

U.S. ARMY COMBAT CAPABILITIES DEVELOPMENT COMMAND
C5ISR CENTER

Analyzing Feature Relevance for Social Media Traffic Classification with Machine Learning

Johnson John, Bela Erdelyi, Metin Ahiskali
U.S. Army, Combat Capabilities Development Command (CCDC)
C5ISR Center
Aberdeen Proving Ground, Maryland, USA
usarmy.apg.ccdc-c5isr.mbx.i2wd-co2-public-contact@mail.mil

13 August 2020

Release 1.0.1

DISTRIBUTION A: APPROVED FOR PUBLIC RELEASE

Table of Contents

I. Introduction 1

II. Background 1

III. Investigation Results 4

A. Feature Profiling 4

B. Feature Profile Comparison 8

C. Feature Selection Methods 10

Chi-Squared 10

Pearson Correlation 11

Model-based Feature Importance 12

Recursive Feature Elimination 14

Principal Component Analysis 14

D. Feature Selection Analysis 16

WA Feature Selection Analysis 16

TW Feature Selection Analysis 16

IV. Conclusions 17

V. References 19

I. Introduction

Prior research completed the in classification of Social Media traffic was able to successfully demonstrate the feasibility of classifying Social Media (SM) network traffic using traditional Machine Learning (ML) techniques, on a packet-by-packet basis. This paper builds on these results and evaluates feature analysis methods which were explored during the ML experiments.

Previously, an exhaustive search to evaluate nearly all possible combinations of input features to find the best subset for the implementation of Support Vector Machine (SVM) models was utilized. Exhaustive feature searches tend to be computational expensive and perhaps even prohibitive for large feature sets. In one case, up to sixteen features were explored, which required 65,535 combinatorial executions. While the project had the benefit of having access to High Performance Computing (HPC) resources, such types of computing resources may not always be available to every project. The potential application of enhanced feature analysis and feature selection techniques could result in an optimum subset of features in the development of ML models, hence reducing computation time and avoiding overfitting[1]. This ML project provides a unique opportunity to evaluate such feature reduction techniques and compare the results to its exhaustive search process.

For classification problems, the underlying goal of a machine learning model is to find information within the input data to produce the best prediction. The selected approach for investigating the results from the original SM ML project will be to analyze the input data features as to identify unique characteristics that may have contributed to the ML model success in Section A and B under the investigation results. Section C and D will then explore automated feature selection techniques on the larger feature set. This was done in order to distinguish if a correlation exists with the features identified in the best performing ML models which were obtained through an exhaustive search.

II. Background

The prior research explored the feasibility of applying traditional ML techniques for encrypted Social Media (SM) traffic classification at the single packet level. Five different SM services were identified for this research, namely Skype, Telegram, Twitter, Viber, and WhatsApp, hereinafter referred to as SK, TL, TW, VB, and WA, respectively. The applied ML techniques consisted of supervised SVM and unsupervised clustering, specifically clustering with K-means and Gaussian Mixture Model (GMM). However, the prior research eventually concluded that unsupervised clustering was ineffective for the classification of SM traffic and that the supervised Radial Basis Function (RBF) kernel SVM (SVM-RBF) technique showed relatively good performance across the SM experiments. For simplicity, the ML project overview and results provided in this report will focus primarily on the WhatsApp (WA) and the Twitter (TW) services to summarize the SVM-RBF model experiments for investigation of the subject topic.

Traffic collection for five SM services (SK, TL, TW, VB, and WA) was conducted on 5 September 2018 for the research purposes described in the prior work. The same network traffic data is here used for the purposes of feature analysis. Data preprocessing consisted of feature selection based on learned characteristics, removal of irrelevant packets, and normalization for standardization. All of which was applied via off-line processing prior to submitting datasets for ML consumption.

[1] Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data.

Data characterization efforts performed in 2018 resulting in the reliable identification of SM service packets by attributable IP source or destination addresses. Targeted SM services were identified to exclusively utilize TCP and UDP as the transport layer. As a result, non-IP based network packets are removed from the dataset collection. At that time, SM service identification could be made based by simple characterization of TCP/UDP ports. The feature extraction process leveraged such network header identifiers in order to properly label the data collection according to service types. The labels “SK”, “TL”, “TW”, “VB”, and “WA” are applied as it corresponds to each SM service packet in the data collection. Packets not identifiable to any of the SM services are then labeled as “XX” (i.e. as other traffic). It should be noted that labeling is done to a single SM service at a time and against all other collected traffic. For instance, with WA assessments the feature extraction labels as “WA” the WA SM traffic while all other traffic being labeled as “XX”. In this case, “XX” is inclusive of the other four SM traffic (i.e. SK, TL, TW, and VB) plus any ancillary non-SM traffic as well. The process equivalently repeats for each of the other four SM assessments.

In addition to the removal of non-IP packets from the data collection, the feature extraction process removed any packet without a payload, since TCP sessions have been observed to contribute about 50% of no payload traffic (e.g. ACK and FIN packets). These payload-empty packets were determined not to contribute any significant information to ML identification processes, but merely introduce noise and increase computation times.

Upon defining these exclusionary rules, feature set definition became the identification of the remaining available elements contained within each traffic packet. Without specifically considering any of the five targeted SM services, the following feature candidates were identified.

1) Payload-Size is auto scaled while provided as a payload byte count for every captured packet. Packets with no payload are assigned a value of 0.
2) Protocol, provided as a categorical feature, is encoded as 1 for TCP: 2 for UDP; 3 for ICMP; and 0 for all other packet types. While ICMP packets were encoded, these were not included in the analysis process as obviously not containing SM traffic.
3) TCP-Payload-Size is auto scaled while provided as a payload byte count of every TCP captured packet. Non-TCP packets, or those with no payload, are assigned a value of 0. This feature is a subgroup of Payload-Size and used to emphasize the TCP protocol.
4) UDP-Payload-Size is auto scaled while provided as a payload byte count of every UDP captured packet. Non-UDP packets, or those with no payload, are assigned a value of 0. This feature is a subgroup of Payload-Size and used to emphasize the UDP protocol.
5) Size-Cat, provided as a categorical feature, is defined at the payload size boundaries for TCP and UDP packets. The 12 size categories, each encoded as a value from 0 to 11, were observed to represent clear packet size breakpoints. This feature is a subgroup of Payload-size and is intended to explore the effects of categorical encoding.
6) TCP-Size-Cat, provided as a categorical feature, is defined at the payload size boundaries for TCP packets only. The 8 size categories, each encoded as a value from 0 to 7, were observed to represent clear TCP packet size breakpoints. Similar to Size-Cat, this feature is a subgroup of Payload-size and is intended to explore the effects of categorical encoding.
7) UDP-Size-Cat, provided as a categorical feature, is defined at the payload size boundaries for UDP packets only. The 6 size categories, each encoded as a value from 0 to 5, were observed to represent clear UDP packet size breakpoints. Similar to Size-Cat, this feature is a subgroup of Payload-size and is intended to explore the effects of categorical encoding.

