Analysis of Necessary Features to Detect Malicious Cyber Attacks

Dimitris Kalafatis, Teymur Hajiyev

1. Importance of cybersecurity and ability to detect cyber threats

Cybersecurity has become a cornerstone of modern organizational resilience, safeguarding critical systems, sensitive data, and digital assets from an ever-evolving array of threats. As businesses accelerate digital transformation and adopt new technologies, the risks associated with cybercrime have grown exponentially, making robust cybersecurity measures indispensable.

Rising Costs and Investments in Cybersecurity
The financial implications of cyber threats drive organizations to allocate unprecedented resources toward cybersecurity. Global spending on cybersecurity is projected to grow by 12.2% in 2025, reaching $377 billion by 2028. This surge reflects the increasing sophistication of cyberattacks and the urgency for businesses to fortify their defenses against ransomware, phishing, and other malicious activities. Similarly, Gartner forecasts a 15% rise in global information security spending in 2025, totaling $212 billion. This growth is fueled by heightened threat environments, cloud adoption, and the need for advanced endpoint protection solutions.

Segment	2023 Spending	2023 Growth (%)	2024 Spending	2024 Growth (%)	2025 Spending	2025 Growth (%)
Security Software	76,574	13.6	87,481	14.2	100,692	15.1
Security Services	65,556	13.6	74,478	13.6	86,073	15.6
Network Security	19,985	6.2	21,912	9.6	24,787	13.1
Total	**162,115**	**12.7**	**183,872**	**13.4**	**211,552**	**15.1**

Table 1.1: Growth of Cybersecurity Spending

These figures underscore the critical role cybersecurity plays in preventing financial losses and maintaining trust and operational continuity.

As these threats evolve, so must our analyses of them. By both creating predictive models to cdetermine cyber attacks and by critically analyzing the features important to determining the pathogenicity of an attack, we can not only create an accurate model and dataset, but also provide a robust workflow capable of creating helpful models far into the future.

2. Creating predictive models to determine pathogenicity of requests

2.1: Datasets and Data Augmentation

To train and evaluate our models, we use the 2023 CiC IoT Dataset, a dataset with 39 unique features useful for classifying various cybersecurity attacks. For our initial datasets, our goal was to determine whether the attack was malicious or benign; for this reason, anything not classified as benign was modified and simply labeled as malicious.

Furthermore, the dataset has a very uneven distribution of benign to malicious labels, approximately 1:20. For this reason, we randomly selected data points to create two other datasets, our 1:1 dataset, and our “realistic” dataset, which is 100:1 benign to malicious. While the “realistic” dataset better conveys the fact that the vast majority of requests are benign, the figures in this report will be made using the 1:1 dataset unless otherwise specified. This is because the 1:1 dataset more clearly conveys the model’s general characteristics, and if either dataset leads to significantly different/unexpected results, they will also be shown. All figures created on all datasets can always be found in the supplementary figures.

For the majority of the following models (unless otherwise specified), we will be using a train : validation : test ratio of 0.7 : 0.15 : 0.15. The

validation dataset will help us tune hyperparameters and choose the best-performing model, and the final test dataset will give us a completely unbiased metric of each model's performance.

2.2: Linear Regression Model

One simple approach to create a predictive model is to simply create a linear regression model, assigning 1 to malicious attacks and 0 to benign ones. By fitting a linear regression model with a cutoff at 0.5, we can assign values of >0.5 to malicious and <0.5 to benign. Evaluating our model:

