Validation to identify the optimal hyperparameters (Figura 6). The hyperparameters we tuned included the number of estimators, maximum depth, minimum samples split, minimum samples leaf, and the maximum number of features considered for splitting at each node. The Grid Search process evaluated 324 different combinations of these hyperparameters over 5-fold cross-validation, ultimately selecting the combination that showed the highest weighted F1-score. This process was highly computationally intensive and required a substantial amount of time to complete, reflecting the complexity and thoroughness of the hyperparameter optimization.

Once the best hyperparameters were identified, we retrained the Random Forest model on the training dataset and evaluated its performance on the test dataset. The evaluation metrics included precision, recall, F1-score, and the confusion matrix. These metrics provide comprehensive insights into the model's ability to accurately classify different types of cyberattacks.

The confusion matrix offers a detailed view of the true positive, false positive, true negative, and false negative predictions, enabling us to visualize the model's performance across different classes. By plotting the confusion matrix, we can easily identify any specific attack types that the model struggles with, thereby guiding future improvements.

Conclusion and Final Analysis

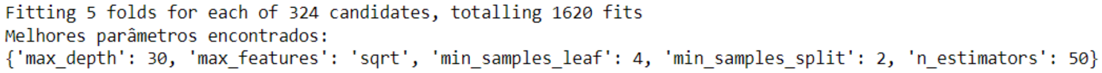
Despite our approach and the implementation of a robust machine learning pipeline, the final results of our model were not very promising. The optimized model achieved an F1-Score of 0.3336, Precision of 0.3344, and Recall of 0.3342 (Figura 7). The detailed breakdown showed that the accuracy for each attack type was consistently around 33%, which is not spectacular.
However, we consider that our methodology followed standard practices for this type of problem, given our available resources and time. We focused on practical and cost-effective techniques without delving into overly complex or high-cost solutions. We conducted comprehensive Exploratory Data Analysis (EDA), handled missing data effectively, applied appropriate techniques for categorical variable encoding and data normalization, performed rigorous feature selection and hyperparameter tuning using Grid Search with Cross-Validation to ensure the best possible model performance.

Furthermore, upon reviewing similar analyses conducted by other users who accessed the same dataset from Kaggle, we observed that our results were either on par with or better than most. This consistency across multiple independent analyses suggests that the limitation might lie more with the dataset itself rather than our methodology. The size of the dataset, along with other intrinsic properties such as the quality and variability of the data, likely contributed to the constrained performance of the models. In machine learning, the quantity and quality of the data are critical factors that significantly influence the performance of predictive models. Smaller datasets often lack the diversity and representativeness needed to capture complex patterns, leading to limited model accuracy. This scenario underscores one of the inherent challenges and realities of machine learning: achieving high performance is not always straightforward or guaranteed.

Factors such as data sparsity, noise, and the presence of subtle, intricate patterns that require larger and more diverse datasets can all hinder the model's ability to generalize well. Additionally, the effectiveness of machine learning models heavily depends on the richness of the features and the underlying relationships within the data. When these elements are suboptimal, even the most sophisticated algorithms can struggle to perform well.

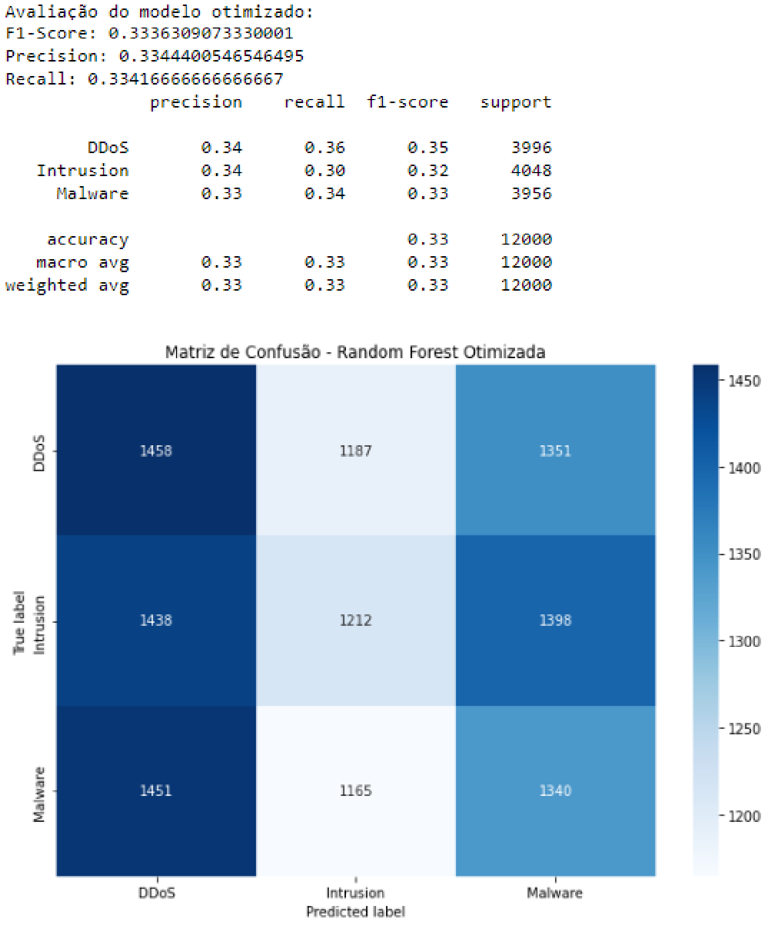
This project exemplifies the importance of realistic expectations in machine learning. It highlights that while machine learning has powerful potential, it also has limitations. Success often requires not only robust methodologies and algorithms but also high-quality, abundant data. As such, our modest results, reflect a comprehensive approach given the constraints of our dataset, reinforcing that continuous improvement in data collection and preprocessing is essential for achieving better performance in future efforts.

In conclusion, while our final results did not achieve high metrics, they are reflective of the challenges inherent in the dataset. We believe that our approach was appropriate for the problem at hand. The insights and techniques applied in this project are foundational and can be scaled to larger datasets, which would likely produce better performance and more significant results.



```
Fitting 5 folds for each of 324 candidates, totalling 1620 fits
Melhores parâmetros encontrados:
{'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}
```

Figura 6 - GridSearch Hyperparameter Tuning



```
Avaliação do modelo otimizado:
F1-Score: 0.3336309073330001
Precision: 0.3344400546546495
Recall: 0.3341666666666667
              precision    recall  f1-score   support

        DDoS       0.34      0.36      0.35      3996
   Intrusion       0.34      0.30      0.32      4048
     Malware       0.33      0.34      0.33      3956

    accuracy                           0.33     12000
   macro avg       0.33      0.33      0.33     12000
weighted avg       0.33      0.33      0.33     12000
```

Figura 7 - Performance on test data of the best Random Forest Classifier

Nota: Na elaboração deste documento, foi utilizada a ferramenta ChatGPT como recurso auxiliar para a revisão e correção de texto.