8) TCP-Flags is represented as an integer value contained in the TCP protocol flag field, or as 0 for non-TCP packets.
9) TCP-Flags-Push is represented as 1 when the PUSH flag of a TCP packet is set, or as 0 when not set to include non-TCP packets.
10) Entropy of payload bytes is expressed as a 5-digit precision float normalized to log256 in order to encode a value between 0 and 1.
11) TLS, provided as a categorical feature, is encoded as 1 for TLS handshake record types (i.e. non-Application Data record types), 2 for TLS Application Data record types, and 0 for non-TLS packets regardless of protocol.
12) TLS-Version, provided as a categorical feature, is encoded as 1 for version 1.0, 2 for version 1.1, 3 for version 1.2 and 0 for non-TLS packets.
13) TLS-Size-Cum is auto scaled while it represents the cumulative sum of all TLS record sizes found within a single payload. Each TLS payload may carry one or more TLS records, each preceded by plain text record header declaring the record size value within the payload.
14) Byte-1, Byte-2, and Byte-3 are three feature elements corresponding to the first three bytes of a payload regardless of protocol. These features were originally formulated to capture a specific observed behavior by one of the SM applications, where a 3-byte payload is used to signal the start of a conversation and to communicate the expected next message size.

In our prior research, we established several constraints. Specifically, ML classification based on IP or on transport layer header identifiers (i.e. IP address or TCP/UDP port) was to be avoided. Further, traffic identification by DNS resolution, or derived from flow information was not permissible either. The primary objective is to determine the feasibility of SM traffic identification on a packet-by-packet basis. Flow-based analysis of network traffic has been well studied in the avenue of malware detection; this research effort focuses on the feasibility to rapidly classification an application based on a single packet.

Finally an exhaustive approach was used to execute all combinations of features to find the best performing binary classification model for each SM service through training and testing of the dataset using an 80/20 (Train/Test) split. For simplicity, performance is generalized as being measured by the Accuracy[2] metric. For further details and discussion on other utilized performance metrics, refer to [1]. The exhaustive search technique will generally always find the best possible features to build a model because it searches every possible combination of features and finds the combination that returns the best performance from the applied ML algorithm. The only drawback being that it is very time and resource intensive to train all the combinations in order to find the optimal model.

The exhaustive investigation over 16 features for WA experiment execution required 65,535 runs to exercise all possible feature combinations. The resulting best performing SVM-RBF model for WA consisted of a subset of 10 features to produce an accuracy of 81.5% for the dataset. Meanwhile, an exhaustive investigation for TW was conducted with a reduced 10 features due to learned characteristics and required only 1,023 runs to exercise all possible feature combinations. The resulting best performing SVM-RBF model for TW consisted of a subset of 6 features to produce an accuracy of 83.1% with the dataset. Overall, the supervised SVM-RBF algorithm has shown relatively good performance across the experiments for each service with TW and WA having the highest accuracies. Table 1 shows the highest obtained accuracy scores as a percentage in order to summarize the experimental results for WhatsApp (WA) and Twitter (TW). Further analysis included performing a search for the top performing SVM models to find the features that appeared most frequently. This was then used to rank the identified features as provided in the table.

[2] Classification Accuracy is the ratio of number of correct predictions to the total number of input samples.

	WA	TW
Input Dataset	September 2018 Dataset	
Testing Accuracy	81.5%	83.1%
Number of Features	10	6
Features by Ranking	1 TLS-Size-Cum 2 Byte-3 3 Byte-1 4 TLS 5 Entropy 6 Byte-2 7 Payload-size 8 TCP-size-cat 9 TCP-Flags-Push 10 UDP-payload-size	1 TLS-Size-Cum 2 TLS-Version 3 TCP-size-cat 4 Entropy 5 Payload-size 6 TCP-Flags

Table 1. SVM-RBF Model Results for WA and TW Traffic Classification

The ML project experiments were implemented with software code written in the Python 3.7 programing language. The key libraries utilized included the NumPy and Pandas python libraries for data engineering and the SciKit-Learn python library for ML tasks (i.e. SVM Classification, K-Means Clustering).

III. Investigation Results

A. Feature Profiling

The first objective of the investigation was to conduct some exploratory data analysis (EDA) to summarize the September 2018 dataset. This processes allows for the exploration of the dataset structure, the variables, and their relationships. Pandas provides a variety of libraries for loading and processing data in Python, but gathering descriptive statistics can be a tedious process. Luckily there are external libraries that exist to perform all of the data processing. Pandas-profiling[3] (version 1.4.1) is one of those libraries that was utilized as it provides out-of-the-box statistical profiling and it works as an extension of the pandas DataFrame[4] to generate a profiling report in HTML format. This library generates a complete report for a given dataset, which includes basic data type information, descriptive statistics (mean, average, etc.), quantile statistics (for frequency distributions), histograms (for visualizing distributions) and correlations (for identifying relationships).

This section, walks through some of the various statistics obtained with Pandas-profiling which allows to better understand the dataset. First, the summary statistics of WA data profile are presented, as an example, and then statistics for all the services are compared in order to provide some insight into how they may affect the model's prediction capabilities.

Table 2 provides some general WA data statistics provided in the generated profiling report, which includes the variable type, count of distinct values, mean, minimum, maximum, and percent of zero values for each feature. The profiling report for the WA dataset consisted of 12 feature variables with 65,075 observation records. Note that only 12 of the 16 features were analyzed because variants of TCP-payload-size and UDP-payload-size were excluded to avoid analyzing redundant features.

[3] Pandas-Profiling, 2016 MIT License, GitHub repository, https://github.com/pandas-profiling/pandas-profiling.
[4] A DataFrame, provided as a class in the Pandas library, is defined as a two-dimensional size-mutable tabular data structure with labeled axes (rows and columns).

Feature/Type	Statistic	Value
Byte-1 Numeric	Distinct Count	256
	Mean	0.46264
	Minimum	0
	Maximum	1
	Zeros (%)	7.0%
Byte-2 Numeric	Distinct Count	256
	Mean	0.4719
	Minimum	0
	Maximum	1
	Zeros (%)	6.1%
Byte-3 Numeric	Distinct Count	256
	Mean	0.31509
	Minimum	0
	Maximum	1
	Zeros (%)	22.5%
Entropy Numeric	Distinct Count	9455
	Mean	0.90367
	Minimum	0.11479
	Maximum	0.98766
	Zeros (%)	0.0%
TCP-Payload-Size Numeric	Distinct Count	583
	Mean	759.53
	Minimum	0
	Maximum	1326
	Zeros (%)	32.0%
UDP-Payload-Size Numeric	Distinct Count	1038
	Mean	136.36
	Minimum	0
	Maximum	1146
	Zeros (%)	68.0%
TCP-Flags Numeric	Distinct Count	4
	Mean	11.745
	Minimum	0
	Maximum	25
	Zeros (%)	32.0%
TCP-Flags-Push Boolean	Distinct Count	2
	Mean	0.10717
TLS Categorical	Variable is highly correlated to *TCP Payload-Size.* Correlation 0.92619	
TLS-Size-Cum Numeric	Distinct Count	336
	Mean	703.37
	Minimum	0
	Maximum	16878
	Zeros (%)	86.5%
TLS-Version Numeric	Distinct Count	3
	Mean	0.071656
	Minimum	0
	Maximum	3
	Zeros (%)	97.4%

Table 2. Statistical Summary for WA Dataset

The table shows that the WA data has 1038 unique values in the UDP-Payload-Size data, while there are only 583 unique values in the TCP Payload-Size data as shown by the "Distinct Count" statistic. This TCP to UDP

unique count ratio for the WA data differs from all the other applications where in most cases the unique TCP record count is similar or greater than the unique UDP record count, as summarized in Table 3.