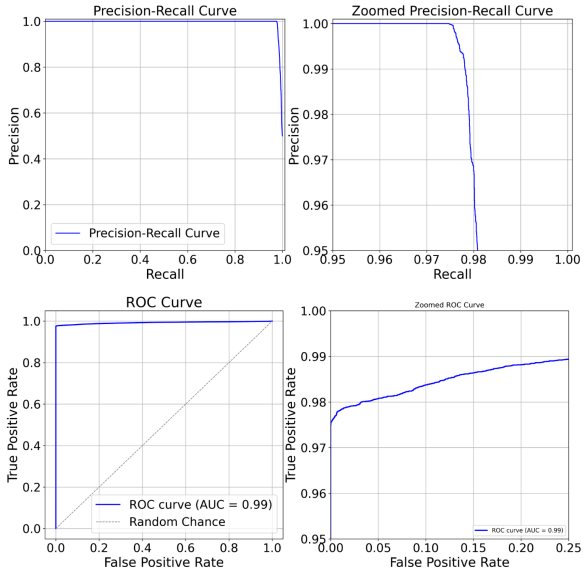

Figure 2.2.1: Linear regression model evaluation over time

Metric	Score
Accuracy	0.9871
Precision	1.0000
Recall	0.9743
F1 Score	0.9870

Table 2.2.2: Analysis of linear regression model performance on 1:1 dataset

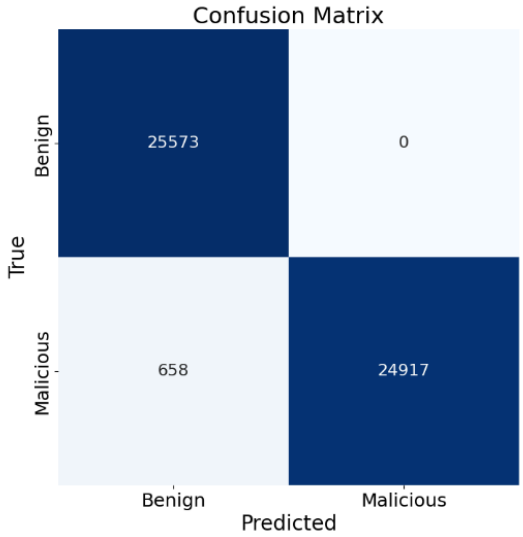

Figure 2.2.3: The confusion matrix for linear regression

These plots show that even a simple linear regression model is able to achieve remarkably high accuracy on this dataset, showing that it is fairly robust. However, while the model does seem accurate, in a world where thousands or millions of requests are sent per day, even a small improvement in the accuracy of a model can lead to significant ramifications, as even the 98.7% accuracy seen in this model can still lead to hundreds to thousands of misclassifications.

2.3: Logistic regression Model

A logistic regression is a good model for predicting boolean values. It will output values between 0 and 1, which are interpreted as probabilities. Similar to the approach we used in the linear regression part of the analysis, cases where the estimation is above 0.5 are classified as malicious, and the ones below 0.5 are classified as non-malicious.

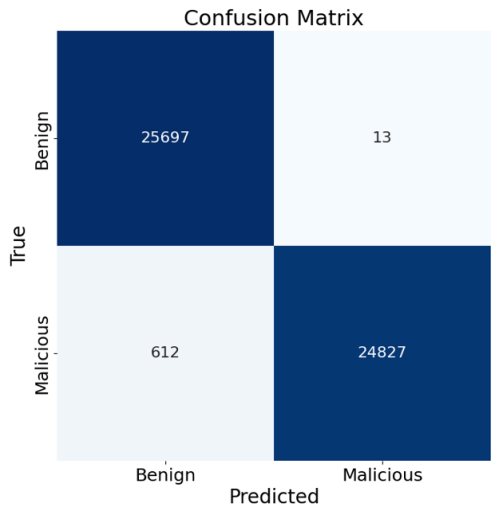

Figure 2.3.1: The confusion matrix for logistic regression

Metric	Score
Accuracy	0.9878
Precision	0.9995
Recall	0.9759
F1 Score	0.9876

Table 2.3.2: Performance metrics for the logistic regression model

2.4 K-nearest neighbors

The K-Nearest Neighbors (KNN) model classifies a test point by finding the n most similar points and using their labels to determine the classification. It will determine the test points label by seeing which labels appears the most in the nearest points.The number of neighbors, or n, can be customized. We decided that our goal would be to minimize the recall, as we want to avoid false negatives.

Neighbors	Recall
2	0.978538
3	0.979745
4	0.97807
5	0.978733
6	0.978265
7	0.978733
8	0.977953
9	0.978265
10	0.977798
11	0.978187
12	0.977642
13	0.977798
14	0.977525
15	0.977876

Table 2.4.1: Recall for different values of n/neighbors

The recall is maximized when the number of neighbors checked is 3. Therefore, we will use that value when evaluating the results of our KNN model.

