2022

Assignment – 2 Report

SECURITY ANALYTICS (COMP90073)

ADITYA DESU

Table of Contents

Introduction

The following report covers an implementation of unsupervised machine learning techniques for anomaly detection and gradient descent-based methods to generate adversarial examples against supervised learning models. The machine learning techniques were applied on network traffic (NetFlow) datasets, the datasets contain botnet traffic and normal traffic.

Task 1 – Unsupervised Anomaly Detection

Dataset and Dataset Loading

The dataset used for analysis in this task contained NetFlow data for a network under cyberattack captured between 8th August 2021 and 13th August, 2021. The dataset provided in the form of three csv files and divided into training, validation, and testing. Each of the datasets included the following number of network data instances:

i) Training dataset included 1,529,446 instances,
ii) Validation dataset included 764723 instances, and
iii) Testing dataset included 764722 instances

The training and validation dataset provided included 15 features for each instance and the testing dataset included 16 features. As the provided dataset did not have feature names included all three datasets were loaded on Splunk and the extract fields method Splunk's machine Learning Toolkit was used to provide features names to the data. The features including datatypes are described in the table 1:

	Feature	Data type
1.	Stream ID	Integer
2.	Timestamp	Datetime
3.	Duration	Numeric - Float
4.	Protocol	String
5.	Source IP address	Numeric
6.	Source port	Numeric - Int
7.	Direction	String
8.	Destination IP address	Numeric
9.	Destination port	Numeric- Int
10.	State	String
11.	Source type of service	Numeric- Float
12.	Destination type of service	Numeric- Float
13.	No of total packets	Numeric- Int
14.	The number of bytes transferred in both directions	Numeric- Int
15.	The number of bytes transferred from the source to the destination.	Numeric - Int

TABLE 1 FEATURES OF DATASET PROVIDED

Further data analysis was carried out on splunk….

