

Project Report

Identifying Malicious/Non-Malicious Network Connections

Security, Privacy and Data Analytics
ECE 572



**University
of Victoria**

Submitted to:

Dr. Zahra Nikdel

Submitted by:

Sanjana Arora (V00966221)

Siddharth Chadda (V00947906)

Tavanpreet S. Oberoi (V00963163)

TABLE OF CONTENTS

1. Abstract	01
2. Summary	01
3. Introduction	01
4. Methodology	02
5. Results and Discussions	05
6. Conclusion	09
7. References	09

1. Abstract

The report focuses on the analyses applied on a Snort Detection Log [1] dataset released by the National Security Agency to identify a malicious or non-malicious network connection within an organization's private network. The project uses Python as a primary programming tool to analyze the dataset and apply different machine learning algorithms to classify the data into malicious or non-malicious connections. The dataset includes 34 columns; therefore, to reduce the dimensionality of the dataset, few columns have been combined into one as a first preprocessing step. Some of the features that did not provide useful information for solving the problem of classifying the data have been removed. Further, data preprocessing steps such as missing value imputation and data normalization are also discussed in detail in this report. After the data preprocessing, appropriate machine learning algorithms have been applied to solve the given binary classification problem. Ensemble supervised machine learning algorithms such as XGBoost, AdaBoost and random forest have been used as these algorithms provide good performance and accuracy even on large datasets. The performance and evaluation metrics have also been compared for each of these machine learning algorithms in the report.

2. Summary

The preliminary activities such as data conversion and preparation are used to lay the groundwork for developing Machine Learning detection techniques. The first step is to convert a log file which is in the form of '.txt' to '.csv' file with required features needed for the analysis. Now the '.csv' file is read, and various pre-processing steps are applied like dropping irrelevant columns having NaN or null values, dropping rows where flag is not set and many more. Further standardization is applied on the given dataset, so that none of the features should dominate the other [2]. In each methodology, different ways of short-listing important features or top-level features are illustrated, which aid in presenting complex high-dimensional data in simple lower dimensions. The report delves into the ideas that are the most significant components of data analysis and are necessary before examining and working with any dataset. The supervised machine learning approaches, like Random Forest Classifier, AdaBoost and XGBoost, have been used to segregate clusters of binary classes (malicious and non-malicious connections) in a dataset. We've also included the findings of the classification, as well as the accuracy. All the work discussed above was done in Python, which has better data analysis packages. We have concluded the report by discussing the importance of some features that convey significant information and help in identifying the type of attack that the connection in the log file was intending to execute on the private network.

3. Introduction

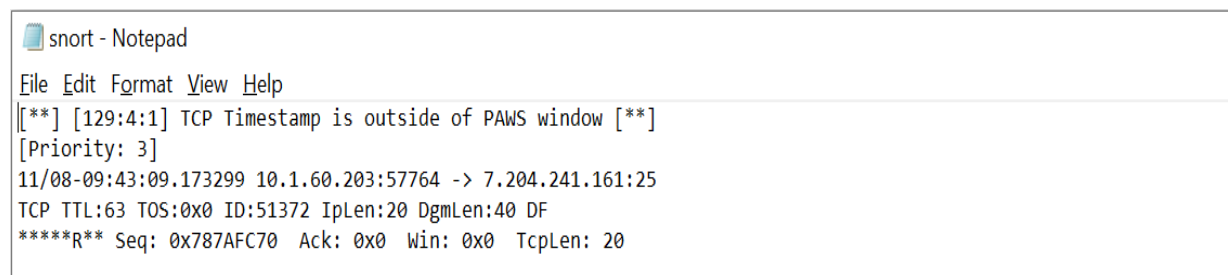
We have used the Snort Detection log dataset to perform the binary classification of data into the malicious and non-malicious connections within an organization's network connections. As the first step of the project, we have performed various pre-processing steps on the data including removal of the features that do not contain much information, missing value imputation and normalizing the ranges of the features. We have applied various powerful ensemble machine learning algorithms to achieve good accuracy of classification. Further, we

have used various evaluation metrics to compare the performance of the various classification algorithms.