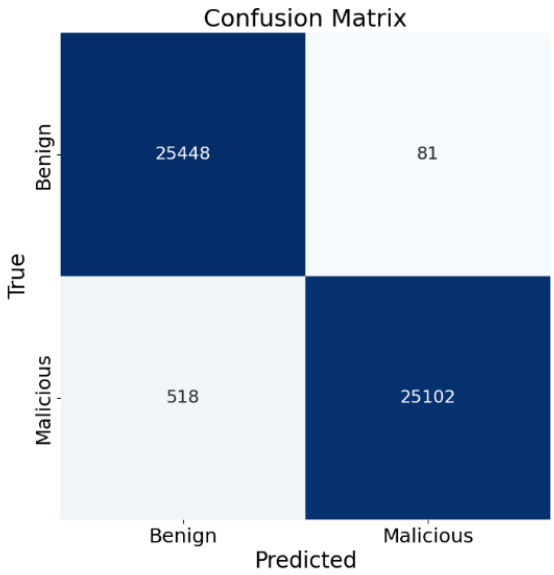

Figure 2.4.2: Confusion Matrix for KNN

Metric	Score
Accuracy	0.9883
Precision	0.9968
Recall	0.9798
F1 Score	0.9882

Table 2.4.3: Performance metrics for KNN model

2.5: Decision tree models

Decision trees are effective models for classification, since they repeatedly split based on certain feature characteristics. This means they are effective at learning non-linear relationships, but are also prone to overfitting if given too much depth.

We created two decision tree models, a Random Forest Analyzer, and an XGBoost Classifier. First, for the Random Forest Analyzer, we first needed to determine what depth would give the best results and would avoid overfitting. By using our validation dataset, we determined that the best depth for the Random Forest was approximately 20.

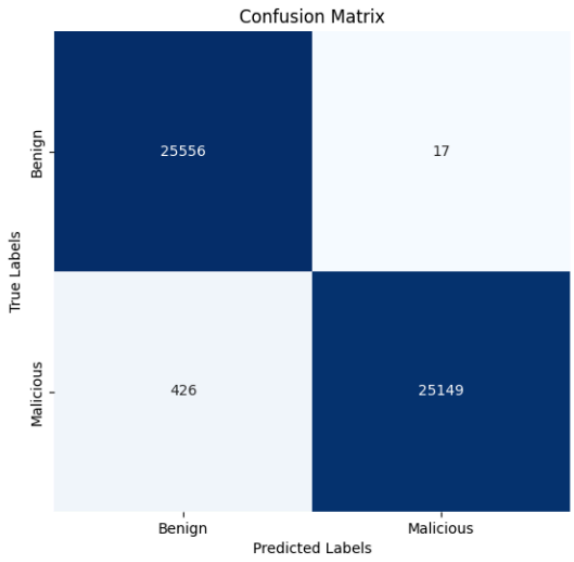

Figure 2.5.1: Confusion Matrix for Random Forest Analyzer

As seen in the confusion matrix, the Random Forest analyzer does significantly better than any of the previous models across all metrics.

Next, the XGBoost model was similarly tuned to the hyperparameters of n_estimators=100, subsample=0.8, colsample_bynode=0.3, scale_pos_weight=10.

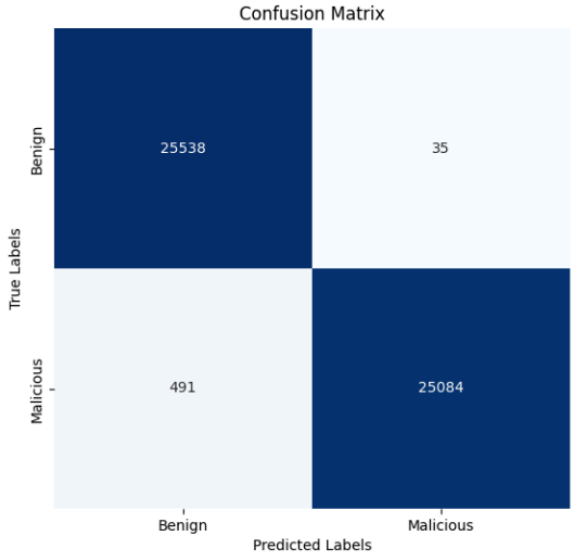

Figure 2.5.2: Confusion matrix for XGBoost Classifier

The XGBoost model does perform better than most of the other models, however, it is clear that the Random Forest Model still has fewer false positives and false negatives, making it a better model.