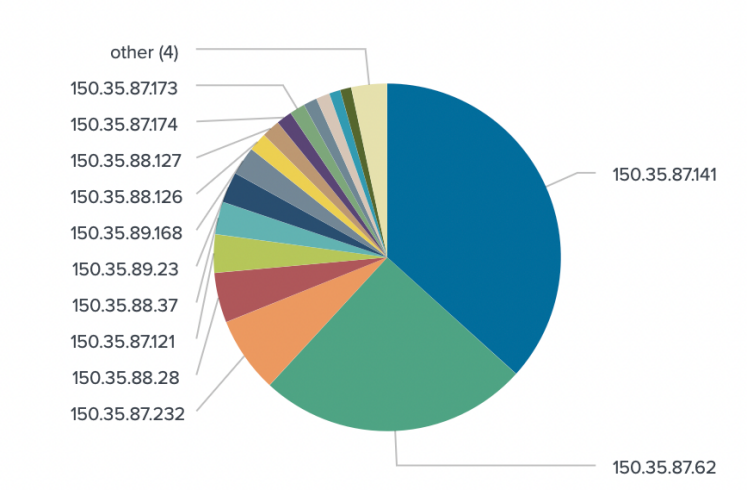


FIGURE 1 TOP 20 SOURCE IP'S IN TRAINING DATA

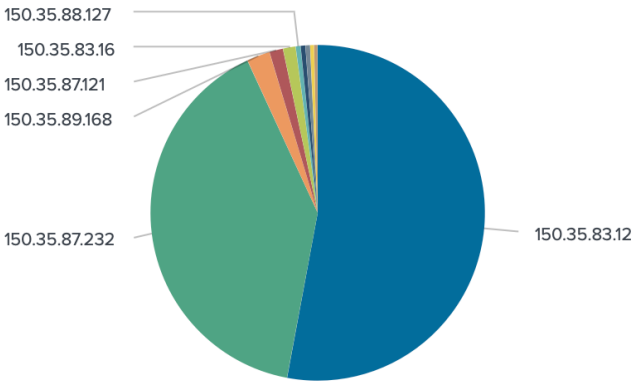


FIGURE 2 TOP 10 DESTINATION IP'S IN TRAINING DATA

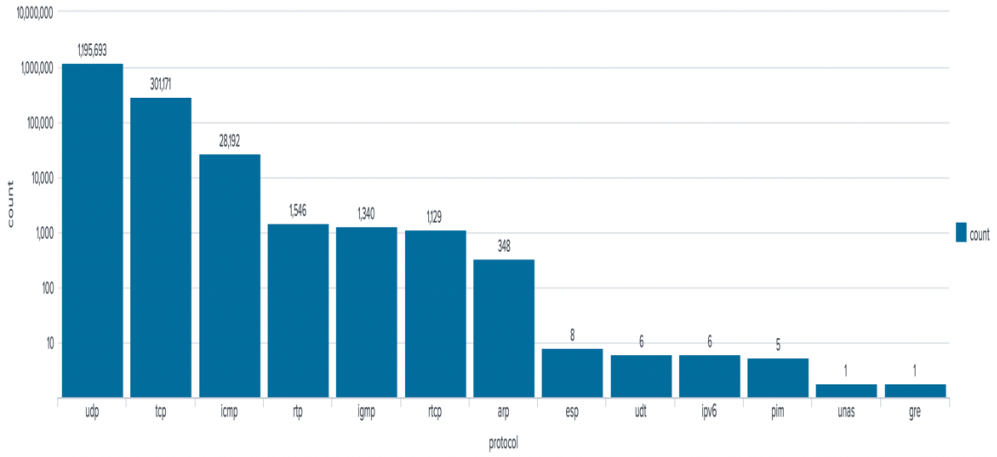


FIGURE 3 TOP PROTOCOLS

The dataset was downloaded and loaded on a Jupyter Notebook using the pandas library.

Data Processing

Firstly, the data was inspected for any null values using the isnull() command and grouped by the features. The total number of instances with null values grouped by features include:

i) State – 12 null values
ii) Source Type – 6041 null values
iii) Destination Type – 255,279 null values

As discarding all these instances would result in a significant reduction of training data available, the 12 instances with missing state values were discarded. The source and destination type included binary values and the most frequently occurring value was 0 and the null values for the source and destination type features were replaced by this mode value.

Features with multiple categories such as protocol – tcp, http, state, IP addresses, were encoded into numeric values using the label encoder function from the sklearn package so they can be fit by machine learning models during training and evaluation. This step ensures the features are standardised.

Feature Generation

In assignment 1, we identified patterns that are useful to find any botnet attacks, the patterns for the respective attacks are as follows:

Attack	Features
Click Fraud	Src_ip + dst_ip + dst_port +uri
Spam	Src_ip + dst_ip +dst_port
IRC	Src_ip + dst_ip + dst_port

TABLE 2 USEFULL FEATURES TO IDENTIFY BOTNET ATTACKS

As these features already exist in the original feature set, we generate new features leveraging the existing features for each instance to help identify anomalies. Some of the features on bytes transferred were identified as important in botnet detection (Ritu & Kaushal, 2015).

The features generated are the following:

Feature	Feature Description
Pps	Packets sent per second (pps)
Bps	Bytes transferred per second (bps)
Bps_src	Bytes transferred per second from source (bps_src)
Bps_two_way	Bytes transferred per second from destination (bps_two_way)
Duration	Duration – time difference between each instance – requests made with more frequency are likely to be a botnet attack

TABLE 3 FEATURE GENERATED FEATURES

Experiment Design

The anomaly detection experiment was completed on three feature sets of data and two anomaly detection methods. The feature sets include a set of generated features, a set of features selected using filter-based methods – Variance Thresholding and Pearson Correlation with Variance Inflation Factor. Feature selection is important for several reasons, the fundamental one being improving prediction quality, the more information we have about each pattern, the better a clustering algorithm is expected to perform (Law et al., 2004). Filter based methods require less computations than wrapper-based feature selectors which often are supervised and sequential. Both variance thresholding and correlation methods are widely used in unsupervised anomaly detection (Doreswamy et al., 2020, Pasillas-Díaz & Ratté, 2016, Shyu et al., 2003).

