Name: Edwin Lin

Progress Report

Introduction

Cyber Attacks and Security was first found to be necessary in 1970 when network connectivity was established, and that was due to new advancements in establishing a network that showed a computer program can move through the network. Now, we have more security concerns to be looking out for in the modern advancements of Machine Learning and generative ai models as upcoming advancements as in the same way Bob showed a computer program creeper and traverse the network, it is found that you can interfere and disrupt the Machine model's learning phase to take control and access it for malicious intentions. This calls for a better defense, and just as the Machine learning model can be used for cyber attacks, we can use them for better defenses as well. Intrusion Detection Systems (IDSs) are essential for monitoring and identifying suspicious behaviors in digital environments. There are three main approaches to intrusion detection: misuse-based, anomaly-based, and hybrid techniques.
Misuse-based detection identifies attacks by comparing system activity to a database of known attack signatures. This works well for recognizing documented threats, but it struggles with new or evolving attacks. In contrast, this detection creates models of normal system behavior and flags any deviations as potential attacks. This method is more flexible and can detect zero-day attacks, but it often leads to high false positive rates. To address these issues, hybrid detection methods combine misuse and anomaly-based strategies, aiming to improve detection accuracy while reducing false alarms.

Problem Definition/Research progress

The main method that will be implemented will be a hybrid detection method on a UNSW_NB15 dataset since they reflect real world host based attacks as samples focusing on worms because of the complexities of Malware being developed and executed, so there is a demand to advance better defenses against these Malwares, especially with Machine Learning.

To start off with setting things with machine learning, I first have to clean the data I took from Kaggle that Host based attacks so that it is readable by my machine learning model. The reason is because machine learning only reads numbers. It doesn't directly read given word data. Naturally whenever we see a word in our dataset, we just convert it to integers. For example for types of attacks in the dataset we have Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms However, Machine Learning can't distinguish what each of the strings mean, so Let Fuzzers be 1, Analysis be 2, Backdoors be 3, etc so when it returns a data output in integers, we can

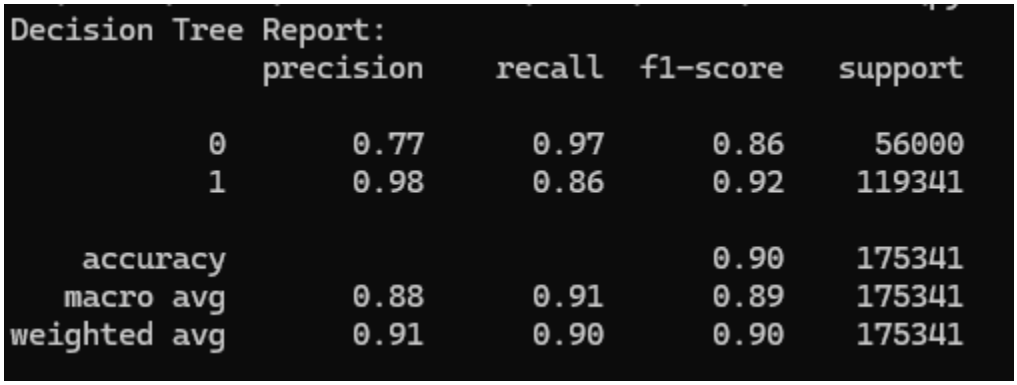
understand which intrusion occurs the most based on which integer the model spits back out. This step is called encoding. Next is preprocessing, which is cleaning the data into usable data. In the data, there are empty boxes ,as there may be no value associated with them. We can fill these data with 0s or Nan so that these empty boxes won't interfere with our machine learning model as not filling and cleaning those data may result in inaccuracy in testing.In machine learning, there is also another step where we train or split the dataset, usually 70% or 80% of the data for training the machine learning model and 80% for testing. Think of this idea as having flashcards. We give 80% of the data to the model to read the data and use that data to learn, allowing the model to read both the question and the answers. Then, we use the other 20% of the data model to test the model, allowing them to only read the question, but not showing them the other side of the flashcard. In this way, the model has to predict and choose their answer, and then we can make calculations to indicate how well they scored based on how many predictions are correct in correlation to the answers. Luckily for us, we didn't need to train/split the dataset as we are already provided with a training dataset, and a testing dataset. Therefore, we can use 100% of the training dataset for the model to learn, and 100% of the testing dataset for the model to predict. Lastly, I want to discuss our goals. We have already got a list of different attack distributions in the dataset, so the plan is to train the model to identify important features that have higher correlation to a certain attack, so we can detect which attack may be occurring with the features, and then test how well our model performed to identify these features. Here is the list of attacks that the model will be working with to learn fully encoded for model readability:

```
Attack Type Distribution in Training Set:
Normal            37000
Generic           18871
Exploits          11132
Fuzzers            6062
DoS                4089
Reconnaissance     3496
Analysis            677
Backdoor            583
Shellcode           378
Worms                44
Name: count, dtype: int64
```