2.6: 1D-CNN

Convolutional Neural Networks (CNNs) normally are used for detecting features in images, due to using a learned kernel which is good for learning local information. We thought that we could apply a 1D kernel to the features in order to still learn some important features. Despite increasing the complexity with 2 kernels and 2 linear layers for 3000 epochs which is quite computationally expensive, the CNN still underperformed compared to the other models.

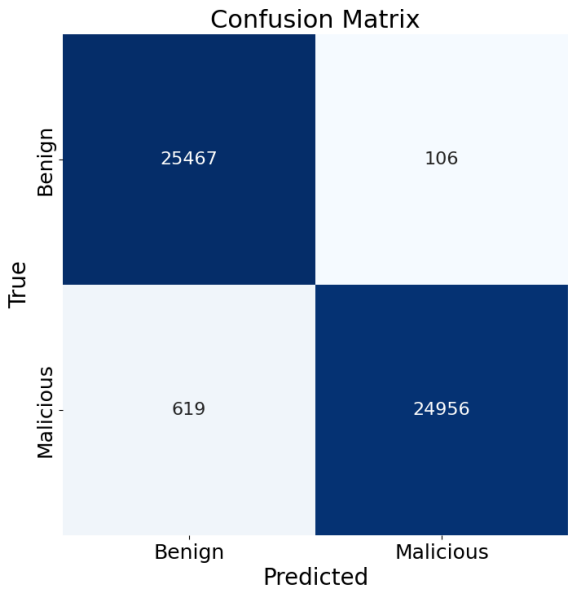

Figure 2.6.1: Confusion matrix for CNN

This shows that CNNs are not suited for tabular information, especially not for ones where the columns are unrelated and do not have learnable patterns.

3. Analysis of important features

While all of these models were fairly successful, one possible way to improve accuracy is to remove redundant features so the models are more capable of learning important information, and does not get confused by random noise.

3.1: Principal Component Analysis

One method of determining the importance of different features is with a Principal Component Analysis (PCA). PCA is a method which attempts to reduces the dimension of features by creating 2 ‘summary’ axes which are linear combinations of other features.

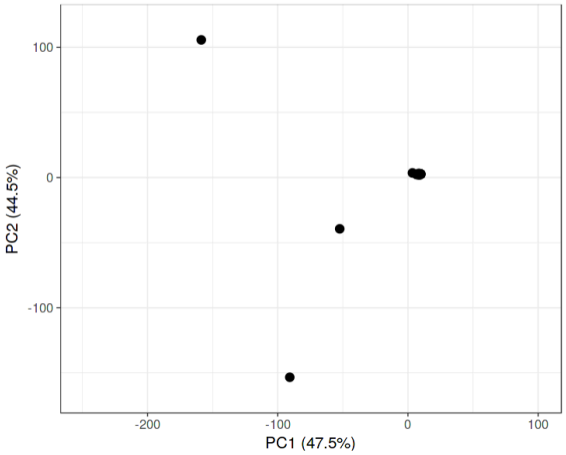

Figure 3.1.1: PCA Analysis of CiC IoT Features

By performing PCA on the CiC IoT 2023 dataset, we can clearly see that the large number of features can be effectively summarized into 2, as PC1 accounts for 47.5% of the variance, and PC2 is responsible for 44.5% of the variance. This means that 92% of all of the variance

contained in all features can be explained by two ‘summary’ components. We can take this one step further and actually cluster each feature based on the significance (z-score) of the change for different types of attacks compared to the average.

