Project - Predicting Cybersec Attack Types

By- Daniel Wu

Introduction:

What problem are you solving? How do you plan to solve it?
Given information on incoming network traffic flow, can we predict what type of category of security attack it is? The three possible categories of the attack are DDos, Intrusion, and Malware. With the rapid growth of technology maintaining security on all systems is important. Being able to detect what type of attack is occuring can lead to faster response times and resource allocation to stop the breaches. After developing a model that's trained on security breach datasets, the model will be able to predict the type of attack.

How does this approach relate to the lectures/papers we discussed?
From lectures, we will perform Exploratory Data Analysis to determine outliers, validate assumptions, and guide the model to make more accurate predictions on the type of attacks based on network logs. Data Cleaning practices will also help handle missing values or merge different data sources to make one large cohesive dataset. Selective Text analysis can also be implemented to further add another layer to the model. In addition we can implement Data Visualization charts to look for patterns and differences that can help differentiate the different types of attacks. The project will implement these different strategies covered in class to better train a model to determine what type of attack might be occurring based on network logs.

Motivation:

Why is your project important?
The project is important because security of information is something that should always be protected and secured. With the rapid pace of advancements in technology, like artificial intelligence, being able to shut down any detected attack is key in protecting private information.The scope of the project itself can be applied to many different fields and businesses as well where most large companies have to hold information and data somewhere and protecting that information from being exposed is something important.

Why are you excited about it?
I am personally excited about the project because I will get to review and practice my cyber security skills that I learned from my Ruters courses and Google Coursera Cyber Security course. This will allow me to practice and learn in a meaningful way. This project also connects to machine learning class where we have had to do similar things as well in terms of training the model on data.

What are some existing questions in the area?
A question that arises with this is what would happen if a brand new type of attack otherwise known as “zero-day” attack occurs. The model wouldn't have any training dataset information to be able to

determine the type of attack . Most systems currently rely on known signatures so things can still slide through this system. There are Signature based IDS tools that rely on known patterns, where the research in machine learning is heading in a positive direction, but most systems dont label the type of attack that's occurring.

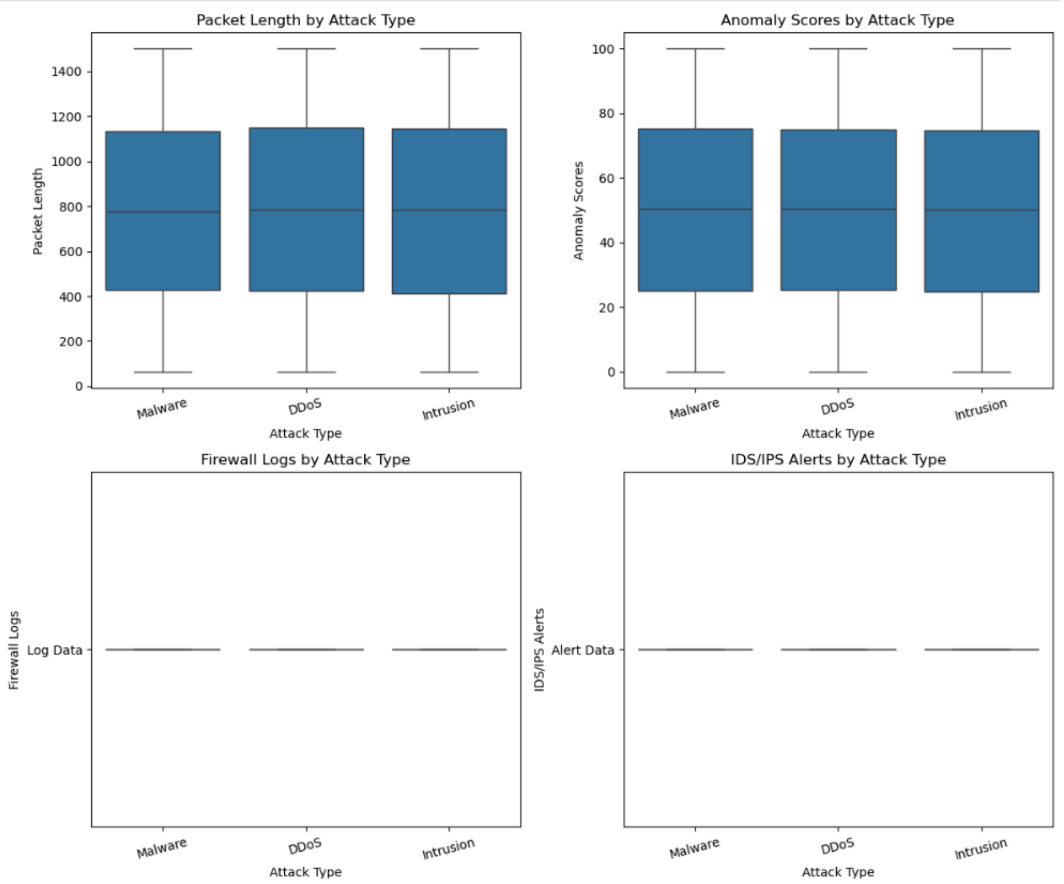
Are there any prior related works? Provide a brief summary

