A Systematic Approach of Feature Selection for Encrypted Network Traffic Classification

Donald McGaughey
Electrical and Computer Engineering
Royal Military College of Canada
Kingston, ON, Canada
mcgaughey-d@rmc.ca

Trevor Semeniuk
Strat J4 Requirements
Department of National Defence
Ottawa, Canada
trevor.semeniuk@forces.gc.ca

Ron Smith
Electrical and Computer Engineering
Royal Military College of Canada
Kingston, ON, Canada
smith-r@rmc.ca

Scott Knight
Electrical and Computer Engineering
Royal Military College of Canada
Kingston, ON, Canada
knight-s@rmc.ca

***Abstract*— In this paper we present a statistical analysis technique for classifying encrypted network traffic. The technique uses the fast orthogonal search (FOS) algorithm to select a subset of features with discriminative power from a large set of features derived from the data. A k-nearest neighbor (kNN) classifier was then used to classify the network traffic using the features selected by FOS. The FOS algorithm selected a 12-feature subset from a set of 2,839 features. A kNN classifier using these 12 features has 106 fewer errors than a kNN using an arbitrary 44-feature set and there was an 81% reduction in computation time for classification.**

Keywords— Network Traffic Classification; Network Traffic Feature Selection, Fast Orthogonal Search

I. INTRODUCTION

Most organizations allow encrypted traffic on their networks so that employees can perform transactions such as personal banking. In addition to legitimate use of encrypted network traffic organizations may inadvertently be permitting non-authorized or malicious traffic in disguise consisting of prohibited protocols, unauthorized webpages or malicious file types. Classification of encrypted traffic, whether legitimate or malicious, introduces many unique challenges, especially for cases where the protocols and port numbers are deliberately obfuscated to misrepresent the type of data being transferred.

Current network classification techniques include: port/protocol pairing, signature analysis, deep packet inspection and statistical anomaly analysis. Blocking port numbers and IP-addresses has become ineffective as most subversive applications and many commercial applications deliberately use non-standard ports to by-pass firewalls [1]. The success of deep packet inspection when used for encrypted traffic analysis has been diminishing [2]–[4]. Deep packet inspection is computational expensive and not feasible for high volume analysis [3].

Recent research has focused on statistical classification approaches which do not require access to the plain-text content of packets and have shown success when confronted with protocol obfuscation, encapsulation, and encryption [3]. Statistical analysis approaches define the characteristics of network traffic using sets of features [5] and use these features to classify the network traffic. With hundreds of traffic features that can be used in classification, it becomes imperative to select a subset of features that has predictive value.

Features have been selected manually, exhaustively or using machine learning techniques. Manual selection of features is time consuming and difficult to optimize. Exhaustive searches for feature sets are computationally infeasible due to the huge number of feature set combinations. Machine learning techniques can lead to optimizing the classifier for the training data with a poor ability to classify unseen data [6].

The objective of this research was to develop a general-purpose method of selecting feature subsets for encrypted traffic classification. In this research, a primary feature set of 44 features was extracted from network data using NetMate [7]. Additional features were derived from this primary feature set by taking the sum, difference or vector-products of the primary feature vectors. The fast orthogonal search (FOS) algorithm was used to select a subset of features from this expanded feature set for the application of classifying Dropbox [8] traffic. Then a k-nearest neighbour classifier (kNN) was used to classify previously unseen testing data. The prediction accuracy and area under the receiver-operating-curve (AUC) were used to compare the classification accuracy of the kNN classifier using: the feature subset selected by FOS; a set of arbitrary features; and a feature subset selected with the best-first algorithm.

Section II reviews the theory of network traffic analysis. Section III provides background knowledge of feature selection, including a brief description of the fast orthogonal search algorithm (FOS) and the best first (BeF) algorithm. In Section IV, the k nearest neighbour classifier is described. Section V describes the method of using the FOS algorithm to select features with predictive power. Section VI concludes this paper.

II. *TRAFFIC* CLASSIFICATION BACKGROUND

Due to the ever-increasing volume of encrypted network traffic it is impossible to ignore encrypted network traffic classification within network traffic analysis. There has also been an increase in encrypted local area network (intranet) traffic as most organizations have migrated to network administration tools that use encryption for security. Wright et al. [9] assessed this growing use of encrypted protocols and noted that while its use has greatly enhanced network security it has equally hindered traffic analysis.