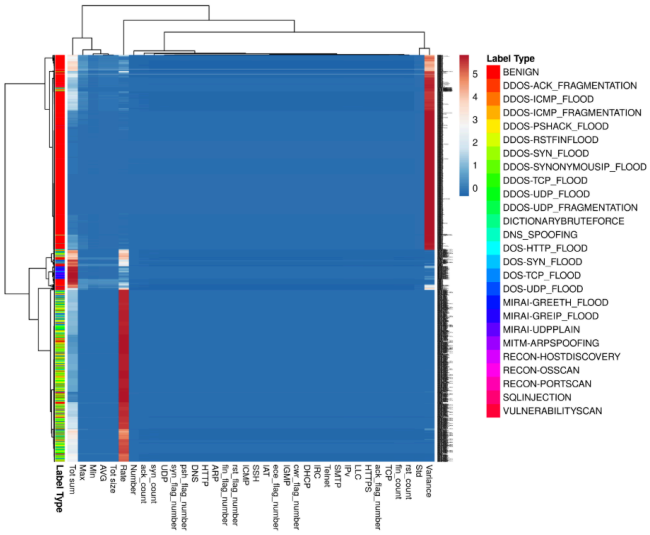

Figure 3.1.2: Heatmap analysis of clustering features based on correlation to attack type

By doing this, we clearly see that there are 3 features which have very significant changes for different types of attacks, while other features have very insignificant changes statistically, meaning that they are not good indicators of attack type or classifying whether a request is an attack or not.

3.2: Linear regression analysis

By using the models from section 2, we can evaluate them to see how important different features are in them. For example, in the linear regression model, each feature's importance can be inferred from the magnitude of its corresponding weight in it's weight vector (the 'beta value') as larger weights indicate a stronger influence of that feature on the predicted outcome.

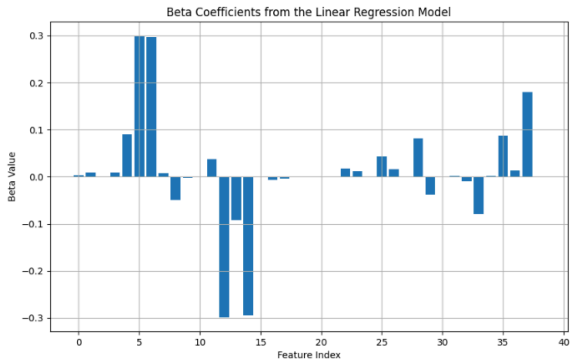

Figure 3.2.1: Analysis of feature importances from weight magnitude

As seen in the plot, 5 of the 39 features have significantly higher weight than the rest of the features. This once again shows how the model disregards the vast majority of given features.

3.3: Logistic Regression

Similar to the linear regression, the logistic regression features also can be evaluated by the magnitude of the betas. Since logistic regression is better suited for classification, we described the top 10 important features according to the logistic regression model.

Feature	Beta value
Number	6.618623
syn_flag_number	2.354904
Max	-1.9789
Std	1.712137
TCP	-1.68174
Tot sum	1.674802
rst_flag_number	1.325738
fin_flag_number	1.241529
Variance	-1.18438
HTTPS	-0.92677

Table 3.3.1: Feature importances of the logistic regression model

Again, this table shows that there are maybe around 5 features that far outweigh the contributions made by the other features.

3.4: Decision Trees

As shown in section 2, decision trees were by far the most effective model. Thus, we looked at the feature importances for each feature in each of our decision tree models.

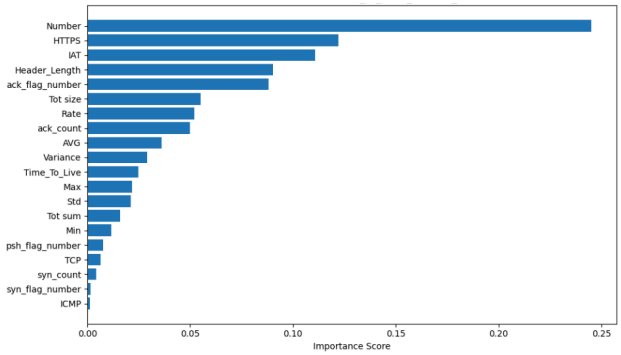

Figure 3.4.1: Feature importances in Random Forest Analyzer model

By first looking at the Random Forest Analyzer, we see the same pattern that we saw in the regression models, where just a handful of features bear the majority of the importance.