4. Methodology

Dataset used

We have used the Snort Intrusion Detection Log [1] dataset released by the National Security Agency (NSA) provided by United State Military Academy WestPoint. The Snort Intrusion Detection Alert Log captures the data of all the connections being initiated or disconnected with an organization's private network. The Log file includes information related to Source IP address, Destination IP address, timestamp of the connection, TCP length, IP length, Source Port Number, Destination Port Number, Protocol, Flags, Type of Service, etc. We have parsed the Snort Log file into a CSV format to capture all these data fields in the form of feature columns. For performing the data analysis, we use these feature columns to classify the activity as malicious or non-malicious. For instance, one log of the SNORT file represents the flags that are set for each connection and this field is of 8-bit length where each bit is reserved for each of 8 flags (Flag 1,2, U, A, P, R, S, F).



```

snort - Notepad
File Edit Format View Help
[**] [129:4:1] TCP Timestamp is outside of PAWS window [**]
[Priority: 3]
11/08-09:43:09.173299 10.1.60.203:57764 -> 7.204.241.161:25
TCP TTL:63 TOS:0x0 ID:51372 IpLen:20 DgmLen:40 DF
***** Seq: 0x787AFC70 Ack: 0x0 Win: 0x0 TcpLen: 20

```

Figure 1 – Snapshot capturing log of SNORT file

1	Action Per	Connectio	Priority	Date-Time	Source IP	Source Poi	Destinatio	Destinatio	Protocol	TimeToLiv	TypeOfSer	ID	IP Length	DataGram
2	TCP Times		3	1.57E+09	10.1.60.20	57764	7.204.241.	25	TCP	63	0x0	51372	20	40
3	TCP Times		3	1.57E+09	7.204.241.	25	10.1.60.20	50176	TCP	64	0x0	1283	20	40
4	TCP Times		3	1.57E+09	7.204.241.	25	10.1.60.20	50176	TCP	64	0x0	1284	20	40
5	TCP Times		3	1.57E+09	154.241.84	80	3.75.190.1	60708	TCP	63	0x0	21269	20	52

Figure 2 – Snapshot of parsed SNORT file

The Figure 2 represents the parsed SNORT Log file in the CSV format. As can be observed from figure 1 and 2, the parsed SNORT file includes the same features as the SNORT file. The parsed SNORT file includes 25741 rows and 34 columns. The SNORT file did not include header names; therefore, we have added the column names to the fields in the CSV file. Table 1 represents the description of the primary feature columns.

Table 1. Columns with their description

Action Performed	Description about the Connection
Connection classification	Two different classes, connections which are malicious and non-malicious . Within non-malicious, we have values each explaining different types of cyber-attacks.
Priority	Order for the actions to be taken for the malicious connections. A priority of 1 (high) is the most severe and 4 (very low) is the least severe.
Date-Time	Date and time at the connection was logged
Source IP	IP address of the system which generates a request and directs it towards the destination system.
Source Port	The port from which the request was made to the network
Destination IP	IP address of the system to which the request or response is directed.
Destination Port	The port to which the request is made.
Protocol	We have 4 classes under this feature - IP, TCP, UDP, ICMP. It gives information about the rules which are being used in establishing the connection.
Time to Live	Time to live (TTL) or hop limit is a mechanism that limits the lifespan or lifetime of data in a computer or network. This takes numbers from 0 to 255. The Time to Live (TTL) field keeps a counter that decrements every time a packet crosses a router.
TOS (Type of Service)	The Type of Service field is used for prioritizing traffic for performance. It is a 8-bit string which tells about precedence, delay, reliability, minimum cost and throughput. In the dataset this feature takes 6 values (61,64,240,0x0,0x10,0xC0).
ID	A unique identifier for each record.
IP length	This specifies the IP packet header length which is 20 bytes for IPv4.
Datagram Length	IP datagram consists of a header part and text part. It specifies the total length of the datagram in bytes.
Flags	It is denoted by a 8-bit string which tells which flags are set for the connection. Each bit indicates a distinct flag.
Seq	The seq keyword is used to check for a specific TCP sequence number.
Ack	acknowledgment number.
TCP length	size of the TCP header in 32 bits.