Encrypted traffic naturally lends itself to statistical analysis, since access to the packet contents is restricted and only the statistical features of the packet are available [9]. A number of techniques of classification of encrypted traffic have been published. [10], [11], [12], [3], [13].

Alshammari et al. [14] used three different machine-learning algorithms to classify Skype [15] traffic flows from non-Skype traffic flows, then investigated the number of features each algorithm used to build its classification model. This work did not attempt to find a general algorithm for feature selection and only used 22 candidate features.

III. Feature Selection Background

A key requirement for statistical classification is the appropriate selection of a minimal set of features with predictive value. As it is not intuitively obvious which features have predictive value, it is desirable to have the feature selector search as large a feature space as possible. This paper proposes a systematic approach to feature selection, using the FOS algorithm for the classification of encrypted network traffic.

A. The Fast Orthogonal Search

The fast orthogonal search (FOS) algorithm [16] is a greedy algorithm that builds a parsimonious functional expansion of a time series using a subset of an arbitrary set of candidate functions that most significantly reduce the mean-squared error (MSE) of the functional expansion. The functional expansion of a signal $y[n]$ in terms of the arbitrary candidate function $p_m[n]$ is given by:

$$y[n] = \sum_{m=0}^{M} a_m p_m[n] + e[n] \quad (1)$$

where a_m are the model weights, $e[n]$ is the residual error and $M+1$ is the number of terms fitted out of the set of P available candidates.

The FOS algorithm performs an implicit orthogonalization of the candidate functions and creates an internal functional expansion using this orthogonal basis given by:

$$y[n] = \sum_{m=0}^{M} g_m w_m[n] + e[n] \quad (2)$$

where $w_m[n]$ are the orthogonal functions and g_m their respective weights.

FOS creates the model term-by-term by testing each candidate as the next model term and selecting the candidate with the maximum MSE reduction as the next model term. FOS continues adding terms to the functional expansion until:

- An arbitrary maximum number of terms have been fitted
- The MSE reduction of the term to be added is less than the MSE reduction of adding white noise
- The MSE reduction of the term to be added is less than a percentage of the energy of the signal
- An arbitrary percentage of the energy of the signal has been fitted.

Additional details of the FOS algorithm can be found in [16].

FOS has been successfully used in feature selection for classification in medical applications. [17], [18]. When using FOS as a feature selector, a training set of network flows with known classes are required. The target function $y[n]=1$ for in-class flows and $y[n]=-1$ for out-of-class flows and the candidate functions $p_m[n]$ are the candidate features of each flow. The FOS algorithm will choose a subset of the features that reduces the MSE between the functional expansion in (1) and the target function $y[n]$.

B. Best First Search

The Best First search algorithm (BeFS) [19] [20] is a graph-based search algorithm. Each node of the search represents a subset of the candidate features. Initially, each feature is assigned its own node. The children of these nodes are subsets with two features: the parent node feature and in turn each of the other features. The third level children have subsets of three features. Similar to the FOS algorithm, the BeFS is a greedy search algorithm, and at each step it will choose the best of all explored nodes within the space. The BeFS algorithm gives an efficient algorithm to search the huge number of feature subsets without performing an exhaustive search.

The BeFS algorithm computes the pairwise correlation between all the features in the graph and the correlation between the class vector and the features in the graph. The correlation between two feature vectors, f_1 and f_2, is given by

$$r_{f_1 f_2} = \frac{1}{N} \sum_{i=0}^{N-1} \left(\frac{\left(f_{1_i} - \overline{f_1}\right)\left(f_{2_i} - \overline{f_2}\right)}{\sigma_{f_1} \sigma_{f_2}} \right) \quad (3)$$

where $\overline{f_1}$ and $\overline{f_2}$ are the average values of the features, and σ_{f_1} and σ_{f_2} are the standard deviations of the features and N is the number of flows in the vector. The correlation between a feature and the class vector r_{fc} can be calculated using the class vector for f_2 in *(3)*

Given a subset S of k features, the merit of the given subset can be calculated in terms of its correlation measures as follows [21]:

$$M_{Sk} = \frac{\overline{r}_{fc}}{\sqrt{k + k(k-1)\overline{r}_{ff}}} \quad (4)$$

where, k is the number of features, $\overline{r}_{fc}$ is the average value of all feature-classifier correlations, and $\overline{r}_{ff}$ is the average of all feature-feature correlations.