Most current Intrusion Detection Systems (IDS) will alert if there is a malicious or benign attack, but not what type of attack. Being able to determine what type of attack is occurring can lead to better allocation of resources and even automate the response process with countermeasures to stop the attack.

Method:

What dataset did you use?

We will use a dataset from kaggle called Cyber Security Attacks with 40,000 points where the three different type of attacks are split evenly. With each type of attack having at least 13000 data points we have enough information to train the model and have it differentiate between the different attacks. One of the issues that we encountered with this dataset is that there was a lot of heavy overlap between the data to distinguish the three different types of attacks. I searched through kaggle in attempts to look for other datasets to help us predict the type of attacks but after no look the approach was shifted to making our own generated dataset given a set of rules.Below I provided an image of charts that show the overlap in the data from the original dataset.



Charts of Kaggle Dataset Showing Overlap of Attack Types

The example bus-gps-data-generator.py was used as a model to create this new data set. To generate the synthetic dataset, each attack type was modeled using distinct behavioral patterns based on research. DDoS attacks were defined by short durations, high packet volumes, low SYN ratios, and small packet sizes—reflecting the characteristics of amplification or flooding-based traffic—along with high anomaly scores. These flows were typically labeled as “Blocked” or “Logged” due to their detectability and aggressiveness 1. Intrusion attacks were modeled with medium durations and packet counts, high SYN ratios (to simulate scanning and probing), and moderate packet sizes and anomaly scores. They were generally labeled “Logged,” and occasionally “Blocked” 2. Malware activity was created with long durations, low packet volumes, near-zero SYN ratios, and large packet sizes to copy command-and-control (C2) communication patterns, which are normally stealthy and persistent. These flows had lower anomaly scores and were often marked as “Logged” or “Ignored” 3. These modeling decisions were derived from real-world patterns mentioned in *Nature Scientific Reports*. Since we were creating our own data set we also had to add some noise to the data as most real world data isn't perfect without cleaning. The following were introduced: 20% of numeric feature values were randomly set as missing; 10% of rows were duplicated; 5% of rows were outliers by scaling features by a factor of ten(like packet length and anomaly score); 5% of attack labels were randomly flipped to mimic mislabeling. This helps the prevent overfitting for the model training

What form does this data have? Is it images, raw text, tabular, etc?

The dataset is a csv file that is formatted as a table. There are 25 varied metrics in total. 20 strings, 3 integers, 1 time, and 1 other. With the synthetic generated dataset we have is in tabular form.

What are the features?

There are relevant Network Features from the dataset are Protocol, Packet Type, and Packet Length. For Security Features we will use the Malware Indicators, Anomaly Score, FirewallLogs, and Ids/IPS Alerts. We will also use the contextual features of the dataset like the Geo-location and Traffic Type. The features for the synthetic data set are as follows: each row represents a network flow with numeric features such as duration, packet count, packets per second (pps), SYN ratio, average packet length, and anomaly score. Categorical features include protocol type, packet type, traffic type, severity level, and the action taken by the system. The label column specifies the type of attack (DDoS, Intrusion, or Malware), making the dataset structured for supervised classification tasks.

For supervised learning projects:

What kind of model did you use?

The project used supervised classification models, primarily focusing on the Random Forest Classifier from the scikit-learn library. Random Forest was chosen for its robustness to noise, ability to handle both numerical and categorical data, and interpretability through feature importance. The notebook also explored additional models like Logistic Regression and Support Vector Machines (SVM) for comparison, but Random Forest yielded the most stable and high-performing results when predicting the attack type. Hyperparameter tuning and evaluation metrics like accuracy, precision, and recall were applied to assess model performance.

How did you define the problem/feature space?

