Intrusion Detection System

Dylan Muecke
ESET
Texas A&M University
College Station, Texas USA
dmuecke@tamu.edu

***Abstract*—This report outlines the process of using machine learning algorithms to create an intrusion detection system that identifies malicious packets from files collected by Wireshark.**

Keywords—machine learning, algorithm, malicious packets, intrusion detection system.

I. Introduction

Malicious packets create an extraordinary security risk for Internet communications. The internet operates by sending packets from one host to another through a network of routers and other nodes. These packets can sometimes be intercepted and modified or replaced to send alternate data to the destination. This alternate data can be anything ranging from modified values to destructive payloads. The goal of an intrusion detection system (IDS) is to recognize and isolate these malicious packets to prevent them from reaching their destination. Machine learning can be used to help improve the accuracy of these detection systems.

II. Data Conditioning

A. IoT-23 Dataset

The two types of attacks the IDS is attempting to identify are C&C and Torri attacks. These types of attacks are classified by the IoT-23 dataset which categorizes different types of malicious packets. Packets are labeled with C&C when an infected device is connected to a CC server. Connections to the unauthorized server can create major security risks. Torri packets can be identified by characteristics that resemble that of a Torri botnet. The training data used for these algorithms are packets with these labels that were verified to be malicious. Verification of the algorithms' effectiveness was accomplished by removing the labels and then testing the algorithms to see if they can predict the label based on the other data held within the packet. This training data used was provided with this assignment, and the information regarding the labeling convention was provided by [1].

B. Feature Removal

I choose to remove *Src_IP*, *Dst_IP*, *Timestamp*, *Sub_Cat*, and *Flow_ID* to prepare the data points for analysis. I originally choose *Src_IP*, *Dst_IP*, and *Timestamp* because these are all packet specific indicators that will not be helpful in identifying generalized malicious packets, as the data cannot be extrapolated to fit a pattern. I also removed the *Sub_Cat* feature because it clearly identifies the anomaly. The goal of this program is to identify the anomaly without this identifier therefore it should not be used to train the algorithms. After further testing I also decided to remove the *Flow_ID* feature because it was causing errors later in the program which successfully executed after it was removed.

C. Parameter Selection

In order to prepare the dataset to train the algorithm the *Label* feature was separated from the primary dataset (*data*) and placed in its owns parallel dataset (*y_data*). This allows the algorithm to test *data* and then compare its results with the *y_data* value of the same index. The first half of the data points from each set were used for training the algorithm and the remainder were reserved for testing it. This allows the algorithms to experience about the same amount of data as they will be testing.

III. Algorithms

The program used to create this IDS implemented two different machine learning algorithms. The first algorithm implemented was the decision tree, followed by the random forest. Both algorithms operate by creating a pattern of decisions based on training data that will allow it to sort packets into malicious and non-malicious categories.

A. Decision Tree

The purpose of a decision tree is to distinguish two different types of data points from a dataset containing both types. In this case our two types of data points are normal and malicious packets. Decision trees operate by choosing some expression that can sort all data points into two distinct groups. The goal of the machine learning algorithm is to choose the decision that produces the largest possible group of a single type of data. This will result in one group that contains a single type and another that contains both types. This process can be repeated on the group that contains both types, until all groups contain a single type, at which point all data points will have been categorized by type. The decision tree algorithm was trained with the *X_train* and *y_train* parameters, which are approximately half of the *data* and *y_data* sets, and took approximately 0.407 seconds for the C&C training and 0.002 seconds for the Torri training.

B. Random Forest

The purpose of a Random Forest is to improve the accuracy of a decision tree and decrease the dependency on the training data. This is done by subdividing the training data into multiple data sets and then creating a decision tree for each set. The multiple trees create a forest that make decisions based on a majority consensus of the trees. The increased number of trees can create more generalized decisions that are less specific to the

training data and generally more accurate. The random forest algorithm was trained with the same parameters as the decision tree algorithm and took approximately 3.21 seconds for the C&C training and 2.006 seconds for the Torri training.