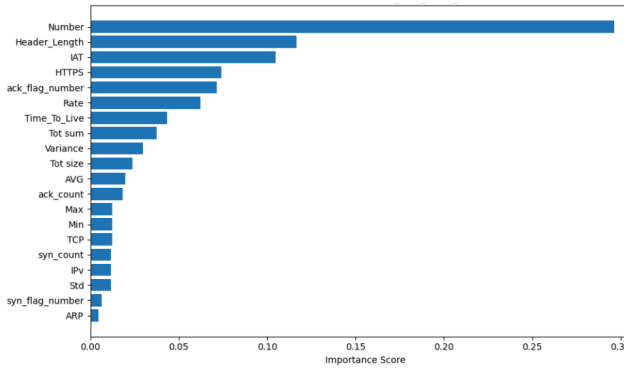

Figure 3.4.2: Feature importances in XGBoost Classifier model

Similarly, in the XGBoost model, the feature specificity seems to be even more exasperated, but the same trend remains, where the majority of features play little to no role in classification.

4. Different datasets include independent features that still lead to robust predictions

4.1: Geographical analysis

To analyze whether geographical location was an important feature, we found an additional dataset that included IP addresses. We converted each IP address to latitude and longitude and created a heatmap to see if there are any trends in the data.

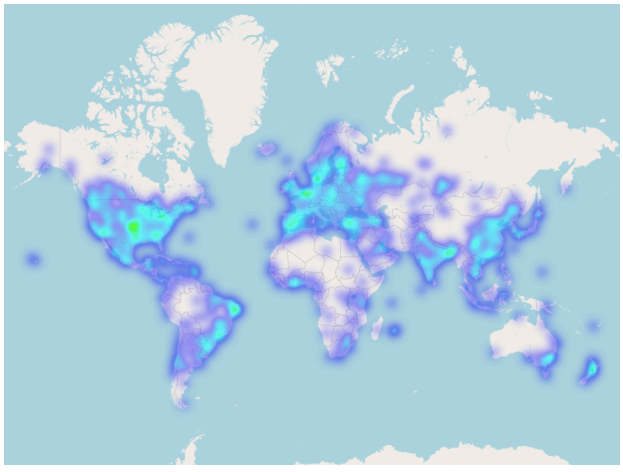

Figure 4.1.1: Heatmap for destination of cyberattacks

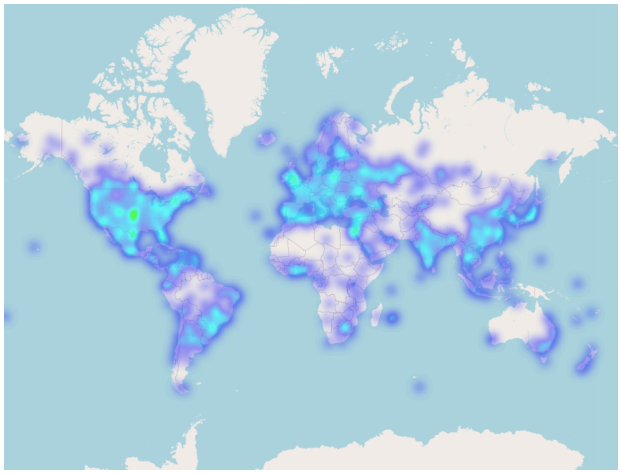

Figure 4.1.2: Heatmap for sources/origins of cyberattacks

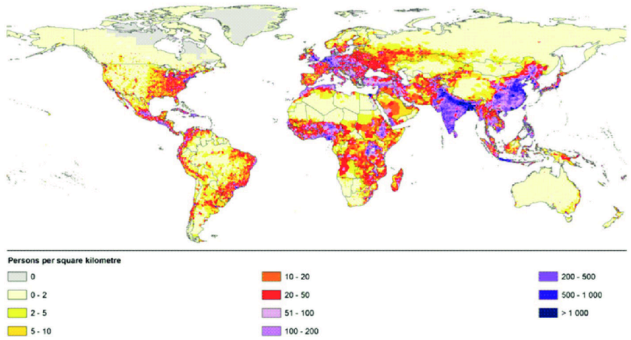

Figure 4.1.3: Population density map from ResearchGate

The heatmaps for the sources and destinations of cyber attack have a strong correlation to the population density map of the world. It is clear that most of the cyber attacks are to and from the major population centers, with the notable exception being the midwest. This is likely because the midwest is very industrialized while also being fairly rural. This leads to a lot of small businesses that don't invest as much into advanced cyber security and rely more on legacy systems, making them more susceptible to cyber attacks. This also creates a lot of opportunities for cyber hackers in the area, leading to a lot of those cyber attacks coming from within the state/s.

4.2: Looking for more features

We found an additional dataset and trained a Decision Tree model on it. The results were similar in accuracy to those from the main dataset used for this report.

Metric	Score
Accuracy	0.9295
Precision	0.9477
Recall	0.9413
F1 Score	0.9445

Table 4.2.1: Metric for the decision tree model on the additional dataset

The main goal of this was to see if there are any features that our dataset did not have that could have been valuable.

Feature	Importance
sttl	0.687375
synack	0.073467
ct_dst_src_ltm	0.069917
smean	0.069897
id	0.045579

Table 4.2.2: Feature importance for the decision tree moel on the additional dataset

This suggests that STTL, which is the time to live in seconds is by far the most important feature. It is the time a packet is allowed to stay in the network and it seems to results in very accurate predictions, suggesting that it could be a very relevant feature for the dataset that we were provided if any additions were to be made.

5. Conclusion and implications