	WA	TW	SK	TL	VB	XX
TCP Distinct Count	583	1323	1232	402	866	1312
UDP Distinct Count	1038	0	1135	19	614	1111

Table 3. TCP and UDP Payload Size Distinct Count Summary

Figure 1 below is a histogram of the TCP-Payload-Size data distribution, which shows a good majority of the data has a value of either 1326 bytes or 0 bytes, which is assigned Non-TCP packets, or UDP packets in this case. The limited variability in TCP payload sizes is an indicator that payload size alone would not be very good contributing feature for classifying TCP records.

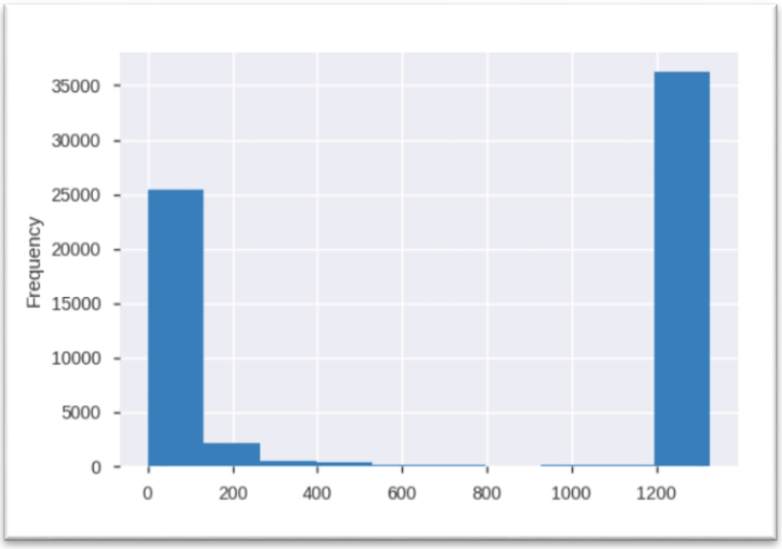

Figure 1. TCP-Payload-Size Feature Histogram for WA Data

Figure 2 below is a histogram of the UDP-Payload-Size data distribution, which shows that the UDP payload sizes are more dispersed between the ranges of 1 to 1146 bytes, with a good concentration below 250 bytes, roughly. In this case, the distribution of sizes may indicate that UDP payload size would be a good contributing feature for classifying UDP records. Note that the observations with 0 bytes, which is assigned Non-UDP packets, represent TCP packets.

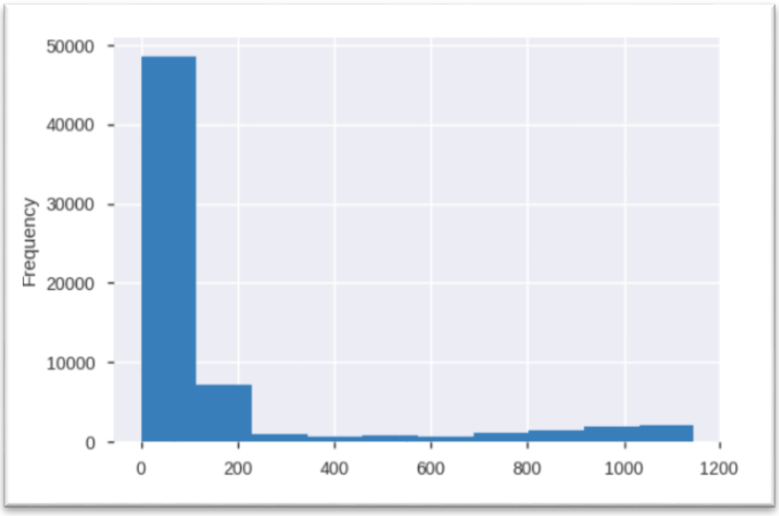

Figure 2. UDP-Payload-Size Feature Histogram for WA Data

Figure 3 below shows a visual representation of the correlation matrix using Pearson Correlation[5] method. The matrix provides a look at the correlations between all different variable pairs. Note that the diagonal elements (dark red) are the correlations between each variable and itself (value of 1.00). Also note that only correlations either above or below the diagonal need to be considered as they mirror each other.

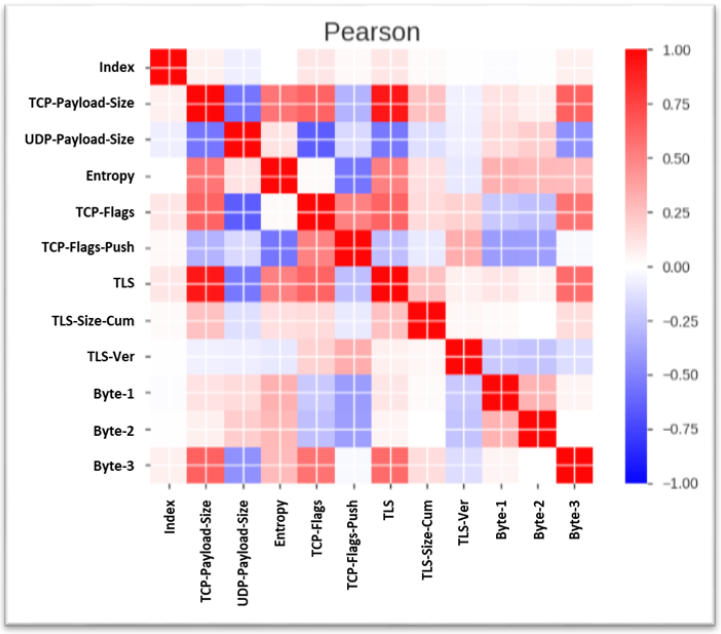

Figure 3. Pearson Correlation Matrix for WA Features

The figure and the details in Table shows that the TLS feature data is rejected from the analysis because it is highly correlated with TCP Payload-Size feature with a correlation of 92.62%. After taking a closer look at Figure 4 for the top common values, it makes sense that a majority of records either have a TCP Payload-Size of 1326 bytes (54.4%) which will likely correlate to a TLS value of 2 (TLS application data) or a TCP Payload-Size of 0 (32.0%) which will likely correlate to a TLS of 0 (indicating UDP records).

[5] Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y.