IV. C&C Attack Results

The goal of the decision tree and random forest algorithms is to identify malicious packets that contain C&C attacks. The success of this goal can be measured by comparing each packet to the algorithm's prediction of that packet. This system can classify every prediction into four categories. A true positive indicates that the algorithm successfully identified a malicious packet. A true negative means that the algorithm successfully identified a non-malicious packet. A false positive is a mistake in which the algorithm incorrectly identifies a non-malicious packet as a malicious one. This is a minor issue that can cause loss of packets but does not threaten the security of the system being protected. A false negative is a mistake in which the algorithm fails to identify a malicious packet. This is the most dangerous of outcomes, as it is a complete failure of the intended purpose of the intrusion detection system and allows a malicious packet to reach its intended target. The frequency of these four outcomes can be expressed in several ways and used to calculate the analysis metrics that will be used to measure the success of the algorithms' results.

A. Analysis Metrics

The four primary metrics being used to measure the success of the algorithms are accuracy, precision, recall, and F1-score. Using multiple metrics can help to verify results if all metrics are reporting similar findings. Using a single metric can lead to misleading results depending on the specific data and the nature of the metric. Different metrics can also favor different types of results and show more detailed findings such as false positives being more common than false negatives.

- Accuracy is a measure of the lack of errors in a process. It is a generally effective metric than can show how effective a process is at producing the proper results and is calculated using (1).

$$Accuracy = \frac{True}{Total} \tag{1}$$

- The accuracy of the decision tree algorithm was approximately 99.929% and the accuracy of the random forest algorithm was approximately 99.942%. These results show a small but measurable difference in the accuracy of these algorithms.
- Precision is a measure of how frequently predicted positive results match their actual results. This metric is particularly useful for identifying the occurrence of false positives and is calculated using (2).

$$Precision = \frac{True\ Positive}{Predicted\ Positive} \tag{2}$$

- The precision of the decision tree algorithm was approximately 99.891% and the precision of the random forest algorithm was approximately 99.908%. These results show a small but measurable difference in the precision of these algorithms.
- Recall is a measure of how effective a process is at effectively identifying positive values. This metric is particularly useful for identifying the occurrence of false negatives and is calculated using (3).

$$Recall = \frac{True\ Positive}{Actual\ Positive} \tag{3}$$

- The recall of the decision tree algorithm was approximately 99.983% and the recall of the random forest algorithm was approximately 99.987%. These results show a small but measurable difference in the recall of these algorithms.
- F1 Score is effectively an average of recall and precision. This is a more generalized metric that evenly represents false positives and false negatives. Similar to accuracy it does not give insight to any specific measure but is a generalized overview of the validity of the predictions. It is calculated using (4).

$$\text{F1 } score = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

- The F1 score of the decision tree algorithm was approximately 99.937% and the F1 score of the random forest algorithm was approximately 99.948%. These results show a small but measurable difference in the F1 scores of these algorithms.

The equations used to calculate the four-analysis metrics were provided by [2] and automatically implemented within the program by the included libraries.

B. Confusion Matrix

The confusion matrix is read by treating the top and bottom halves as the actual results and the left and right halves as the predicted results. This creates four boxes with true negative in the top left, false positive in the top right, false negative in the bottom left and true positive in the bottom right. The darkness of each of the boxes is representative of a gradient showing the number of values within that category similar to the height of a histogram. This allows a three-dimensional graph to represented on a two-dimensional plane.