Note that the merit M_{Sk} increases when the features are highly correlated to the class vector. The merit also increases when the correlation between the features decreases. It is generally accepted that features highly correlated to the class but not strongly correlated to each other result in high classification accuracy.

IV. K-NN CLASSIFIER

The k nearest neighbour (kNN) classifier is a method for classifying data based on a distance measure between the sample to be classified and the k nearest neighbors in an exemplar set [22]. There is an expectation that members of a given class will tend to form a natural cluster in the feature space while being separated from points of different classes. The kNN algorithm typically uses the Euclidean distance or the correlation coefficient as its distance measure.

The Euclidean distances between the data being classified and the training set exemplars are given by:

$$ED(j) = \sqrt{\sum_{i=1}^{M} \left[f_{test}(i) - f_j(i) \right]^2} \quad (5)$$

where f_{test} is the test feature vector, f_j is the j^{th} training feature vector, i is the feature vector index, and M is the total number of features in the vector.

In this work, the distance measure is the sum of the reciprocal of the k=3 smallest Euclidean distances as given by:

$$C = \sum_{j=1}^{K} \frac{T(j)}{ED(j)} \quad (6)$$

where T(j) = +1 for distances to in-class training data, T(j) = -1 for distances to out-class training data and ED(j) are the Euclidean distances between the data being classified and the three closest exemplars. If $C \geq 0$ the data is predicted to be in class; and if $C < 0$ then the data is predicted to be out of class.

V. METHODOLOGY

This research developed a systematic feature selection algorithm using the FOS algorithm to select a subset of features with predictive values. This method consists of three phases: collection and preparation of data; feature selection; and validation of the predictive value of the selected feature subsets.

A. Collection and Preparation of Data

The first step of the feature selection process is the collection of two sets of network traffic data: a training set and a testing set. In this research the training data was captured using tcpdump [23] from a typical university network over a 24 hour period, and the test data was captured from the same network over a 24 hour period 3 days later. From these raw captures, encrypted traffic was extracted by filtering only traffic originating from or terminating on port 443.

For this research we used an open source tool NetMate (Network Measurement and Accounting System) [7] to generate network traffic flows from raw packet data. Flows are defined in NetMate using 5-tuple keys consisting of source IP address, destination IP address, transport protocol, source port, and destination port. This research examined bidirectional transmission control protocol (TCP) and user datagram protocol (UDP) flows where the forward direction was determined by the first observed packet of the flow. The termination of TCP flows was either a proper connection closure or a flow timeout, and the termination of UDP flows was defined by a flow timeout. NetMate also defines sub-flows as a means of delineating periods of inactivity of 1s or more within a flow.

Dropbox traffic was chosen as the type of encrypted traffic to test this methodology. During the collection of data for this experiment it was observed that each time a Dropbox session is initiated between the local user and a Dropbox server, one of the initiation packets contained the text string ‘dropbox’ in the unencrypted content. Note, dropbox flows no longer contain this text string so the true class of the flow must be determined from the data.

An open source program called network grep (ngrep) [24] was used to extract all packets that contained the string ‘dropbox’ and a list of remote IP addresses was compiled from these packets. The network flows associated with the remote addresses attributed to the Dropbox server comprised the in-class set. Network flows that were not associated with these remote addresses comprised the outclass set. This analysis was performed independently on both the training and testing data sets.

Over the 24-hour period that the training data was gathered there were approximately 106,000 inclass flows and 2,945,000 outclass flows. The outclass flows were randomly sorted, and the total number of flows was reduced to equal the number of in-class flows. By setting an equal number of in-class and outclass instances, any sample size bias was eliminated during feature selection and classification.

B. Computing Flow Features

Statistical features for each flow were generated using packet-processing modules provided by netAI [25]. The features measure properties of the flow such as packet lengths, packet volumes, duration and time between the arrival of packets. The primary feature set consists of thirty-eight features from netAI and 6 features computed separately which represent the traffic burstiness.