Statistics | Histogram | Common Values | Extreme Values

Value	Count	Frequency (%)
1326	35431	54.4%
0	20794	32.0%
3	1133	1.7%
256	1021	1.6%
1321	505	0.8%
31	308	0.5%
59	295	0.5%
39	283	0.4%
34	261	0.4%
51	195	0.3%
Other values (573)	4849	7.5%

Figure 4. Top 10 Common Values in TCP-Payload-Size for WA Data

Additionally, Figure 3 of the correlation matrix shows that the TCP-Flags and Entropy features show a relatively high linear correlation to TCP Payload-Size which may be an indicator of how well they may aggregate with TCP payload size to contribute in the classification of TCP records.

B. Feature Profile Comparison

Common statistics from each SM service for three features, namely TCP-Payload-Size, TCP-Flags, and TLS-Size-Cum, are compared in order to visualize how data is distributed among the services. Figure 5 below shows the combined top 10 common values from the TCP-Payload-Size feature for each service. The chart reveals that 31 of the 39 most common TCP payload sizes are unique to only one service, while 5 more sizes are unique to 2 or 3 of the services. Of course this is a generalization of only the top common values for each service, however, it is still a good indication of how valuable the payload size or even one of its variations can be for the classification process.

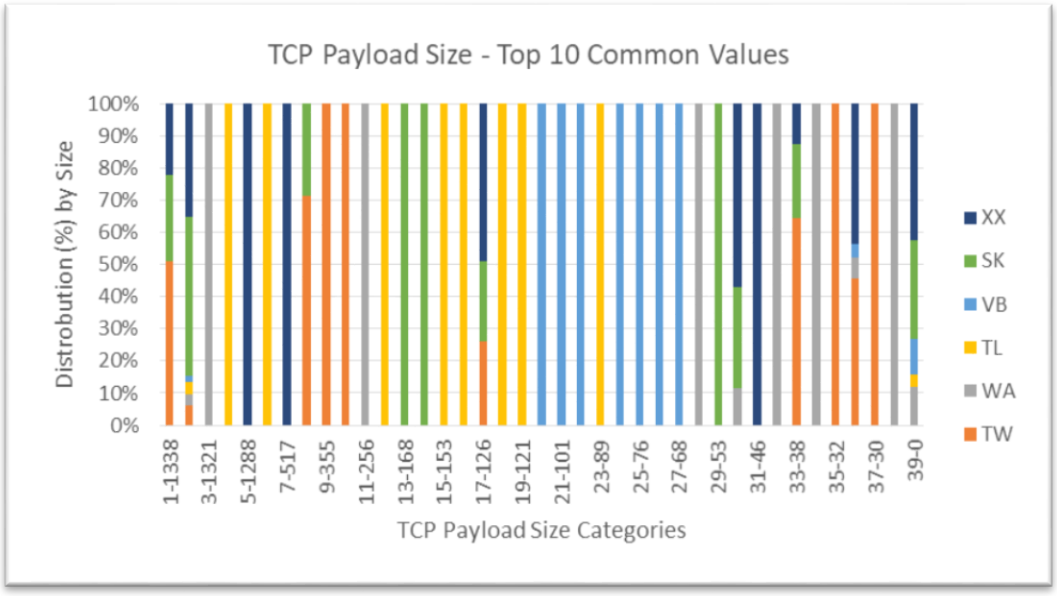

Figure 5. Distribution Chart of Top 10 Common Values for TCP-Payload-Size Data

Figure 6 below shows the combined distribution of values from the TCP-Flags feature for each service. The only significant information that can be derived is that the flag value of 17 (or category 3) is unique to only other traffic (or 'XX'). However, it does confirm that the 4 out of the 5 different flag values are used by most if not all the services, which may prove useful when aggregated with other TCP-type features for classification of TCP records.

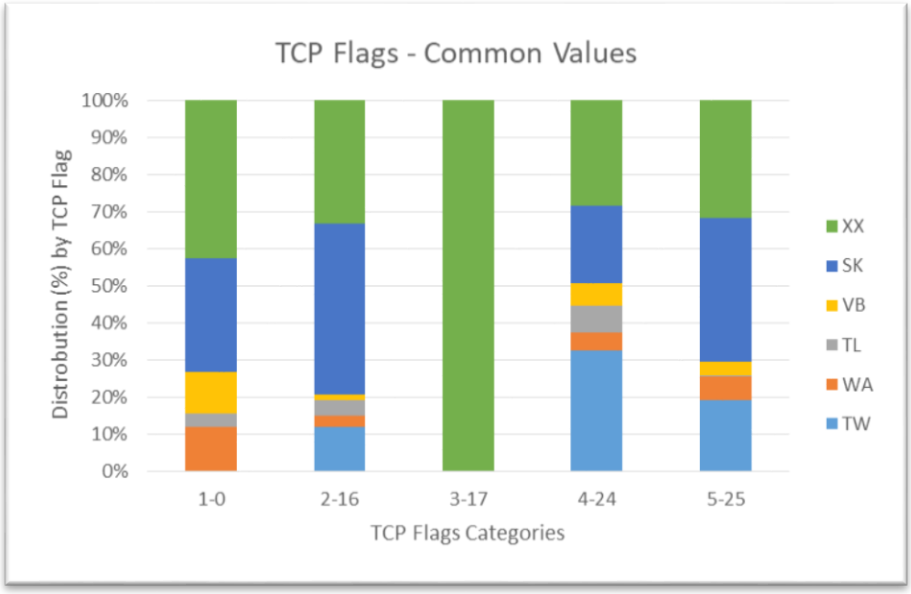

Figure 6. Distribution Chart of Top 10 Common Values for TCP-Flags Data

Figure 7 below shows the combined top 10 common values from the TLS-Size-Cum feature for each service. The chart shows that 21 of the 31 TLS sizes are unique to only one service, while 7 more sizes are unique to 2 or 3 of the SM services. Once again, this is a generalization of only the top common values, but it is still a good indication of how valuable the TLS size cumulative data can be for the classification process.