All the models we're tested generated high scores across all metrics, but the model most fit for dealing with cyber attacks would be the decision trees model. While the error is similar across all models, false negatives are the most dangerous mistake to make, meaning that the model that does a good job minimizing the false negatives is the safest one to use.

Through the investigations of additional datasets, we found that geographical location is not a significant feature for a dataset to have, as area with high population have higher traffic and therefore highest number of attacks targeting and originating form the area. The time to live metric, on the other hand, would be a very valuable. It helped create a very high performing decision tree model discussed in Section 4.2 while having an importance value of 0.687. Adding it to the dataset provided to us originally for analysis could substantially improve the performance of models detecting cyber attacks. While we proved that the data on cyber attacks helps us see through some trends that help identify malicious activity, it is important to recognize that cybersecurity is a fast evolving field and the relevant features could change in the future. It is important to analyze newer cyber attacks and make proper adjustments to keep the digital world safe.

6. Supplementary Code and Figures

Our GitHub Repository can be found at https://github.com/dimi1729/Cyberwise_MM. All of our supplementary figures are in the plots/ directory inside of the repository. Information about running the code can be found in the README or email dimi@tamu.edu or thajiyev@tamu.edu.

References

Adilson Tadeu Basquerote. (2024). *Perspectivas de las ciencias sociales aplicadas: reflexiones sobre la sociedad y el cambio*. Yegorov, Yuri. https://doi.org/10.22533/at.ed.700241005

Incribo. (2024). *Cyber Security Attacks*. Kaggle.com. https://www.kaggle.com/datasets/teamincribo/cyber-security-attacks?select=cybersecurity_attacks.csv

IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Www.unb.ca. https://www.unb.ca/cic/datasets/iotdataset-2023.html

Moustafa, N. (2021, June 2). *The UNSW-NB15 Dataset | UNSW Research*. Research.unsw.edu.au. https://research.unsw.edu.au/projects/unsw-nb15-dataset

Versace, C. (2024, October 10). *We're Lifting Price Targets for Three Holdings to Reflect the Latest Data*. TheStreet Pro. https://pro.thestreet.com/portfolio/lifting-price-targets-for-three-holdings-to-reflect-the-latest-data

Metsalu, T., & Vilo, J. (2015). ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap. Nucleic acids research, 43(W1), W566–W570. https://doi.org/10.1093/nar/gkv468