The abbreviations and descriptions of these 44 features are shown in Table I below. The network protocol (TCP, UDP, etc) is the first feature (Line 1) called *proto*. The number of packets and bytes in the forward and reverse directions are given by *total_fpackets, total_fvolume, total_bpackets* and *total_bvolume* respectively (Lines 2-5). The minimum, mean, maximum and standard deviation of the forward and backward packet lengths are the next features (Lines 6-13). There are eight features measuring the minimum, maximum, mean and standard deviation of the packet inter-arrival times (Lines 14-21). Nine features characterize the duration, active and idle times of the subflows (Lines 22-30). The mean number of packets and mean number of bytes in subflows are the next four features (Lines 31-34). The number of packets in each flow with a PUSH Flag enabled in the forward and backward directions are represented by *fpsh_cnt* and *bpsh_cnt* (Lines 35 and 36). The length of the forward and backwards header are respectively *total_fhlen* and *total_bhlen* (Lines 37 and 38). The last 6 features in represent calculated values for the burstiness of the traffic (Lines 39-44). These features were derived by dividing *total_fpackets, total_fvolume, total_bpackets, total_bvolume, mean_fpktl* and *mean_bpktl* respectively by Mean Active Time.

TABLE I. PRIMARY FEATURE SET

	Feature Abbreviation	Feature Description
1	proto	Protocol
2	total_fpackets	Number of Packets in forward direction
3	total_fvolume	Number of Bytes in forward direction
4	total_bpackets	Number of Packets in backward direction
5	total_bvolume	Number of Bytes in backward direction
6	min_fpktl	Min forward packet length
7	mean_fpktl	Mean forward packet length
8	max_fpktl	Max forward packet length
9	std_fpktl	STD of forward packet length
10	min_bpktl	Min backward packet length
11	mean_bpktl	Mean backward packet length
12	max_bpktl	Max backward packet length
13	std_bpktl	STD of backward packet length
14	min_fiat	Min forward inter-arrival time
15	mean_fiat	Mean forward inter-arrival time
16	max_fiat	Max forward inter-arrival time
17	std_fiat	STD of forward inter-arrival times
18	min_biat	Min backward inter-arrival time
19	mean_biat	Mean backward inter-arrival time
20	max_biat	Max backward inter-arrival time
21	std_biat	STD of backward inter-arrival times
22	duration	Duration of Flow
23	min_active	Min active time
24	mean_active	Mean active time
25	max_active	Max active time
26	std_active	STD of active time
27	min_idle	Min idle time
28	mean_idle	Mean idle time
29	max_idle	Max idle time
30	std_idle	STD idle time
31	sflow_fpackets	Mean number packets in a forward sub-flow
32	sflow_fbytes	Mean number bytes in a forward sub-flow
33	sflow_bpackets	Mean number packets in a backward sub-flow
34	sflow_bbytes	Mean number bytes in a backward sub-flow
35	fpsh_cnt	Push count in forward direction
36	bpsh_cnt	Push count in backward direction
37	total_fhlen	Total forward header length
38	total_bhlen	Total backward header length
39	total_fpacket_rate	Forward Packet Burstiness
40	total_fvolume_rate	Forward Volume Burstiness
41	total_bpackets_rate	Backward Packet Burstiness
42	total_bvolume_rate	Backward Volume Burstiness
43	mean_fpktl_rate	Mean Forward Packet Length Burstiness
44	mean_bpktl_rate	Mean Backward Packet Length Burstiness

The collection of network traffic, construction of flows, and classification of the training data is shown in Fig. 1. This method was applied to both the training and testing data, resulting in four distinct sets, consisting of in-class and out-class files for both the training and testing sets respectively.

VI. FEATURE SELECTION

A vector of the values of each feature (e,g, *total_fpackets*) for all N data flows is used as the candidate FOS function $p_m(n)$ in (1). The target vector $y(n)$ is set to +1 for in-class and -1 for out-class flows. The FOS algorithm is run and it selects a subset of M features that have predictive value from the set of C candidate features. The FOS algorithm stops selecting features when adding a feature reduces the MSE no more than adding a noise term, the MSE is below a threshold, an arbitrary number of features have been fitted or fitting the term does not reduce the MSE by an arbitrary percentage of the initial energy in $y(n)$.