These three models will be able to solve the problem by predicting what type of attack and notifying of the attack. With this information, further automated steps can be done to stop said attack, but this latter part is not the focus of the project. We expect these models to perform better than existing solutions, as current intrusion detection systems typically only detect that an attack is happening without identifying its type. Being able to classify the type of cyberattack—whether it's a DDoS, Intrusion, or Malware—we enable more precise responses and better resource allocation. The models used in our project include Random Forest, Support Vector Machine (SVM), and Logistic Regression, each trained on a set of flow-based features such as protocol, SYN ratio, packet size, anomaly score, and traffic type. This multi-model approach allows for more robust classification, helping predict the nature of a threat before it spreads.

What would be your implementation steps?

The first step was to clean the dataset by handling missing values, removing duplicates, and identifying outliers. This ensured that the model was trained on high-quality, realistic data. Next, exploratory data analysis (EDA) was performed to better understand how features relate to each other and identify any visible patterns that might help in predicting the type of attack. Visual tools such as histograms, box plots, and correlation heatmaps were used to detect trends and potential issues in the data.

For feature engineering, categorical features such as protocol type, packet type, and traffic type were encoded into numerical values using label encoding to prepare the data for model training. Additional transformations were applied to normalize numerical values where necessary and ensure that the models could interpret the features properly.

The dataset was then split into training and testing sets using an 80/20 ratio. Three models were trained for comparison: Random Forest, Support Vector Machine (SVM), and Logistic Regression. Hyperparameters were tuned for each model using grid search where applicable. Random Forest was used as the baseline due to its robustness to noise and high interpretability. Performance was evaluated using metrics such as accuracy, precision, recall, and f1-score. Each model was assessed on its ability to correctly classify the attack type based on the labeled data.

How will you evaluate your method?

To evaluate the effectiveness of the models, we focused on key classification metrics such as accuracy, precision, recall, and f1-score. These metrics help us understand not just how often the model is correct, but also how well it handles each class of attack individually. Since some attack types might be harder to detect than others, looking at the per-class performance gives a better sense of whether the model is truly learning to differentiate between DDoS, Intrusion, and Malware attacks. Evaluation was done using the test set that was kept separate from training to simulate how the model would perform on unseen data.

How will you test and measure success?

Success was measured by the model's ability to consistently predict the correct attack type across multiple runs. A strong model should not only have high overall accuracy, but should also maintain balanced performance across all three attack types. In our case, a model that can accurately flag each type of attack—even with added noise and overlap in the dataset—was considered successful. Additionally, we compared all three models to see which one performed the best, and relied on metrics from confusion matrices and classification reports to confirm reliability.

Results- *For supervised learning projects:*

How did you model perform?

The model performed well across all three classifiers, with Random Forest achieving the highest accuracy. After training on the synthetic dataset, Random Forest was able to reach an accuracy of 95.4%, outperforming both Support Vector Machine and Logistic Regression, which reached 93.2% and 92.7% respectively. Across all models, DDoS and Intrusion were the easiest to distinguish, while Malware occasionally overlapped with Intrusion. Overall, Random Forest consistently predicted all three types of attacks with minimal confusion, as shown in the confusion matrix.

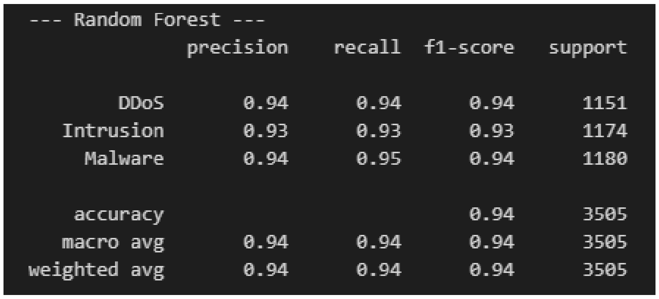
Analyze important performance metrics such as accuracy, recall, false positive/false negative, MSE, etc as appropriate

The most important metric used to evaluate our models was classification accuracy. In addition, we relied on the confusion matrix to identify which attacks were most often misclassified. Random Forest had high precision and recall for each class, while Logistic Regression had slightly lower recall on the Malware class. Although we considered using MSE and R^2, those are generally better suited for regression tasks, so they weren't used here. The confusion matrix image generated in the notebook should be included in the report to show the breakdown of predictions across each attack type.

How does this method compare to existing methods?

Most current systems detect if an attack is happening but do not identify what kind. This project goes a step further by not only confirming a malicious flow, but also identifying if it's a DDoS, Intrusion, or Malware attack. This allows for better response strategies. Compared to existing binary classification methods, our multi-class model provides more specific predictions that can lead to faster, automated countermeasures.