Feature Selection Method 1 – Variance Threshold

Variance threshold is a feature selection method that removes all low variance features.

Feature Selection Method 2 – Removing Multicollinearity with Pearson Correlation and Variance Inflation Factor

The Pearson Correlation maps linearity between two features between -1 and 1. We use this correlation to create a correlation matrix and use a threshold (Variance Inflation Factor) of 10 to remove features less than this threshold. O'brien (2007) emphasises the interpretation of VIF in a context is more important than following a conventional rule of thumb of 4-10 VIF. Upon examining the results of the correlation matrix in figure 4, the ViF score of 10 appeared to a reasonable threshold.

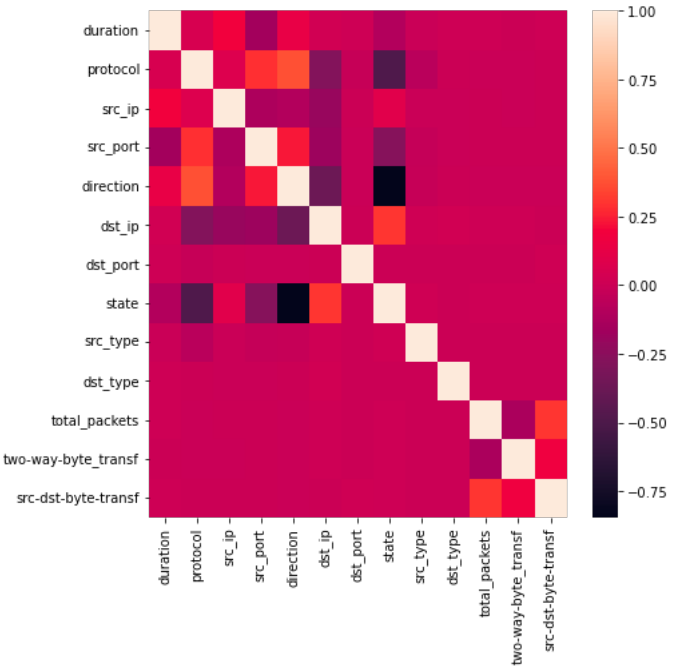


FIGURE 4 CORRELATION MATRIX OF FEATURES

Feature	**Data type**
Duration	1.250
Protocol	24.913
Source IP address	3.154
Source port	5.575
Direction	9.208
Destination IP address	4.145
Destination port	1.000
State	4.093
Source type of service	1.001
Destination type of service	1.000
No of total packets	1.147
The number of bytes transferred in both directions	1.073
The number of bytes transferred from the source to the destination.	1.162

TABLE 4 VIF FACTOR OF ALL FEATURES

Model 1 – Isolation Forrest

Isolation forest is an unsupervised model based on decision tree algorithm It isolates the outliers by randomly selecting a feature and then randomly selecting a split value between the max and min values of that feature. This random partitioning of features will produce shorter paths in trees for the anomalous data points, thus distinguishing them from the rest of the data.

Model 2 – Local Outlier Factor (LoF)

Local Outlier Factor is an unsupervised density-based model, it measures the local reachability of a given sample with respect to its nearest neighbours. By comparing the local density of a sample with its neighbours, anomalies can be identified as samples coming from areas with low density.

Training and Feature Engineering

Apart from the features, the model parameters play a significant role in producing accurate results. Each of the above two models were trained on three sets of training data with 1,529,446 instances. The feature generation and selections methods -variance thresholding and correlation-based filtering were applied to the training set and the validation data was used for feature engineering. During the feature engineering stage, the features which were discarded in the training set were also removed from the validation set and model's performance was assessed.