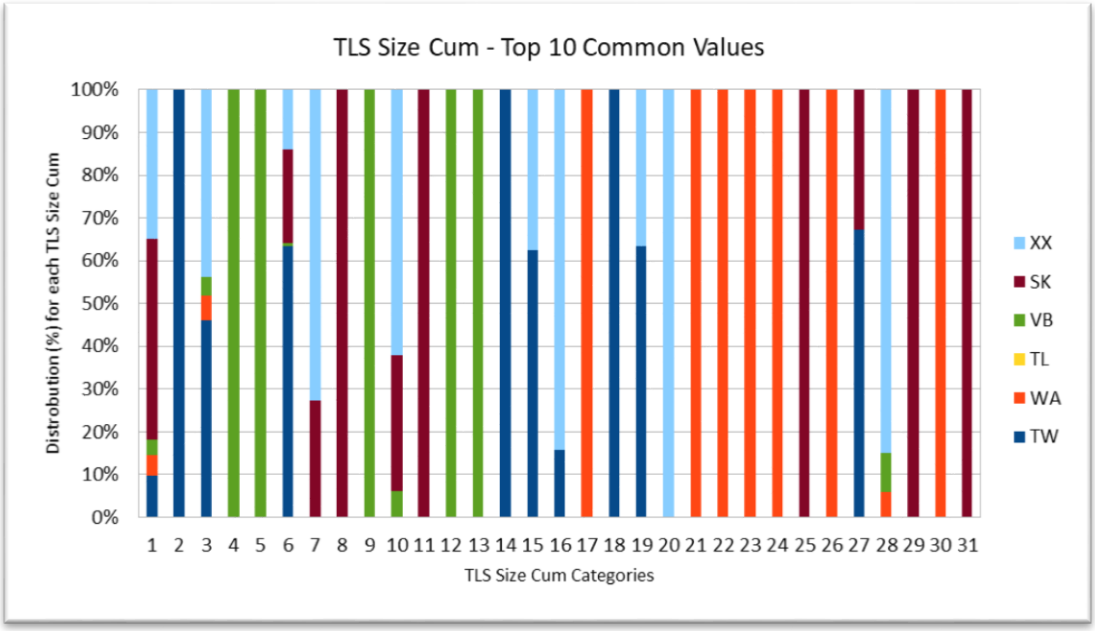

Figure 7. Distribution Chart of Top 10 Common Values for TCP-Size-Cum Data

Based on the EDA results, some interesting statistics and unique characteristics within the feature profiles has been identified. This allows to understand how the aggregate of features, specifically TCP-Payload-Size, TCP-Flags, and TLS-Size-Cum, can be useful in the ML process where a set of hierarchical decisions will eventually give us a final classification prediction, like in a decision tree model.

C. Feature Selection Methods

As mentioned previously, the secondary goal of the investigation is to experiment with various feature selection techniques to determine feature relevance and whether a correlation can be made with the features identified by an exhaustive search with the SVM-RBF model, as shown in Table 1. Features selected to train ML models have a large influence on prediction performance. Specifically, having irrelevant features in the data could decrease the accuracy of a model causing overfitting and a lack of generalization.

Most feature selection methods can be divided into three major buckets; filter based, wrapper based, and embedded methods. Filter based methods use some statistical metric to filter features, such as correlation coefficient and Chi-squared (chi^2). Wrapper based methods consider the selection of a set of features as a search problem, such as Recursive Feature Elimination (RFE). Embedded methods use algorithms that have built-in feature selection methods, such as tree-based classifiers that have their own feature importance methods. The following automated feature selection techniques were explored.

Chi-Squared

The Scikit-learn library provides the *SelectKBest* class that is used with a suite of different statistical tests to select those features that have the strongest relationship with the target variable. The chi^2 statistical test, a filter based method, was selected for classification with non-negative numerical and categorical features. This method calculates the chi-square metric between the target and the numerical variable and only selects variables with the maximum chi-squared values.

The chi² technique was applied via the *SelectKBest* class to the WA dataset with parameters including a scoring function equal to the chi-squared and the number of features to select, 'k' equal to 5. Based on the results, the top 10 relevant features were identified as UDP-Payload-Size, TCP-Payload-Size, Payload-Size, UDP-Size-Cat, TCP-Flags, TCP-Size-Cat, Size-Cat, TLS-Version, TLS-Size-Cum, and Protocol, in that order, as shown in Table 4.

Model Input	SKB Parameters	Feature Relevance – chi2 metric
September 2018 Dataset **Labels Key:** **{'xx': 0, 'wa': 1}** **Value Counts:** **0 1410696** **1 65075**	• score_func=chi2 • k=5	1 UDP-Payload-Size 9.855e+06 2 TCP-Payload-Size 5.016e+06 3 Payload-Size 2.445e+06 4 UDP-Size-Cat 3.709e+04 5 TCP-Flags 3.568e+04 6 TCP-Size-Cat 1.926e+04 7 Size-Cat 1.528e+04 8 TLS-Version 5.971e+03 9 TLS-Size-Cum 4.908e+03 10 Proto 1.928e+03 11 TLS 1.751e+03 12 Byte-3 1.614e+03 13 Byte-2 1.119e+02 14 Entropy 7.821e+01 15 TCP-Flags-Push 5.278e+01 16 Byte-1 5.991e+00

Table 4. Chi-squared Feature Relevance Analysis for WA Data

Overall these results did not correlate with the set of features identified in the SVM-RBF model results of Table 1. Specifically, most of the top contributors from the SVM-RBF analysis are ranked low on the list for the chi-squared test, such as TLS, Byte-1, Byte-2, Byte-3, and Entropy.

Pearson Correlation

Another filter based selection technique is the Pearson correlation coefficient which is also known as the "product moment correlation coefficient" (PMCC) or simply "correlation". A Pearson correlation is a number between -1 and 1 that indicates the extent to which two variables are linearly related, where a negative number implies a negative or inverse relationship. It is obtained by dividing the covariance of the two variables by the product of their standard deviations. Pearson correlation may be applied to metric variables and can be used to find features with a high correlation to the target variable or it can be used to find linearly dependent features that would have almost the same effect on the target variable. In this case, the absolute value is used of the Pearson's correlation between the target and numerical features in the dataset.

The correlation analysis was applied to the WA dataset with the default parameters where method is equal to the 'pearson'. Based on the resulting ranking, the top 10 relevant features that correlate to the target feature were identified as TCP-Size-Cat, TCP-Payload-Size, Protocol, Size-Cat, UDP-Size-Cat, TCP-Flags, Payload-Size, UDP-Payload-Size, Byte-3 and Entropy, in that order, as shown in Table 5. Note that further testing was also conducted with Spearman and Kendall methods for correlation with similar results.

Model Input	Corr Parameters	Feature Correlation – corr metric
September 2018 Dataset **Labels Key:** **{'xx': 0, 'wa': 1}** **Value Counts:** **0 1410696** **1 65075**	• method= 'pearson'	1 TCP-Size-Cat 0.144446 2 TCP-Payload-Size 0.135618 3 Proto 0.131365 4 Size-Cat 0.131240 5 UDP-Size-Cat 0.123611 6 TCP-Flags 0.112892 7 Payload-Size 0.111167

		8 UDP-Payload-Size 0.102288 9 Byte-3 0.077742 10 Entropy 0.072845 11 TLS 0.049159 12 TLS-Version 0.042896 13 Byte-2 0.020432 14 TCP-Flags-Push 0.007957 15 Byte-1 0.005259 16 TLS-Size-Cum 0.001199

Table 5. Pearson Correlation Feature Relevance Analysis for WA Data

Similar to the chi-squared test, the Pearson correlation analysis did no correlate with the top features identified in the SVM-RBF model results of Table 1, specifically because most of the top contributors from the SVM-RBF analysis are ranked low on the list for Pearson correlation, such as TLS, Byte-1, Byte-2, Byte-3, Entropy, and TLS-Size-Cum.

Model-based Feature Importance

Embedded methods that use ensembles of decision trees can compute the relative importance of each input feature. Feature importance is a built-in class that comes with tree based classifiers and here eXtreme Gradient Boost (XGBoost), an open-source library which implements the gradient boosting decision tree algorithm, is used for ranking the features in the dataset. The feature importance attribute gives a score or weight for each feature of the data, where the higher the score, the more important or relevant the feature is towards the output variable. Features that have a weight of 0 are not used by any tree splits. Presented in this section are two different parameter sets executed with the XGBoost method which produced seemingly different rankings with similar overall model accuracies. This is done to explore the potential for instability with the selected parameters on the feature importance scoring.