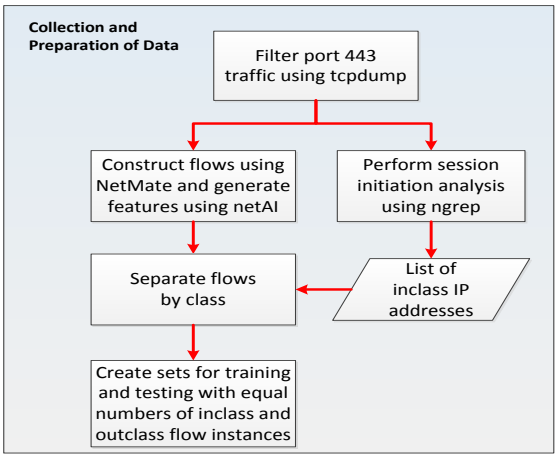

Fig. 1 The collection of network data and generation of the flow features.

Initially, FOS is used to select a subset of predictive features from the primary set of 44 features listed in TABLE I. Additional candidate features were derived from the primary feature set and FOS was then used to select features with predictive value from this larger feature set.

The derived features include the sum, difference and vector product of the primary features. The sum features are the sum of two features given by

$$p_i(n) = p_a(n) + p_b(n) \tag{7}$$

where $p_a(n)$ and $p_b(n)$ are two of the primary features, $p_i(n)$ is the derived sum feature and i is an unique integer identifier for the new candidate feature. Similarly, the difference features are given by

$$p_i(n) = p_a(n) - p_b(n) \tag{8}$$

where i is an unique integer identifier. For the sum and difference features the same feature cannot be used twice $(a \neq b)$. FOS can fit both positive and negative weights to the candidate terms, so only one difference between pairs a, b is required.

The second order vector product candidates are the point-by-point vector product of candidates as given by

$$p_i(n) = p_a(n) \times p_b(n). \tag{9}$$

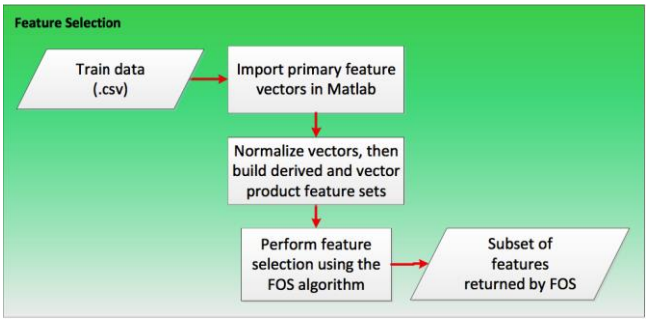

Fig. 2 Feature Selection Phase

Similarly, the third order vector product terms are given as the point-by-point product of three candidate function as

$$p_i(n) = p_a(n) \times p_b(n) \times p_c(n). \quad (10)$$

Note, for the vector product terms, the square of features ($a = b$) or cube of features ($a = b = c$) are allowed as features. For the sum, difference, and vector product features, only unique features generated by all combinations of a, b are added to the candidate feature set.

Building Feature Subsets Using the FOS Algorithm

For each feature in the candidate matrix, the FOS algorithm calculates the resulting MSE reduction for each feature assuming it were the next fitted candidate. The candidate that produces the greatest mean squared error reduction is chosen as the next feature for inclusion in the model. Features are fitted to the model until one of the stopping criteria is met. This subset of features selected by FOS is then used by the kNN classifier.

The FOS algorithm can be biased when features are not of the same order of magnitude [26]. To avoid these errors, each feature was normalized to have zero mean and unit energy. The normalized features and corresponding ground truth array were the inputs to the FOS algorithm.

Fig. 2 provides a graphical depiction of the feature selection phase, from building a candidate feature set, to building a subset using the features selected by the FOS algorithm. Note that the derived features are computed from the primary features and passed into FOS as candidate feature vectors. However, due to the large number of vector product features, the FOS algorithm computed the vector-product features as needed from the primary and derived features. The memory requirement for candidate features was significantly reduced by not precomputing all the vector product features. The mean and standard deviations used to normalize the training set candidate features can be saved and used to normalize the test set data in real-time. Alternatively, the set of test data can be normalized using the mean and standard deviation of the test set when batch processing network data.