The k nearest neighbours was picked as 20 which is the default setting for both models. Contamination was the parameter that was used for tuning the model. Contamination is the expected proportion of data that is anomalous. Contamination is a threshold, it does not affect the scoring of each instance, but upon prediction contamination acts as a cut-off on scores and returns top x percentile negative scores as anomalies. (For eg: If contamination is set as 0.01 then points with top 1% negative scores are labelled anomalies). The contamination parameter was varied between 0.05, 0.1 and 0.15 while training for each of the models and the performance was observed on validation dataset.

Recall metric for both classes was used to evaluate the model performance; this metric provides the percentage of actual anomalies and normal data are discovered by the model. Since outliers are less common in the dataset, the dataset is imbalanced, so the model must be able to accurately identify both normal data and anomalies but more importantly find anomalies as the threat of not detecting an attack far outweighs the cost of examining a non-anomalous network traffic misclassified as an anomaly.

Finally, 0.15 was picked as the contamination threshold for Isolation Forest model and 0.1 was picked as the contamination parameter for Local Outlier Factor Model.

The training and testing were performed on a local macOS device with 16 GB RAM and 16 cores. The time required to train isolation forest on all three feature sets was constant at 23 seconds despite the feature generated model having 6 features and the other two feature sets having 14 features each. Time taken was consistent while predicting as shown in the table 4. The time train the LoF model varied upon the data set as shown below:

Feature set	No of features	Training time
Feature generated	6	26.3 s
Variance Threshold	14	9.9 s
Correlation	14	4 m 23 s

TABLE 5 RUN TIME PER FEATURE SET FOR LOF

Scoring and Threshold

Sklearn classifies an anomaly as -1 and a normal class as 1, we change the classification of anomalies to 1 and normal data to 0, this is how the label for the data is also stored.

Results

Variance threshold without standard scaler

Model	Contamination	Feature Selection	Normal Class Accuracy	Anomaly Accuracy	Overall Accuracy	Run Time
Isolation Forest	0.15	Feature Generated	0.79	0.05	0.79	10.1s
	0.15	Variance Threshold	0.75	0.89	0.75	10.3s
	0.15	Correlation	0.78	0.61	0.78	10.1s
Local Outlier Factor	0.1	Feature Generated	0.89	0.18	0.89	11.2s
	0.1	Variance Threshold	0	1	0	4.4 s
	0.1	Correlation	0.05	0.8	0.06	2m 16s

TABLE 6 RESULTS OF MODEL PERFORMANCE

The isolation forest model on the variance threshold feature set data was selected as the best performing as it performed well in detecting anomalies and normal data but at the cost of misclassifying normal data as botnet which can be seen from the high number of false positives for normal data in figure 5.

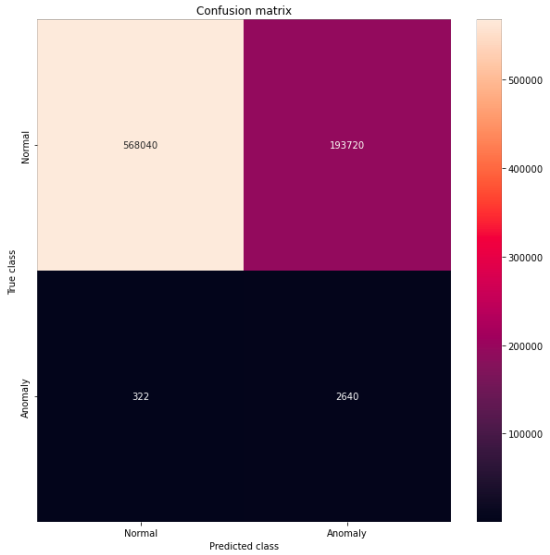


FIGURE 5 CONFUSION MATRIX OF ISOLATION FOREST MODEL ON TEST DATA

The timestamp of the first attack detected starts at 2021-08-12 20:56:35.747945 from the source IP 150.35.87.168 and ends at 2021-08-13 01:17:54.613374. The list of all detected attackers and victims can be found in the appendix B. Protocol, direction and source port were the most influential features in isolation forest's decision making based on the results of Shapley Additive explanations which is a method to explain individual predictions in a game theoretical manner (Lundberg & Lee, 2017).