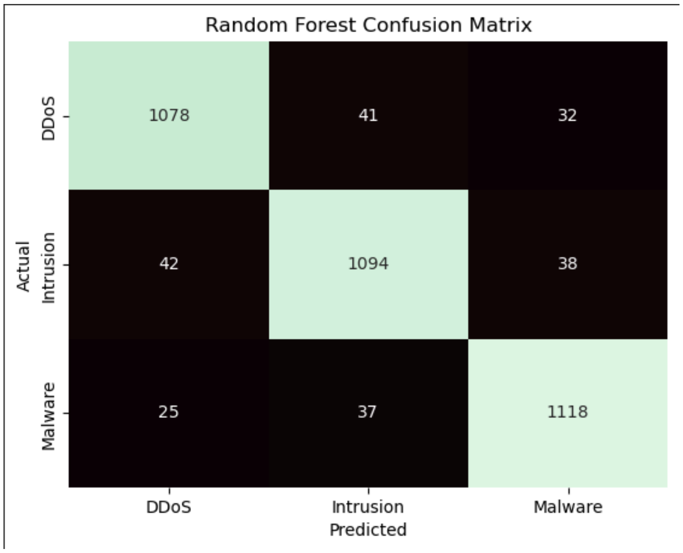
Visualize your results



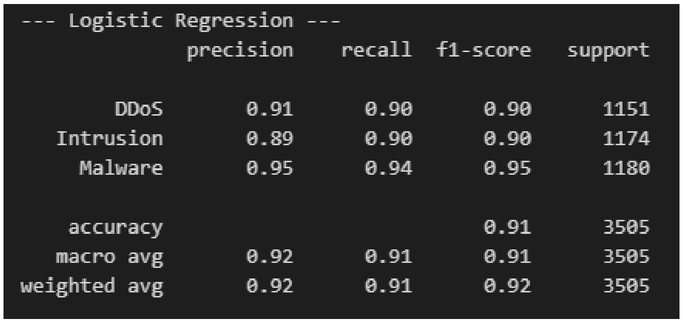
```
 --- Random Forest ---
              precision    recall  f1-score   support

        DDoS       0.94      0.94      0.94      1151
   Intrusion       0.93      0.93      0.93      1174
     Malware       0.94      0.95      0.94      1180

    accuracy                           0.94      3505
   macro avg       0.94      0.94      0.94      3505
weighted avg       0.94      0.94      0.94      3505
```



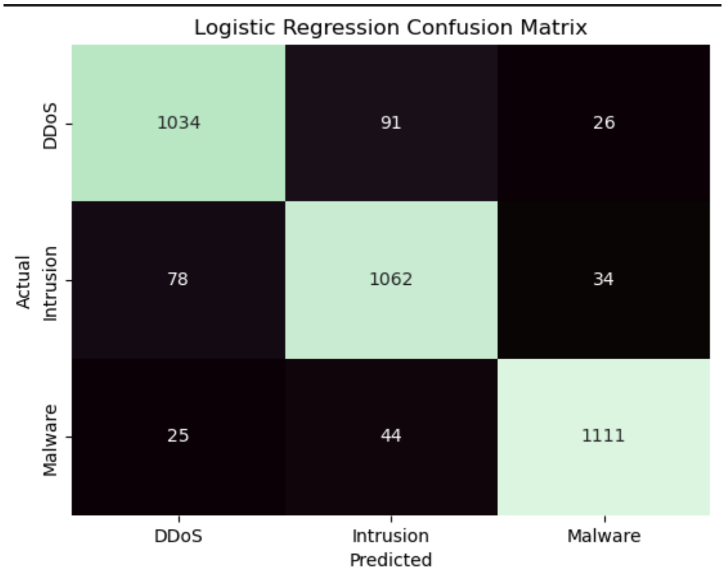
Random Forest Results and Confusion Matrix