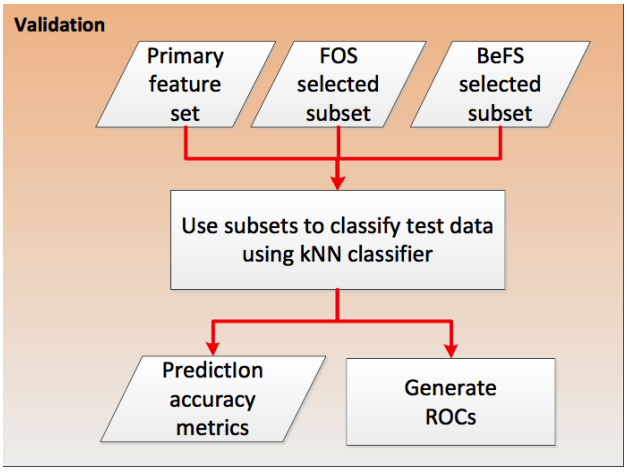

Fig. 3 Validation Phase

VII. Validation

In order to validate that the feature subsets chosen by the FOS algorithm have predictive value, their prediction accuracy was ascertained using the kNN classifier. Comparisons were also made between the prediction accuracy using feature subsets selected by the FOS algorithm against that resulting from subsets selected by the BeFS algorithm and the original primary feature set. Fig 3 shows a flow chart of the validation phase.

A. Receiver Operator Characteristics Curves

The primary validation technique used was a comparison of the area under the curve (AUC) of receiver operator characteristics (ROC) curves. A ROC curve is a graphical plot of the true positive rate (TP_{rate}) verses false positive rate (FP_{rate}). The true positive rate is calculated by:

$$TP_{rate} = \frac{TP}{TP + FN}. \quad (11)$$

and the FP rate is calculated using:

$$FP_{rate} = \frac{FP}{FP + TN}. \quad (12)$$

The ROC curves are then generated by varying the threshold used in the predictor's threshold detector and plotting the TP_{rate} versus the FP_{rate}. For the kNN classifier, the input to the threshold detector is the sum of the k reciprocal Euclidean distances for each instance.

B. Detection Rate

In addition to the AUC of the ROC, the detection rate, or the rate of correct predictions, was also used as a comparison metric. The detection rate is defined as the probability of correctly detecting, or classifying, flows and is given by:

$$DR = 1 - P_e. \quad (13)$$

where

$$P_e = P(inclass)P(e \mid inclass) + P(outclass)P(e \mid outclass).$$

If there are an equal number of inclass and outclass instances this equation can be simplified to:

$$P_{error} = \frac{1}{2} FN_{rate} + \frac{1}{2} FP_{rate}. \quad (15)$$

C. Phi Coefficient of Association

The phi coefficient of association, also known as the Matthews' correlation coefficient, was also used as a comparison metric and is given by [27]:

$$\phi = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \quad (16)$$

where ϕ is the phi coefficient of correlation, TP is true positive, TN is true negative, FP is false positive and FN is false negative. The phi coefficient is a value between −1 and +1, where a coefficient of +1 indicates a perfect prediction, −1 indicates a complete disagreement between prediction and observation, and 0 indicates a random prediction.

VIII. RESULTS

The feature selection and classifier training was conducted on 50,000 flow instances of the training dataset, while validation of the selected subsets was conducted on 50,000 flow instances of the test dataset. The ROC curves, AUC, ϕ coefficient, detection rate and total number of errors were computed for each classifier using the previously unseen testing dataset. Each dataset contained 25,000 inclass and 25,000 outclass flow instances. Feature selection was performed using four candidate feature sets, which are described in Table II

The first dataset consist of only the 44 primary features listed in Table I. The second candidate feature set consists of the primary features and all the unique pairwise sum and difference features given by (9) and (10). The third set includes the all the candidates from the second feature set and all the second order vector product terms given in (11). The final set includes all the features in the third set as well as third order vector product features given by (12).