Data Preprocessing

As the dataset included 34 columns, we selected the important features for our analysis as it is difficult to analyze such a high dimensional dataset. We removed all the individual columns of the flags namely Flag 1, 2, U, A, P, R, S, F and used a single column “All flags” instead to understand the type of connection established. Further, we also dropped the protocol, ID and IP length columns as they were not having much information for classifying between malicious or non-malicious attacks. As some of the columns such as Connection Classification, All flags, NOP NOP TS, TCP options, MSS, NOP WS, SackOS TS include NA values, we followed appropriate methods to treat the NA values. We imputed the samples having NA in the Connection classification column with 0 as 0 would simply mean that there is no classification

category information for these samples. Further, we used only those rows where the “All flags” column had some value as only those rows have an established connection. For the remaining dataset, we imputed the columns having NA values in NOP NOP TS, TCP options, MSS, NOP WS, SackOS TS with 0. Finally, we achieved a dataset with 0 NA in the dataset and applied Minmax Scalar transformation for scaling the features to (0,1) range. We applied this transformation as the feature columns were varying in different ranges and a feature with higher range could bias the classification models.

Machine Learning Classifier

Supervised Learning algorithms are implemented on the pre-processed dataset as the dataset already contains a label vector to categorize between malicious and non-malicious attacks.

Pre-processing was applied on column name ‘ConnectionClassification’ to convert label data into binary classification problems. Initially the column contains ‘text’ as the type of attack recorded and ‘empty’ if it is not malicious. All the text is being converted to value 1 (Malicious) and empty is replaced with value 0 (non-Malicious). As the problem has been converted to binary classification problem, so Supervised Learning Algorithms are applied on the pre-processed dataset.

Further dataset is split into training and testing dataset, using the ‘train_test_split’ model available in sklearn library. The training data contains 75% of the dataset while testing contains the remaining 25% of the dataset.

The approach shown in Figure 3 has been followed for classification.

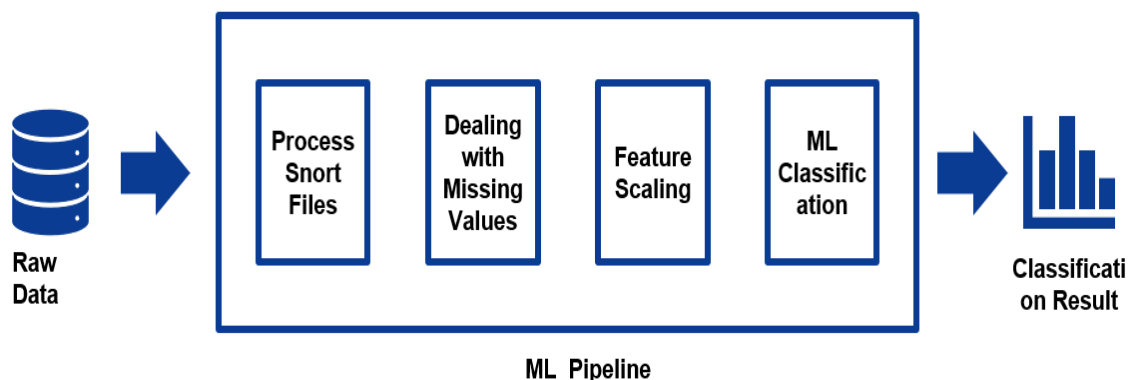


Figure 3. Machine Learning Pipeline

Analysis of the Classification Algorithms

All the deployed classification algorithms are analyzed based on their classification report which provide us four important evaluation parameters [3].

- Precision: How many data points are correctly classified in the class
- Recall: How many of this class you find over the whole number of elements of this class
- F-1 score: Harmonic mean between precision & recall
- Support: Number of occurrences of the given class in your dataset

5. Results and Discussions

Because of the classification-based nature of the problem we can use the following machine learning algorithms to classify each data point and assign the appropriate type to it.

1. Random Forest

Random Forest is a supervised machine learning algorithm used for classification tasks. Random Forest belongs to the family of ensemble algorithms. The Random Forest algorithm builds a “Forest” using an ensemble of Decision Trees trained using the “bagging” method. The main idea behind “bagging” is that by combining multiple machine learning models, the overall accuracy of the result increases. In other words, the random forest algorithm builds multiple decision trees and then combines their results to get a high accuracy classification.

Classification Report is shown in figure 4.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1666
1	1.00	1.00	1.00	4157
accuracy			1.00	5823
macro avg	1.00	1.00	1.00	5823
weighted avg	1.00	1.00	1.00	5823

Figure 4. Classification Report Random Forest

Feature Importance is Visualized in figure 5.