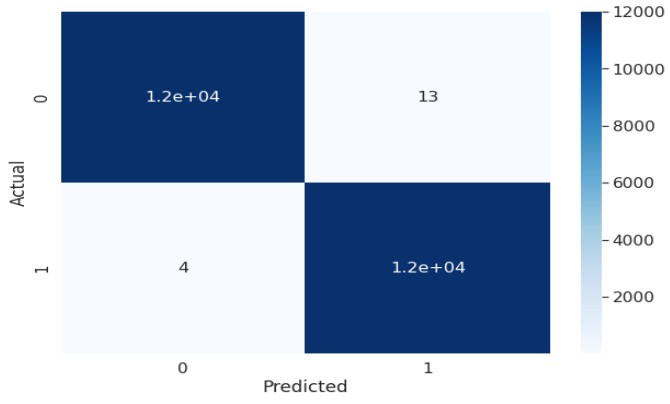

Fig. 1. Confusion matrix of C&C decision tree algorithm

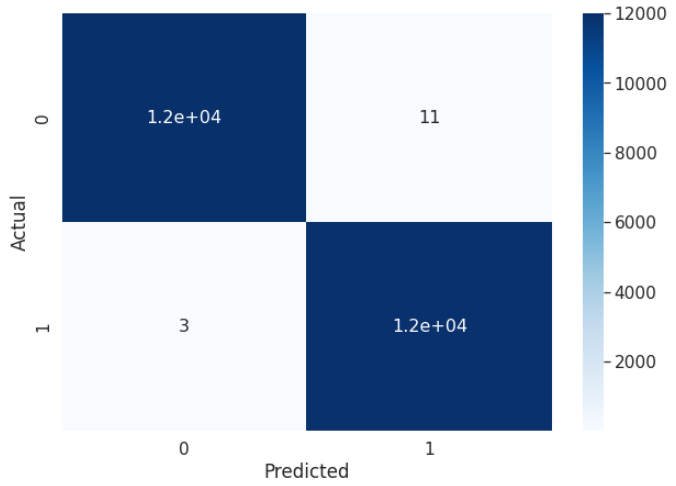

Fig. 2. Confusion matrix of C&C random forest algorithm

C. *Conclusions*

After comparing the results of the decision tree and random forest algorithms, I believe that the random forest algorithm was slightly more successful than the decision tree algorithm because it performed slightly more favorably in every metric. It had similar but measurably higher values in all four of the calculated metrics and a lower number of both false positives and false negatives. Both algorithms were very successful and had comparable results however, the random forest algorithm, which is more robust and time consuming, was slightly more effective at detecting malicious packets.

V. Torri Attack Results

After completing the analysis of the C&C attack data, the same process was replicated to test for Torri attacks instead. The algorithms were retrained with new data to recognize the new type of attack. Once trained, packets could be sent through both programs in series to test for both types of attacks.

A. *Analysis Metrics*

The same four primary metrics were used to measure the success of the of the Torri attack detection algorithms. Using the same metrics allows for a proper comparison of the effectiveness of the decision tree and random forest algorithms at detecting each of the two different types of attacks.

- The accuracies of the decision tree algorithm and random forest algorithm were both equal and approximately 99.994%. These results show no difference in the accuracy of these algorithms when calculated with (1).
- The precision of the decision tree algorithm was approximately 99.998% and the precision of the random forest algorithm was exactly 100%. These results show a small but measurable difference in the precision of these algorithms when calculated with (2).
- The recall of the decision tree algorithm was exactly 100% and the recall of the random forest algorithm was approximately 99.994%. These results show a small but measurable difference in the recall of these algorithms when calculated with (3).
- The F1 scores of the decision tree algorithm and random forest algorithm were both equal and approximately 99.997%. These results show no difference in the F1 scores of these algorithms when calculated with (4).

The accuracy and F1 scores produced by the two algorithms were identical even though they encountered different errors. This indicates the importance of using multiple metrics to measure the success of the algorithms. Without the precision and recall metrics, there would be no indication that the algorithms produced different results.