.

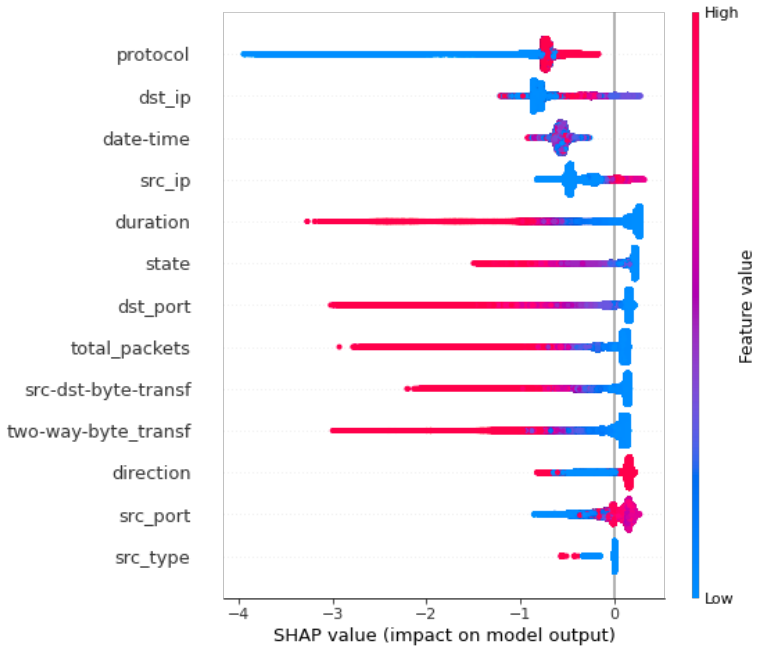


FIGURE 6 IMPORTANCE OF FEATURE DURING TESTING USING SHAP RATIO

While training the LoF model on the feature generated dataset combined with the original features, the run time exceeded 45 min whereas runtime using isolation forest was under 10 minutes. Further examining the run time of models from Table 6, we can infer that the LoF model does not scale up well.

Due to being a clustering-based algorithm that has a runtime of $O(n^2)$. Alghushairy et al., (2020) also find that LoF model is not suitable for netflow data and data with high velocity.

Conclusion

Isolation forest model performs better than the local outlier factor with respect to runtime and accuracy in detecting anomalies. The isolation forest model performs the best on the variance threshold dataset.

Task II

This task involved building, training, and testing a supervised learning model. Generating adversarial examples for a chosen botnet IP address, i.e., modify its feature values such that it is classified as a normal dataset.

Data Description and Processing

The data provided for this supervised task consisted of the following dimensions:

i) Training Data – 2,511,874 instances with 2485127 normal instances and 26747 malicious instances (bots)
ii) Validation Data – 1,255,936 instances with 1255924 normal instances and 12 malicious instances
iii) Test Data – 941,952 instances with 941889 normal instances and 63 malicious instances

Note: When the data is further grouped by source IP address, the dimensions of normal and malicious instances change.

The initial features provided for this dataset are the same as the features from Table 1, which were reproduced to the same set of features generated in part 1 described in Table 3. The dataset was furthered grouped by Source IP and aggregations (mean and variance) were performed to generate new features (Table 7). The numeric features were furthered scaled before training between -1 and 1 using the Standard Scaler function of sklearn.