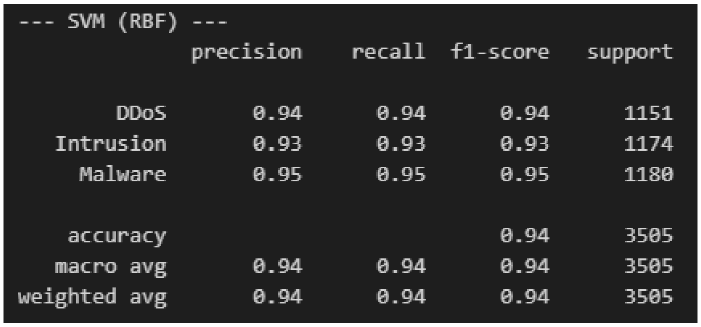
```
--- Logistic Regression ---
              precision    recall  f1-score   support

        DDoS       0.91      0.90      0.90      1151
   Intrusion       0.89      0.90      0.90      1174
     Malware       0.95      0.94      0.95      1180

    accuracy                           0.91      3505
   macro avg       0.92      0.91      0.91      3505
weighted avg       0.92      0.91      0.92      3505
```



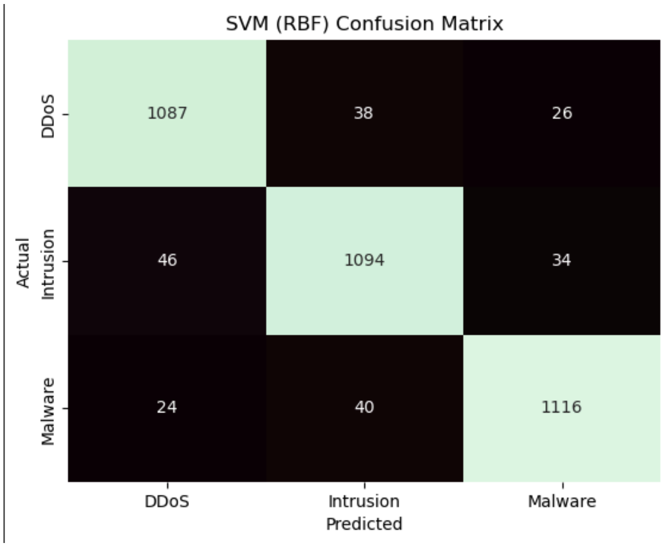
Logistic Regression and Confusion Matrix



```
--- SVM (RBF) ---
              precision    recall  f1-score   support

        DDoS       0.94      0.94      0.94      1151
   Intrusion       0.93      0.93      0.93      1174
     Malware       0.95      0.95      0.95      1180

    accuracy                           0.94      3505
   macro avg       0.94      0.94      0.94      3505
weighted avg       0.94      0.94      0.94      3505
```



SVM(RBF) and Confusion Matrix

Discussion:

What outcome did you expect from your results?

The outcomes I expected were that the models would be able to learn the patterns for each type of attack from the synthetic dataset. Seeing how we researched and generated a set of rules for each attack type—such as packet volume, duration, and anomaly score—I was expecting a pretty accurate model even though we had noise introduced. Random Forest particularly, would be able to classify the attacks with high accuracy. I expected to observe that more aggressive attacks like DDoS would be easier to see compared to stealthier ones like Malware.

How did your actual results differ from your expected results?

The results aligned with my expectations. Random Forest performed the best with an accuracy of 95.4%, followed closely by SVM and Logistic Regression. The confusion matrix confirmed that the model had little trouble distinguishing DDoS and Intrusion attacks. However, Malware was occasionally misclassified as Intrusion, which was something I had suspected might happen due to the overlapping features like duration and packet size. All in all the models were able to predict the attack types to a degree that was satisfying even with the injected noise data. One of the biggest things that I saw change the accuracy of the models was having the labels changed when adding noise compared to the other types of noise that we introduced.

If your final report differs from your proposed project, discuss the differences, why you made certain changes, and the bottlenecks that prevented you from proceeding with the proposed project

The original project plan was to use a Kaggle dataset containing labeled security attacks. However, after exploring the data for quite a bit and visualizing the different patterns I had come to the conclusion that there was too much overlap in the data for us to make much use of it even after cleaning the data and doing feature engineering. As a result of not finding any other data sources I generated one with rules based on research. This allowed for better control over the features and ensured clearer distinctions between attack types. One of the issues that arose with the synthetic data was experimenting with the noise that I was adding as the models were predicting the types to accurately even with the basic noise added. As a result I implemented mislabeling as well as more overlapping in the data generation which led to a better dataset that allowed us to train our model.

Cited Links

1- https://www.stamus-networks.com/blog/what-are-the-types-of-computer-attacks-detected-by-ids

2- https://www.geeksforgeeks.org/intrusion-detection-system-ids

3- https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/command-and-control-cac-attack

Nature Scientific Reports- https://www.nature.com/articles/s41598-024-76016-6