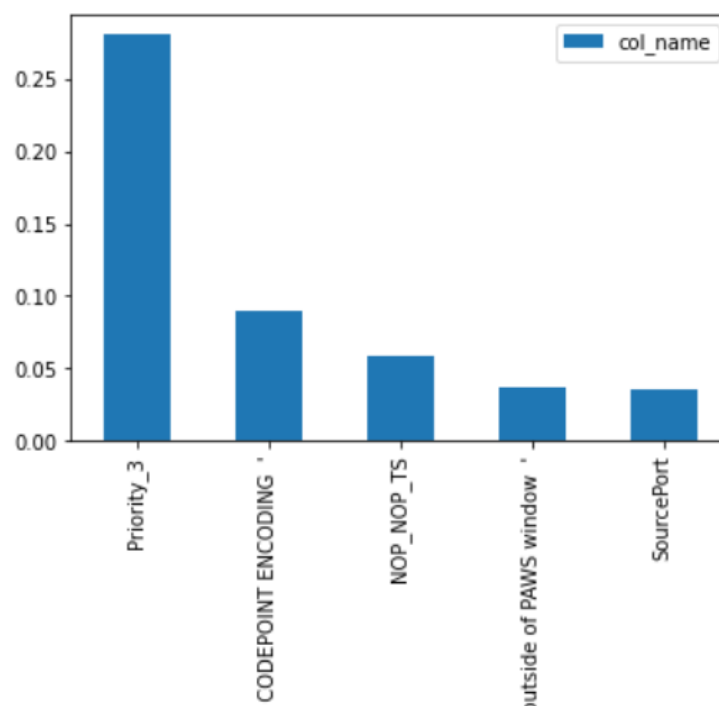


Figure 5. Feature Importance Random Forest

Observation:

1. Priority 3 has maximum feature importance because all the data points having 'Priority' = 3, is classified as 0(non-Malicious), whereas for all other 'Priority' it is classified as 1(Malicious).

A	B	C	D	E	F	G	H
ActionP	Connec	Priority	Date-Ti	SourceI	SourcePort	Destina	Destina
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.881699905	7.204.241.	25
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.887030380	154.241.8	443

Figure 6. Input Data Observation for Priority = 3

2. Feature 'ActionPerformed' with value 'b' (http_inspect) IIS UNICODE CODEPOINT ENCODING ' ' is always classified as 0(non-Malicious). It can be considered as safe 'ActionPerformed'.

2. Adaptive Boosting (AdaBoost)

The Adaptive Boosting algorithm, also called AdaBoost, is a “boosting” technique based supervised ensemble machine learning algorithm. In Adaptive Boosting we start with multiple “weak” machine learning classifiers like Decision Trees and then progressively combine them into a single “strong” classifier. The algorithm achieves this by continuously re-assigning the model's weights during the training process. The algorithm assigns a higher weight to instances which are difficult to classify, as compared to easier data points. Through this method, we can create powerful classifiers even when the constituent models are extremely simple.

Classification Report is shown in figure 7

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1666
1	1.00	1.00	1.00	4157
accuracy			1.00	5823
macro avg	1.00	1.00	1.00	5823
weighted avg	1.00	1.00	1.00	5823

Figure 7. Classification Report AdaBoost

Feature Importance is Visualized in figure 8.

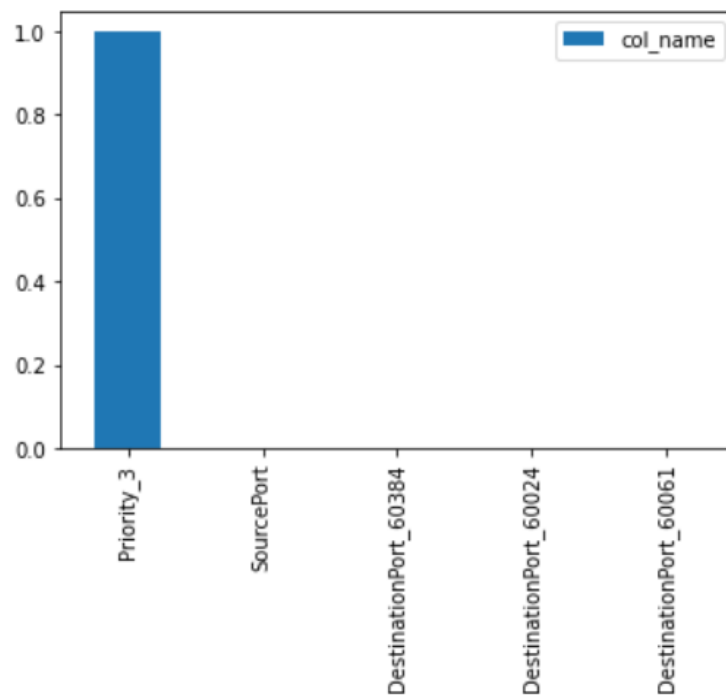


Figure 8. Feature Importance AdaBoost

Observation:

1. Priority 3 has maximum feature importance because all the data points having 'Priority' = 3, is classified as 0(non-Malicious), whereas for all other 'Priority' it is classified as 1(Malicious).

A	B	C	D	E	F	G	H
ActionP	Connec	Priority	Date-Ti	SourceI	SourcePort	Destina	Destina
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.881699905	7.204.241.	25
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.887030380	154.241.8	443

Figure 9. Input Data Observation for Priority = 3

2. Feature 'DestinationPort' with values '(60384,60024,60061)' is always classified as 0(non-Malicious). It can be considered as a safe 'DestinationPort', but since importance is negligible, it should not be treated as a 'safe' port. Less number of data records could be one of the reasons for such low importance.

3. XGBoost

XGBoost stands for eXtreme Gradient Boosting, it is also a decision tree based supervised ensemble machine learning algorithm which uses gradient descent boosting for optimization.

The XGBoost algorithm was developed at the University of Washington, the algorithm is extremely optimized for solving classification problems. The algorithm achieves this by using a combination of parallel-processing-based tree pruning, regularization, and the gradient descent algorithm to minimize the classification error.

Classification Report is shown in figure 10.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1666
1	1.00	1.00	1.00	4157
accuracy			1.00	5823
macro avg	1.00	1.00	1.00	5823
weighted avg	1.00	1.00	1.00	5823

Figure 10. Classification report

Feature Importance is visualized in figure 11.

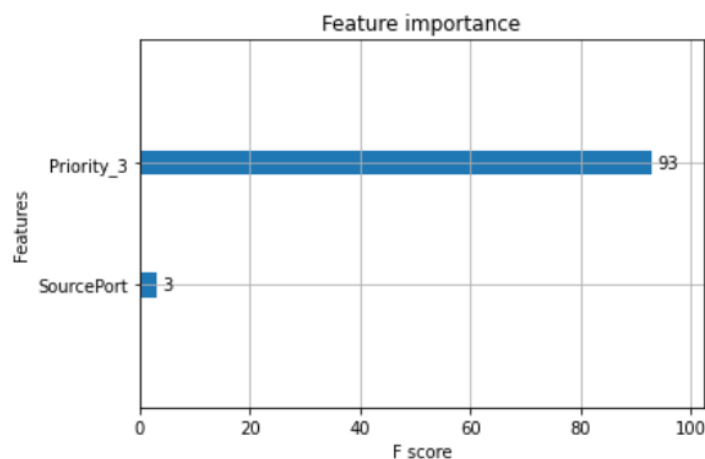


Figure 11. Feature Importance XGBoost

Observation:

1. Priority 3 has maximum feature importance because all the data points having 'Priority' = 3, is classified as 0(non-Malicious), whereas for all other 'Priority' it is classified as 1(Malicious).

A	B	C	D	E	F	G	H
ActionP	Connec	Priority	Date-Ti	SourceI	SourcePort	Destina	Destina
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.881699905	7.204.241.	25
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	7.204.241.	0	10.1.60.20	50176
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	154.241.8	0.000839874	3.75.190.1	60708
b' TCP Tim	0	3	1.57E+09	10.1.60.20	0.887030380	154.241.8	442

Figure 12. Verification of observation from input file

6. Conclusions

The report represents the techniques and algorithms used in Machine Learning to detect malicious connections. The three different classification algorithms are compared and XGBoost dominates others in terms of execution time with 100% accuracy. We worked on data normalisation, data pre-processing, feature extraction techniques, and classification approaches, as well as their implementation.

7. References

- [1] <https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets>
- [2] <http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
- [3] <https://datascience.stackexchange.com/questions/64441/how-to-interpret-classification-report-of-scikit-learn>