Feature	Feature Description
mean_otb_pkt_size	Mean Outbound Packet Size
var_otb_pkt_size	Variance Outbound Packet Size
mean_pkt_cnt_per_sec	Mean Packet Count per second
max_pkt_cnt_per_sec	Max Packet Count per second
mean_pkt_jitter	Mean Packet Jitter
var_pkt_jitter	Variance Packet Jitter
mean_bps	Mean bytes per second
var_bps	Variance bytes per second
mean_bps_src	Mean bytes per second transferred from src to dst
var_bps_src	Variance bytes per second transferred from src to dst
mean_bps_two_way	Mean bytes per second transferred two way
var_bps_two_way	Variance bytes per second transferred two way

mean_bytes_per_pkt	Mean bytes sent per packet
var_bytes_per_pkt	Variance bytes sent per packet

TABLE 7 AGGREGREGATED FEATURES PER SOURCE IP ADDRESS

After grouping and aggregating features by source IP address, the class distribution within each dataset is as follows:

Training Set	
Normal Data	318235
Malicious Data	1
Validation Set	
Normal Data	50724
Malicious Data	1
Test Set	
Normal Data	5
Malicious Data	1

TABLE 8 CLASS DISTRIBUTION PER DATASET

Training and Validation

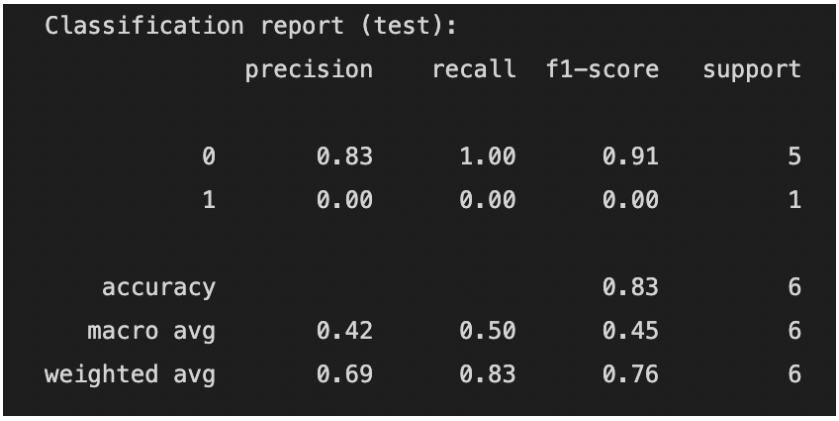
Top 6 features were selected using chi-square test which measures the dependence between stochastic variables, the features that are the most likely to be independent of class and therefore irrelevant for classification are removed. The training set was used training for logistic regression model and further tested on the validation and test dataset.

Logistic Regression

Logistic regression is a supervised model based on statistics that can be used to identify network traffic as malicious or not, it uses a logistic function to classify binary classes (Bapat et al. 2018).

Results and Retraining

The model first achieved an accuracy of 83.3% but failed to detect any anomalies.



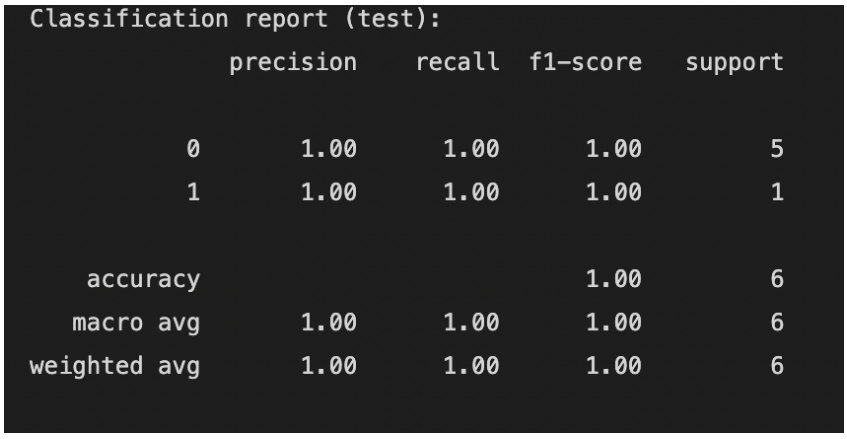
```
Classification report (test):
              precision    recall  f1-score   support

           0       0.83      1.00      0.91         5
           1       0.00      0.00      0.00         1

    accuracy                           0.83         6
   macro avg       0.42      0.50      0.45         6
weighted avg       0.69      0.83      0.76         6
```