Machine Learning Models Implementation

Here comes the fun part. Implementing the Machine Learning models. Our first model we are implementing is the Decision tree which follows Misuse based methods in our hybrid detection approach by labeling our known attacks to train it for supervised

learning.A supervised learning means that it takes in human inputs to train, and we use labels to guide the model into making the correct answer. In our model, our decision tree classifies if a result is just a normal network packet, or an attack using 0 for normal, and 1 for attack. It is pretty straight forward as the tree is just making a decision. This is how well our model performed:

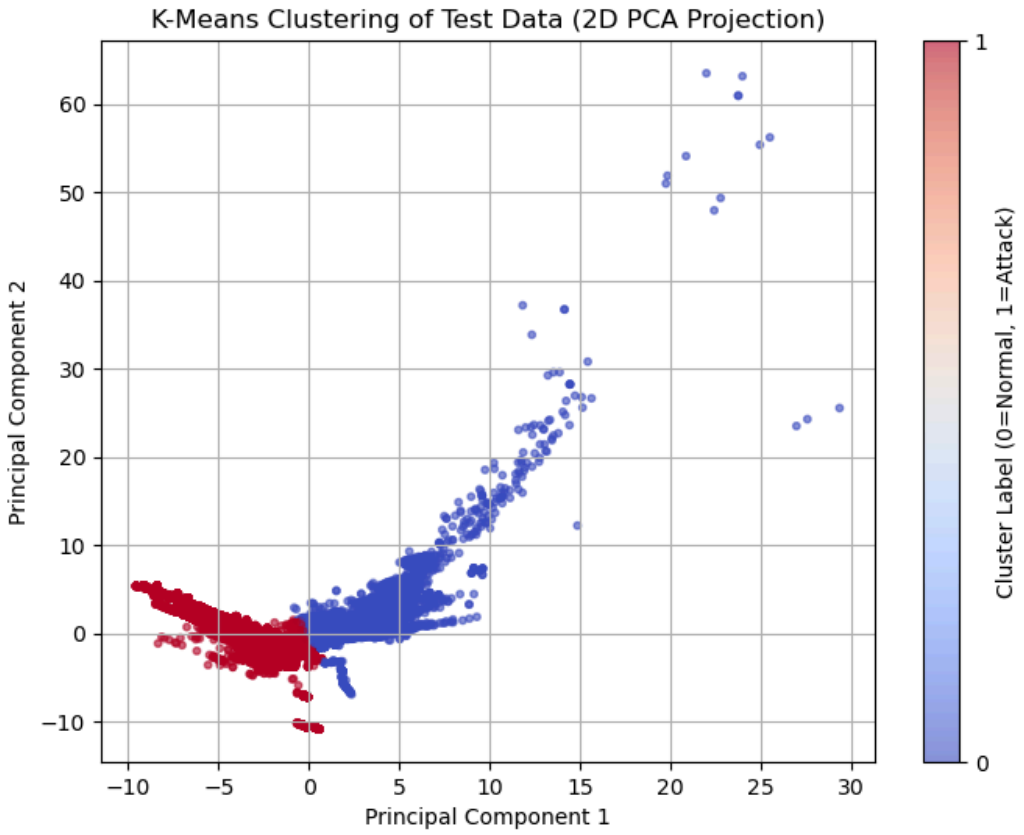


Decision Tree Report:

	precision	recall	f1-score	support
0	0.77	0.97	0.86	56000
1	0.98	0.86	0.92	119341
accuracy			0.90	175341
macro avg	0.88	0.91	0.89	175341
weighted avg	0.91	0.90	0.90	175341

Our next model is K-means clustering which follows Anomaly-based detection of our hybrid approach. Unlike decision trees, the K-means model is using unsupervised learning which means that the model learns by itself without labels to help guide it. This is to produce interesting results, not necessarily correct results. The idea is to find interesting patterns that can allow us to identify anomalies. For K-means, we first choose a k cluster, which we have 2, one for normal and one for attack. Then, we have a centroid which represents the point where the data is the group closest to the cluster. Around this centroid, we have random data points which the model predicts. Think of it like this, there is a billboard and you are blindfolded, and you throw a dart and it lands

around the general location. That is the idea of k-means clustering. Here are the results:



K-Means Report (Anomaly Detection):

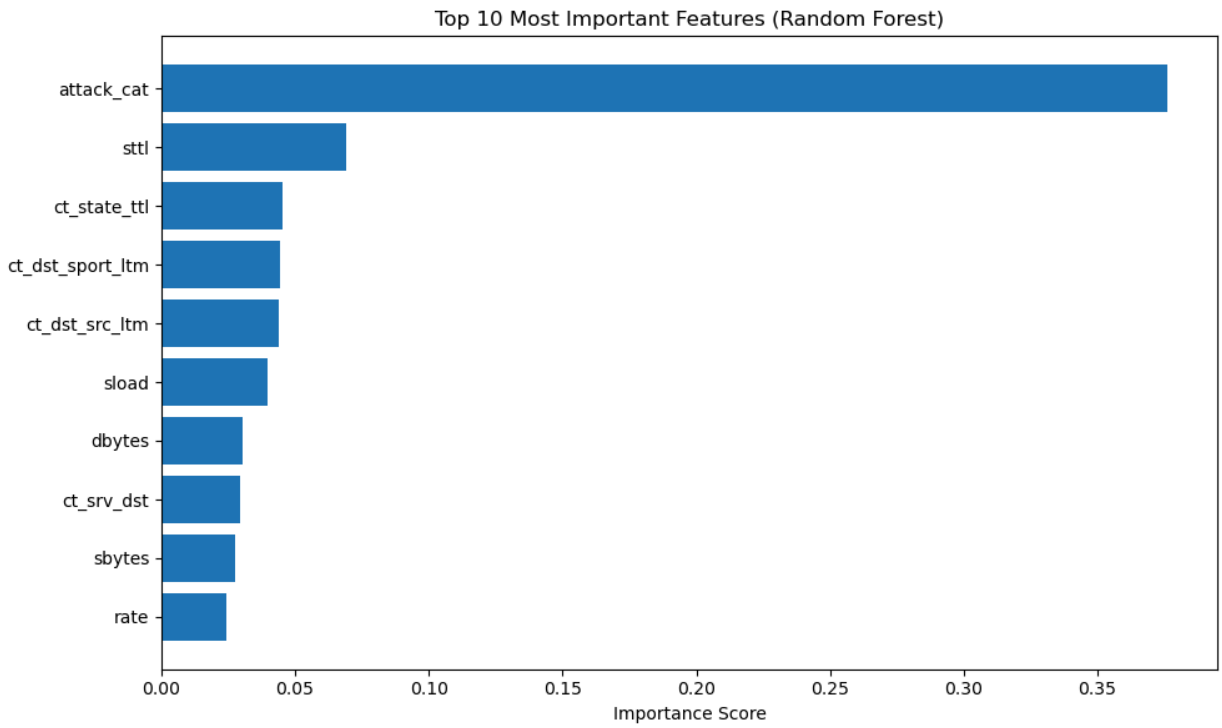
	precision	recall	f1-score	support
0	0.54	0.88	0.67	56000
1	0.92	0.65	0.76	119341
accuracy			0.72	175341
macro avg	0.73	0.77	0.72	175341
weighted avg	0.80	0.72	0.73	175341

Our last model is A Random Forest which is an ensemble machine learning algorithm for classification and regression. In summary, it is a combination of many decision trees that was used earlier to make a final prediction. Here is the result:

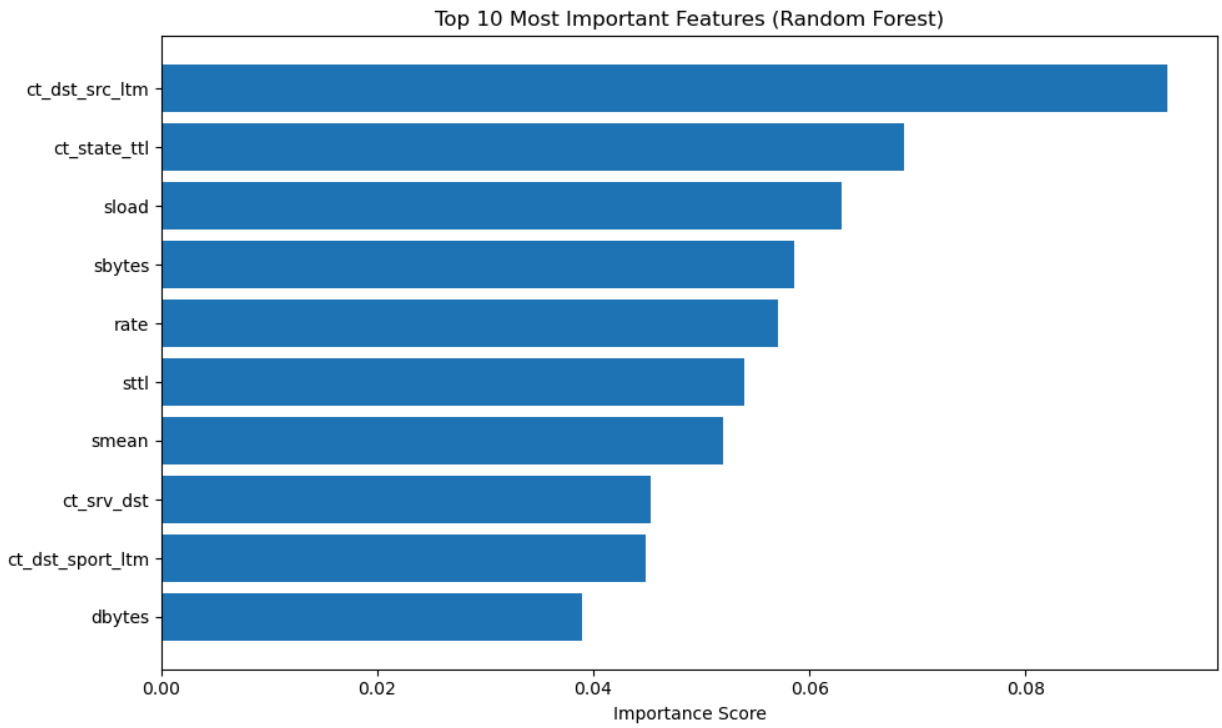
```
Random Forest Report (Hybrid Detection):
              precision    recall  f1-score   support

           0       0.77      0.98      0.86     56000
           1       0.99      0.86      0.92    119341

    accuracy                           0.90    175341
   macro avg       0.88      0.92      0.89    175341
weighted avg       0.92      0.90      0.90    175341
```

Performance analysis

Here is a breakdown of our data results, 0 represents normal and 1 represents attacks. Precision describes how often the model is correct, Recall tells us how many predictions the model makes were correct, F1- Score is average of both precision and recall, and support is the total number of actual data.Macro average treats all classes equally, which weighted is the overall performance. We use macro average to find any low performance mainly. Think of it like this, in classes in high school, students who take AP classes gain an extra 10% added into their final grade so a student can score a 70% but weighted grade ends up being 80%. The 70% raw grade is the Macro average while weighted average is the final grade of 80%. In comparison, our K-means performance was the worst out of the 3 which is to be expected as it was trying to find interesting patterns without any guidance of labels, but overall, our model performed pretty good. However, you will notice that it has a lower accuracy in predicting normal compared to finding attacks. That means that our model made a lot of false positives on a normal packet, marking it as false positive, but got 99% of the attacks correct. This is important because it shows that machine learning is good at finding attacks, but can generate false positives so human evaluation will be necessary before making a final decision. In addition, the reason why there is such a big difference in higher attack accuracy is because we conditioned the model to place more importance in finding attacks as missing an attack is detrimental so we rather have normal packets be mismarked rather than completely missing an attack. Here is the extracted features that show which features have highest importance:



attack_cat was used during training, but not included during real prediction, since it's essentially the label/category hence the prediction is giving 100% accuracy because the model is essentially cheating because of data leak.



Research Comparisons

To utilize these techniques effectively, machine learning provides powerful tools to automate the detection process. In this project, I'll use three specific ML models suited for each detection type: a Decision Tree classifier for misuse-based detection, K-Means clustering for anomaly-based detection, and an ensemble Random Forest model for hybrid detection.

Works Cited

EK_. “Awesome Machine Learning for Cyber Security.” *GitHub*, 13 Apr. 2023, github.com/jivoi/awesome-ml-for-cybersecurity.

GeeksforGeeks. “Intrusion Detection System Using Machine Learning Algorithms.” *GeeksforGeeks*, 24 Aug. 2020, www.geeksforgeeks.org/machine-learning/intrusion-detection-system-using-machine-learning-algorithms. Accessed 31 July 2025.

“UNSW_NB15.” *Www.kaggle.com*, www.kaggle.com/datasets/mrwellsdavid/unsw-nb15.