B. *Confusion Matrix*

The confusion matrices of the decision tree and random forest algorithms were able to visually represent the subtle differences between the results of the two algorithms. Although the number of errors that occurred with each algorithm is the same, the types of errors which can be seen on the confusion matrices varied. The visualization of Fig.3 and Fig.4 helps to show the differences between the similar results of the decision tree and random forest algorithms when predicting Torri attacks.

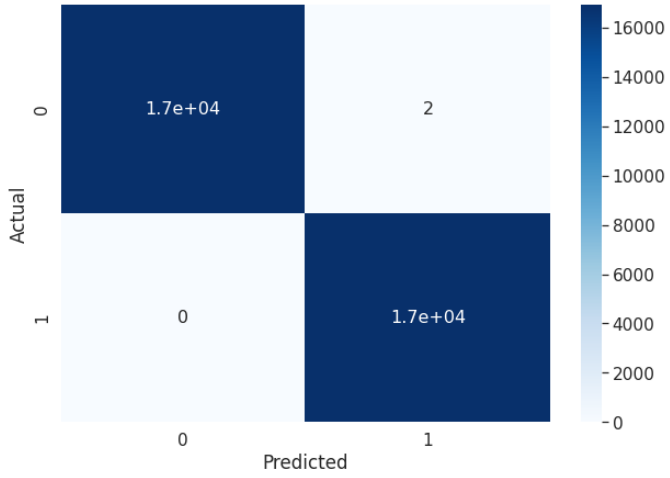

Fig. 3. Confusion matrix of Torri decision tree algorithm

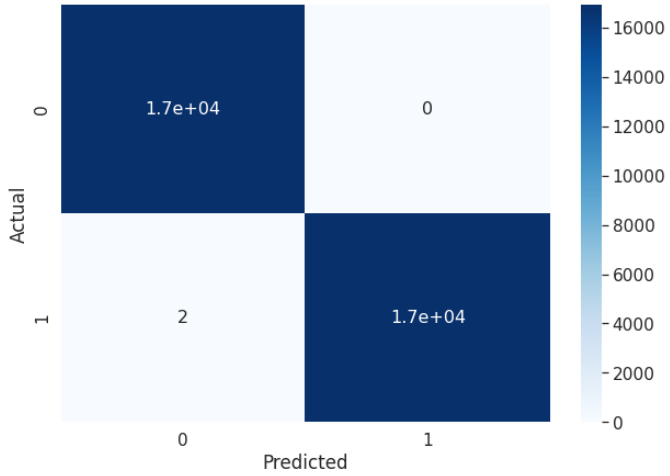

Fig. 4. Confusion matrix of Torri attack random forest algorithm

C. *Conclusions*

After comparing the results of the decision tree and random forest algorithms, I believe that the decision tree algorithm was slightly more successful than the random forest algorithm because of its perfect recall score. Despite having matching accuracy and F1 scores as well as a slightly lower precision, the decision tree algorithm was able to successfully identify 100% of malicious packets. I believe that in this situation recall is a more significant metric than precision due to the difference between the potential outcomes of a false positive compared to a false negative.

APPENDICES

- **Appendix 1**. This code was used to create, train, and test both algorithms for both attack types.
- **Appendix 2.** These results were generated by the code found in Appendix 1.

REFERENCES

[1] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IOT-23 dataset: A labeled dataset of malware and benign IOT traffic.," *Stratosphere IPS*, 2020. [Online]. Available: https://www.stratosphereips.org/datasets-iot23. [Accessed: 13-Dec-2021].

[2] K. P. Shung, "Accuracy, precision, recall or F1?," *Medium*, 10-Apr-2020. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. [Accessed: 13-Dec-2021].

Appendix 1

https://colab.research.google.com/drive/1gQ47iWN1MPSZpJlr5jWGfDrCSzaazNg-?usp=sharing

Appendix 2

https://drive.google.com/file/d/1Zc4Wc4Q3uCZCilzBRnxgfhigGqjkRfjn/view?usp=sharing