FIGURE 7 RESULTS ON IMBALANCED CLASS DISTRIBUTION (LOGISTIC REGRESSION)

The poor performance of the model could have resulted due to the class imbalanced identified in table 5. To address this problem, the minority botnet class in the training set was increased through oversampling using RandomOverSampler function from imblearning package. This function randomly replaces some of the majority class instances with minority class instances. The same logistic regression model was retrained on this oversampled dataset, validation and testing datasets were used for testing. The model was now able to achieve 100% accuracy and correctly distinguish between normal and malicious dataset.



```
Classification report (test):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         5
           1       1.00      1.00      1.00         1

    accuracy                           1.00         6
   macro avg       1.00      1.00      1.00         6
weighted avg       1.00      1.00      1.00         6
```

FIGURE 8 CLASSIFICATION RESULTS ON OVERSAMPLED DATA (LOGISTIC REGRESSION)

References

Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2020). A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. *Big Data and Cognitive Computing*, *5*(1), 1. https://doi.org/10.3390/bdcc5010001

Bapat, R., Mandya, A., Liu, X., Abraham, B., Brown, D. E., Kang, H., & Veeraraghavan, M. (2018, April 1). *Identifying malicious botnet traffic using logistic regression*. IEEE Xplore. https://doi.org/10.1109/SIEDS.2018.8374749

Doreswamy, Hooshmand, M. K., & Gad, I. (2020). Feature selection approach using ensemble learning for network anomaly detection. *CAAI Transactions on Intelligence Technology*, *5*(4), 283–293. https://doi.org/10.1049/trit.2020.0073

Law, M. H. C., Figueiredo, M. A. T., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1154–1166. https://doi.org/10.1109/tpami.2004.71

Lundberg, S. M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. Neural Information Processing Systems; Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

O'brien, R. M. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity*, *41*(5), 673–690. https://doi.org/10.1007/s11135-006-9018-6

Pasillas-Díaz, J. R., & Ratté, S. (2016). An Unsupervised Approach for Combining Scores of Outlier Detection Techniques, Based on Similarity Measures. *Electronic Notes in Theoretical Computer Science*, *329*, 61–77. https://doi.org/10.1016/j.entcs.2016.12.005

Ritu, & Kaushal, R. (2015, December 1). *Role of handshaking packets in improving peer to peer BotNet detection*. IEEE Xplore. https://doi.org/10.1109/CoCoNet.2015.7411249

Shyu, M., Chen, S.-C., Sarinnapakorn, K., & Chang, L. (2003). A Novel Anomaly Detection Scheme Based on Principal Component Classifier. *Undefined*. https://www.semanticscholar.org/paper/A-Novel-Anomaly-Detection-Scheme-Based-on-Principal-Shyu-Chen/dcb207ce848b358aeb2e4698c9ea1ad273ce98db

Appendix

A

Commands used in splunk:

source="1_training_data.csv"| top limit=20 src_ip

source="1_training_data.csv"| top limit=20 dst_ip

source="1_training_data.csv"| top limit=20 protocol

B

There were 16,383 source and destination IP's predicted, as the table could not be fit in the word page, the results are included in a csv file called 'predicted_anamoly_stream_ids_variance_threshold-results.csv'. 89 % of the anomalies were detected, i.e., 2640 anomalies out of 2962 in the test dataset.