Overfitting occurs when the accuracy of prediction actually decreases when additional model terms are added during the training of the classifier. The predictor models the random noise of the training data but has larger prediction errors when presented with previously unseen data than without the additional model term. During feature selection it was found that the AUC values would rise to a peak in the range of 8 to 12 features, and then slowly decline as additional features were added. The stopping thresholds of the FOS and the BeFS algorithm were chosen to prevent the overfitting of data.

The FOS algorithm was run on each of these four data sets to select a subset of predictive features. The stopping conditions for the FOS algorithm were set such that:

TABLE II. SUMMARY OF FEATURE SETS

Set Name	Number	Description
Primary	44	38 netAI features 6 rate features
Derived	1893	44 primary features 1849 sum and difference features
2nd Order Vector-products	2839	1893 derived features 946 2nd order vector-product features
3rd Order Vector-products	16,083	2839 2nd order vector-product features 13,244 3rd order vector-product features

- a maximum of 44 features could be selected;
- each term fitted at least 1% of the energy in the target;
- fitting was stopped if a term fitted no more energy than would fitting a WGN term; and
- fitting was stopped when 75% of the energy in $y(n)$ was fitted.

A. FOS feature selection with a kNN predictor

A summary of the kNN predictor`s performance using feature subsets chosen by the FOS and BeFS algorithms are shown in Table III. The best performance for a kNN predictor was obtained using the 12 features selected from 2nd order vector-product feature set as shown in the highlighted row in Table III. Compared to a kNN using all 44 features of the primary feature set, the kNN using the FOS selected subset resulted in 106 fewer errors using 32 fewer features and took 81% less time to classify. Note: these experiments were run on a 2.2 GHz Intel Core i7 microprocessor and the time is used to compare the relative speeds of the algorithms.

The first three rows of Table III contain the result when only the primary feature set was used to select features with predictive value. The first row shows the results when all 44 primary features were used by the kNN predictor without any feature selection. The FOS and BeFS algorithms both selected 10 features when run on the primary feature set. The kNN predictors using the FOS and BeFS selected feature sets run 6 times faster than the kNN predictor which uses all 44 primary features.

The second and third rows of Table III show that with only 10 features selected, FOS has a higher AUC, DR and ϕ coefficient than BeFS. In addition, the FOS algorithm runs in 1/3 the time of the BeFS algorithm.

Table III also includes the kNN prediction results for feature subsets selected from the derived, 2nd order and 3rd order vector-product feature sets. Note, the FOS algorithm created the cross-order features as required and didn't require the cross-order features to be precomputed and stored. The BeFS algorithm was not run on the 2nd order and 3rd order vector-product feature sets as we did not have enough memory to precompute and store all the vector-product features for the BeFS algorithm. For the kNN predictor using FOS selected features from the derived candidate set, the AUC is slightly lower but the DR and ϕ are higher and total errors are lower than for the FOS selected feature selected from the primary features. For the derived feature sets, the FOS algorithm is about 25× faster than the BeFS algorithm. Note the prediction results were better for the 2nd order vector-product feature set than the and 3rd order vector-product feature set even though

TABLE III SUMMARY OF RESULTS

	Feature Selection			kNN Classification Results				
Feature Set	**Selection Method**	**Subset Size**	**Time to Select**	**AUC**	**DR**	**Phi**	**Total Errors**	**Time (min)**
Primary (44)	None	44	-	0.9893	0.9646	0.9292	1772	18
	FOS	10	17s	0.9783	0.9457	0.8914	2716	3
	BeFS	10	46s	0.9534	0.9143	0.8288	4285	
Derived (1893)	FOS	12	57s	0.9773	0.9470	0.8941	2650	3.5
	BeFS	12	~25min	0.9433	0.9132	0.8266	4340	
2nd Order Vector-product(2839)	FOS	12	118s	0.9898	0.9667	0.9334	1666	
3rd Order Vector-product (16,083)	FOS	10	~4min	0.9861	0.9568	0.9137	2160	3

the FOS model had a lower MSE for the 3rd order vector-product feature set.