The initial XGBoost feature importance analysis, referred to as XGBoost-1 from here on, was run with model parameters including a learning rate of 0.5 and a maximum depth of 3 on the WA dataset. The analysis ranked the top 10 relevant features as Entropy, TLS-Size-Cum, Byte-2, Payload-Size, Byte-1, Byte-3, UDP-Payload-Size, TCP-Payload-Size and TLS, in that order, as shown in Figure 8 and Table 6. Note that only 12 or 16 features are ranked which means TCP-Flags-Push, Size-Cat, UDP-Size-Cat and TCP-Size-Cat features must have had a score of 0 and therefore have been excluded.

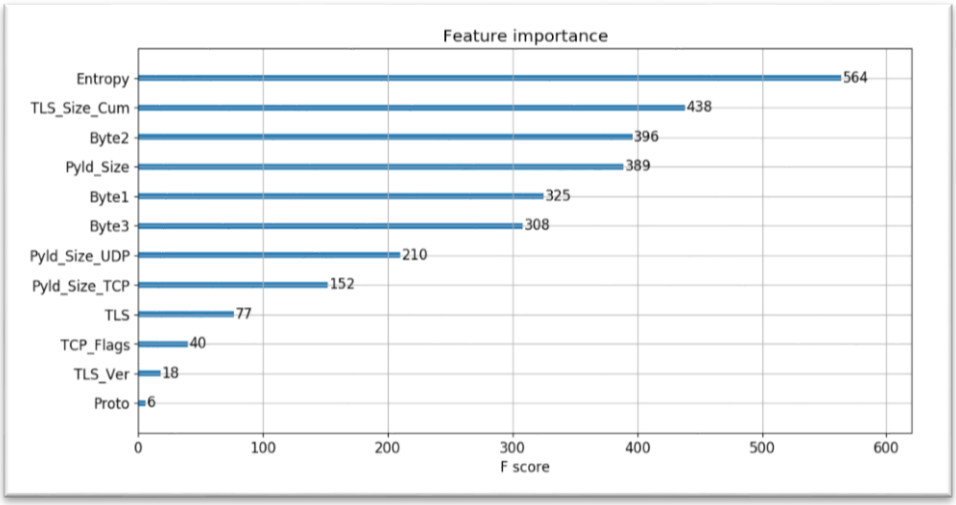

Figure 8. XGBoost-1 Feature Importance Chart for WA Data

Model Input	XGBoost Parameters	Feature Importance - score
September 2018 Dataset **Value Counts:** **'xx' 1410696** **'wa' 65075**	• learning_rate=0.5, • max_depth=3, • n_estimators=500, • min_child_weight=6, • scoring='roc_auc'	1 Entropy 564 2 TLS-Size-Cum 438 3 Byte-2 396 4 Payload-Size 389 5 Byte-1 325 6 Byte-3 308 7 UDP-Payload-Size 210 8 TCP-Payload-Size 152 9 TLS 77 10 TCP-Flags 40 11 TLS-Version 18 12 Proto 6

Table 6. XGBoost-1 Feature Relevance Analysis for WA Data

The second XGBoost feature importance analysis, referred to as XGBoost-2 from here on , was run with parameters including a learning rate equal to 0.2 and a maximum depth equal to 7 on the same WA dataset. The analysis ranked the top 10 relevant features as Entropy, Byte-2, Byte-1, Byte-3, TLS Size Cum, Payload-Size, TCP-Payload-Size, UDP-Payload-Size, TLS and TCP-Flags, in that order, as shown in Figure 9 and Table 7. Note once again that only 12 or 16 features are ranked which means TCP-Flags-Push, Size-Cat, UDP-Size-Cat and TCP-Size-Cat features must have been excluded, therefore indicating they had a score of 0.

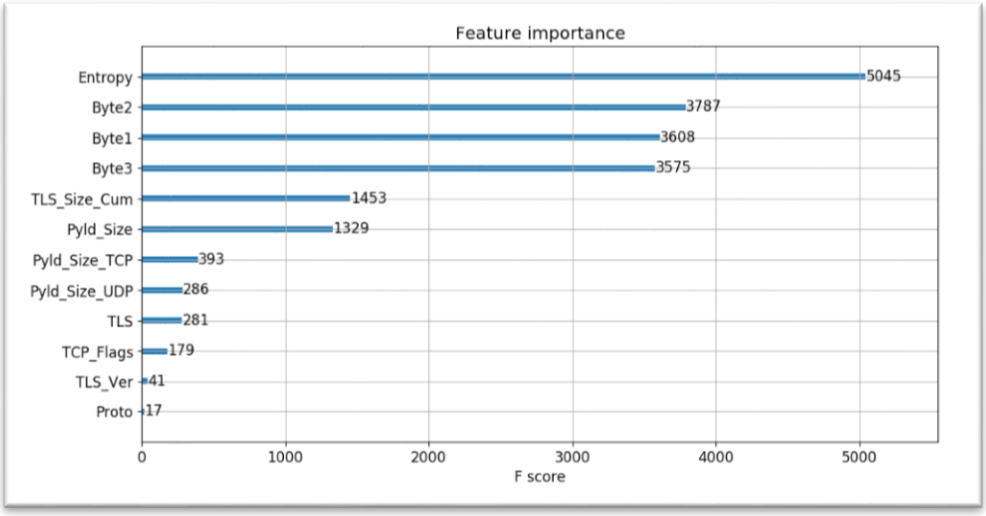

Figure 9. XGBoost-2 Feature Importance Chart for WA Data

Model Input	XGBoost Parameters	Feature Importance - score
September 2018 Dataset **Labels Key:** **{'xx': 0, 'wa': 1}** **Value Counts:** **0 1410696** **1 65075**	learning_rate=0.2, max_depth=7, n_estimators=1000, min_child_weight=1, scoring='roc_auc'	1 Entropy 5045 2 Byte-2 3787 3 Byte-1 3608 4 Byte-3 3575 5 TLS-Size-Cum 1453 6 Payload-Size 1329 7 TCP-Payload-Size 393 8 UDP-Payload-Size 286 9 TLS 281 10 TCP-Flags 179 11 TLS-Version 41 12 Proto 17

Table 7. XGBoost-2 Feature Relevance Analysis for WA Data

So while the exact ordering does not quite align between both the XGBoost-1 and XGBoost-2 analyses, the results in general do correlate with each other if the group of top 10 features is considered as a whole. Furthermore, the top 10 features for both XGBoost analyses do correlate, for the most part, with the top 10 features identified with the SVM-RBF model results in Table 1, assuming the Size-Cat features are redundant and thus interchangeable with Payload-Size, which is already included.

Recursive Feature Elimination

The RFE method is a wrapper based method which works by recursively removing features, builds a model using the remaining attributes and calculates model accuracy. This technique begins by building a model on the entire set of predictors and computing an importance score for each predictor. The least important predictors are then removed, the model is re-built, and importance scores are computed again. In this case, the Random Forrest Classifier (RFC) is used.

The RFE analysis utilizes RFC with parameters including a learning rate equal to 0.5 and a maximum depth equal to 3 which was applied to the WA dataset. Based on the resulting ranking, the top 10 relevant features were identified as Byte-2, TLS-Size-Cum, Byte-3, TCP-Flags, TLS, TCP-Payload-Size, Byte-1, UDP-Payload-Size, Payload-Size and TLS Version, in that order, as shown in Table 8.

Model Input	RFC Parameters	Feature Ranking
September 2018 Dataset **Labels Key:** **{'xx': 0, 'wa': 1}** **Value Counts:** **0 1410696** **1 65075**	• learning_rate=0.5 • class_weight='balanced' • n_estimators=100 • max_depth=3	1 Byte-2 2 TLS-Size-Cum 3 Byte-3 4 TCP-Flags 5 TLS 6 TCP-Payload-Size 7 Byte-1 8 UDP-Payload-Size 9 Payload-Size 10 TLS-Version 11 Entropy 12 Proto 13 TCP-Flags-Push 14 UDP-Size-Cat 15 TCP-Size-Cat 16 Size-Cat

Table 8. RFE Feature Ranking for WA Data

Similar to the XGBoost analysis, the top features for the RFE analysis do correlate, for the most part, with the top 10 features identified with the SVM-RBF model results of Table 1, with one main exception being that the Entropy feature was ranked 5th with the SVM-RBF model but ranked only 11th by the RFE analysis. While not explored in this investigation, it is reasonable to believe that tweaking the model parameters should provide a more complete correlation with the SVM-RBF results.

Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality-reduction method that is often used to reduce the dimension of large data sets, by transforming a large set of variables into a smaller one that still contains most of the variation in the large set. In general, PCA uses linear algebra to transform the dataset into a compressed form which reduces the number of variables, while preserving as much information as possible. The new set of variables, which are known as the principal components, or PCs, are orthogonal, such that the retention of variation present in the original variables decreases as we move down in the order. While PCA is not

specifically a feature selection technique, it can help identify the most important variables in the original feature space.

The PCA analysis was applied to the WA dataset with a parameter of 5 principal components. But to get a read on the actual information that individual features contribute to the PCA's, the data was first normalized and evaluated the mean and variance of each feature contribution being aggregated over all principal components, as shown in Figure 10. Given this result features were ranked based on their absolute mean as identified in Table 9, where the top 10 features include TLS-Size-Cum, TCP-Flags, Byte-1, Protocol, UDP-Size-Cat, TCP-Flags-Push, UDP-Payload-Size, Byte-2, TCP-Size-Cat and TCP-Payload-Size, in that order. The thought being that a feature that doesn't have much variability cannot be of much use as a distinguishable characteristic and thus is not as important as more variable feature.

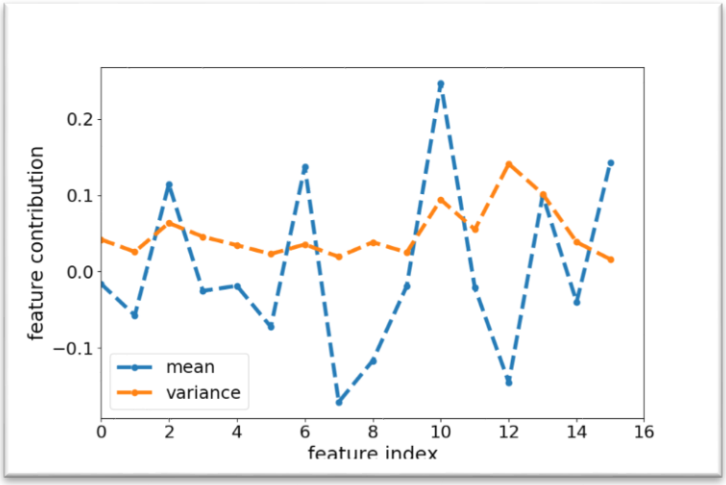

Figure 10. PCA Analysis Chart for WA Data

Model Input	PCA Parameters	Feature Ranking – mean (abs)	
September 2018 Dataset **Labels Key:** **{'xx': 0, 'wa': 1}** **Value Counts:** **0 1410696** **1 65075**	• n_components=5	1 TLS-Size-Cum	0.25
		2 TCP-Flags	0.18
		3 Byte-1	0.15
		4 Proto	0.15
		5 UDP-Size-Cat	0.14
		6 TCP-Flags-Push	0.12
		7 UDP-Payload-Size	0.12
		8 Byte-2	0.12
		9 TCP-Size-Cat	0.08
		10 TCP-Payload-Size	0.06
		11 Entropy	0.04
		12 Byte-3	0.04
		13 Size-Cat	0.03
		14 Payload-Size	0.02
		15 TLS	0.02
		16 TLS-Version	0.02

Table 9. PCA Feature Ranking for WA Data

While some of the top ranked features identified in the PCA analysis did correlate with the SVM-RBF model results in Table 1, there were 3 specific features, namely Entropy, Byte-3 and TLS, which were ranked low in the PCA analysis but ranked high for the SVM-RBF model.

D. Feature Selection Analysis

WA Feature Selection Analysis

For comparison, Table 10 provides a summary of the feature ranking analysis for the WA dataset features based the analysis method utilized. In the table, the features have been color coded randomly to better visualize any potential correlation between the features in each method (by columns).

Rank	SVM Search	Xgboost-1 Importance	Xgboost-2 Importance	RFE	Chi-squared	PCA	Pearson
1	TLS-Size-Cum	Entropy	Entropy	Byte-2	UDP-Payload-Size	TLS-Size-Cum	TCP-Size-Cat
2	Byte-3	TLS-Size-Cum	Byte-2	TLS-Size-Cum	TCP-Payload-Size	TCP-Flags	TCP-Payload-Size
3	Byte-1	Byte-2	Byte-1	Byte-3	Payload-Size	Byte-1	Proto
4	TLS	Payload-Size	Byte-3	TCP-Flags	UDP-Size-Cat	Proto	Size-Cat
5	Entropy	Byte-1	TLS-Size-Cum	TLS	TCP-Flags	UDP-Size-Cat	UDP-Size-Cat
6	Byte-2	Byte-3	Payload-Size	TCP-Payload-Size	TCP-Size-Cat	TCP-Flags-Push	TCP-Flags
7	Payload-Size	UDP-Payload-Size	TCP-Payload-Size	Byte-1	Size-Cat	UDP-Payload-Size	Payload-Size
8	Size-Cat	TCP-Payload-Size	UDP-Payload-Size	UDP-Payload-Size	TLS-Version	Byte-2	UDP-Payload-Size
9	TCP-Flags	TLS	TLS	Payload-Size	TLS-Size-Cum	TCP-Size-Cat	Byte-3
10	TCP-Size-Cat	TCP-Flags	TCP-Flags	TLS-Version	Proto	TCP-Payload-Size	Entropy
11	TCP-Payload-Size	TLS-Version	TLS-Version	Entropy	TLS	Entropy	TLS
12	UDP-Payload-Size	Proto	Proto	Proto	Byte-3	Byte-3	TLS-Version
13	UDP-Size-Cat	Size-Cat	Size-Cat	TCP-Flags-Push	Byte-2	Size-Cat	Byte-2
14	Proto	TCP-Size-Cat	TCP-Size-Cat	UDP-Size-Cat	Entropy	Payload-Size	TCP-Flags-Push
15	TCP-Flags-Push	UDP-Size-Cat	UDP-Size-Cat	TCP-Size-Cat	TCP-Flags-Push	TLS	Byte-1
16	TLS-Version	TCP-Flags-Push	TCP-Flags-Push	Size-Cat	Byte-1	TLS-Version	TLS-Size-Cum

Table 10. Feature Ranking Comparison of Feature Selection Methods for WA Data

In general, both XGBoost analyses and the RFE analysis came the closest to correlating to the top performing features identified by the SVM-RBF model for SM traffic classification. Meanwhile, the Chi-squared, PCA and Pearson Correlation methods had much lower correlation to the model. This supports the theory that these analysis methods may be useful in conducting feature selection for finding the optimal features to be used in a ML model in lieu of conducting an exhaustive search of all the feature combinations.

TW Feature Selection Analysis

While the detail have been omitted, a similar analysis was also conducted for the TW dataset using the same process as described for the WA dataset. Table 11 provides a summary of the feature ranking analysis for the TW dataset features based the analysis method utilized. Once again, the features have been color coded randomly to better visualize any potential correlation between the features in each method.

Rank	SVM Search	Xgboost-1	Xgboost-2	RFE	Chi-squared	PCA	Pearson
1	TLS-Size-Cum	TLS-Size-Cum	Entropy	TLS-Version	Payload-Size	TLS-Size-Cum	TLS-Version
2	TLS-Version	Entropy	Byte-1	Payload-Size	TLS-Version	Byte-3	TCP-Flags-Push
3	TCP-Size-Cat	Payload-Size	Byte-2	TLS-Size-Cum	TCP-Flags-Push	Byte-1	TCP-Flags
4	Entropy	TCP-Size-Cat	Byte-3	TCP-Flags	TLS-Size-Cum	TCP-Flags	Payload-Size
5	Payload-Size	Byte-1	TLS-Size-Cum	Byte-1	TCP-Flags	TCP-Flags-Push	Entropy
6	TCP-Flags-Push	Byte-3	Payload-Size	Byte-3	Byte-3	TLS-Version	Byte-3
7	TCP-Flags	Byte-2	TCP-Size-Cat	Entropy	Byte-2	Byte-2	Byte-2
8	Byte-2	TLS-Version	TCP-Flags	Byte-2	Byte-1	Payload-Size	Byte-1
9	Byte-3	TCP-Flags	TCP-Flags-Push	TCP-Flags-Push	TCP-Size-Cat	TCP-Size-Cat	TCP-Size-Cat
10	Byte-1	TCP-Flags-Push	TLS-Version	TCP-Size-Cat	Entropy	Entropy	TLS-Size-Cum

Table 11. Feature Ranking Comparison of Feature Selection Methods for TW Data

Unlike the previous results, it would be hard to explain how any of the analysis methods could correlate to the top 6 performing features identified by the SVM-RBF model for SM traffic classification. So while an argument could be made for XGBoost-1, RFE and the Chi-squared analysis methods, they all are missing one or two features in the top 6 features that showed to be relevant by the model. However, it is also possible that a more thorough search of the selection method parameters could produce a relevant feature ranking, but that is outside the scope of this investigation and would still require more computational time. This further supports the understanding that because the ranking order of the top ranking features is quite unstable between the different methods, it is important to carefully determine the cutoff criteria in order to avoid removing a critical feature out of the ML experimentation.

IV. Conclusions

In this report, multiple filter, wrapper, and embedded methods were explored for feature selection with also the inclusion of PCA. The investigation shows that the embedded and wrapper methods based on predictive models were found to be effective in finding relevant features that correlated with the SVM-RBF model results for classification of the WA dataset, while the filter and PCA methods proved to be less effective.

In general, the results point to a limitation of filter type selection methods which look at features individually and do not take into consideration a subset of features or a particular predictive model. This leads to the conclusion that filter and PCA methods are good for gaining a better understanding of data, but not necessarily good for optimizing the feature set.

Specifically, the investigation found XGBoost model-based feature importance and RFE methods to be effective methods for feature selection when correlated to the SVM-RBF model results for the WA dataset, which was obtained through an exhaustive search. However, with the TW dataset, these same methods were evaluated to be much less effective, that is without conducting a more rigorous search of the applied model parameters. This leads to the conclusion that while feature selection techniques based on predictive models could be used to reduce the feature space for the ML process, it is not guaranteed to provide similar results as an exhaustive feature search without a reduction in performance. It is postulated that an exhaustive search of optimum parameters for feature selection methods could produce enhanced performance. However, such is not guaranteed and only trades-off one exhaustive search for another. In essence, the degree of performance reduction warrants further investigation; such is beyond the scope of this report and should be considered for future work.

In summary, there are various methods available to approach the task of feature selection, however each method on their own can be quite unstable based on the input data and parameters. In some cases, the selected feature selection method was shown to provide good correlation to the features identified in the best predictive model, while in other cases it was hard to find a definitive correlation between the set of top ranking features. Thus contributing to the theory, that while the information needed to select the best features for a ML model may be extracted through multiple methods and models, finding the optimal method and parameter can be difficult to distinguish, and may result in a loss of information relevant to the classification task.

V. References

1. R. Duangsoithong and T. Windeatt, “Relevant and Redundant Feature Analysis with Ensemble Classification”, Seventh International Conference on Advances in Pattern Recognition, 2009.

2. A. Blum and P. Langley, “Selection of relevant features and examples in machine learning”, Artificial Intelligence Volume 97 (Issue 1–2), 1997.

3. V. Kumar and S. MinzFeature, “Feature Selection: A literature Review”, Smart Computing Review, vol. 4 (no. 3), June 2014.

4. J. Tang, S. Alelyani, and H. Liu, “Feature selection for classification: A review”, Data Classification: Algorithms and Applications (37), 2014.

5. A. Fahad et al., “Toward an efficient and scalable feature selection approach for internet traffic classification”, Computer Networks, Volume 57 (Issue 9) 19 June 2013

6. C. Jie, F. Zhiyi, Z. Dan, and Q. Guannan, “Network traffic classification using feature selection and parameter optimization”, Journal of Communications Vol. 10 (No. 10), October 2015.

7. Johannes Otterbach, “Principal Component Analysis (PCA) for Feature Selection and some of its Pitfalls”, 24 Mar 2016, https://jotterbach.github.io/2016/03/24/Principal_Component_Analysis.

8. D. Gilbert, C. Mateu, J. Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges,” Journal of Network and Computer Applications Vol 153, March 2020.