From Table III, we conclude that the FOS algorithm can efficiently (in time and memory usage) select a small number of features with strong predictive results from a large candidate set. The candidate set of features can include features that are the sum, difference and vector-product of the primary feature set. In addition, the kNN algorithm for 10 or 12 features is at least 5× faster than the kNN for the 44 primary features. Fig. 4 shows the ROC curves for the kNN predictor using primary feature set, FOS selected features, and BeFS selected features from the derived candidate sets. The operating point of the kNN predictor is marked on each curve. It can be clearly seen that the FOS algorithm selects a subset of features with better predictive value than the BeFS algorithm. The DR and ϕ coefficients for the predictor with the FOS selected features are lower than when using all 44 primary features.

Features are often chosen subjectively by an expert and may be highly correlated to each other. It may be impossible for an expert to select a subset of predictive features. Table IV lists the features selected by the FOS and BeFS algorithms for the primary, derived and 2nd order vector product feature sets respectively.

Highlighted in Table IV are the two features that both FOS and BeFS selected from the primary candidate set. The features selected from the derived feature set are also shown in Table IV and it can be seen that all the features fitted by FOS and all but one fitted by the BeFS algorithm (*max_bpktl*) are derived features.

In Table V, the MSE in the training data and the AUC, DR and ϕ for a kNN predictor using the only the first N terms selected is shown. For the derived feature set, a derived feature was selected by FOS first because it fitted more energy in the training set than any of the primary features. Note that the MSE in the training set reduced as each term was selected by FOS. In addition, the AUC, DR and ϕ increased as each feature was selected. Thus, the derived feature and vector product features have more predictive power than the primary feature set as can be seen in Table V.

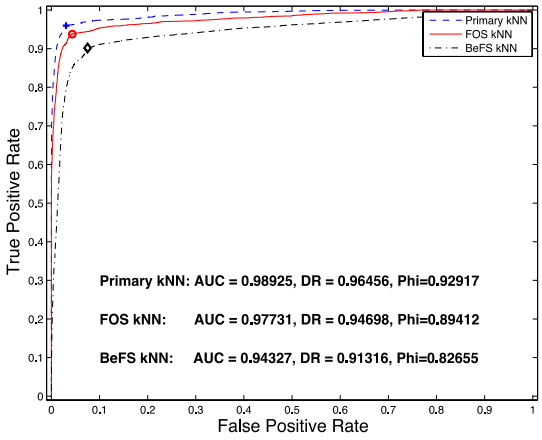

Fig. 4 The ROC curves for the kNN predictor for features selected by FOS for the Derived Feature Set.

It is also worth noting that the BeFS and FOS algorithms did not select the same features and from Table III it can be seen that the kNN predictor with FOS selected features had higher AUC, DR and ϕ coefficient values than the kNN with the BeFS selected features.

IX. Conclusion

This research presented a general-purpose technique for building reduced feature sets with high prediction accuracy for classification of encrypted traffic. This technique was shown to minimize the error of the classifier using a subset of features selected from a primary feature set. Feature selection was separated from the data classification process through the use of independent training and test datasets.

It is acknowledged that the method was tested only on Dropbox traffic [8]; however, we postulate that equivalent successes would be achieved on other types of encrypted network traffic. Also, if additional features are added to Netmate, they and their derived and cross-product features can also be searched by FOS. Other potential features may include those based on entropy or encryption schemes. As this work

TABLE IV. Features Selected by FOS and BeFS for the Primary, Derived and Vector Product Feature Sets

	Primary Features		Derived Features		Derived and Vector Product Features
	FOS	**BeFS**	**FOS**	**BeFS**	**FOS**
1	std_fpktl	total_bvolume	mean_fpktl -max_fpktl	*max_bpktl*	mean_fpktl - max_fpktl
2	mean_fpktl	max_fpktl	std_fiat -mean_biat	proto - max_bpktl	std_fiat - mean_biat
3	min_active	max_bpktl	duration -std_active	total_bvolume - total_bhlen	max_bpktl × mean_bpktl
4	duration	min_fiat	mean_fiat -std_biat	min_fpktl - min_bpktl	max_idle × std_active
5	std_fit	max_biat	duration -max_idle	min_bpktl - min_active	mean_bpktl - max_bpktl
6	mean_biat	min_active	proto -min_fpktl	min_active - bpsh_cnt	min_active - min_idle
7	std_active	std_active	mean_fpktl -std_fpktl	proto + max_bpktl	max_idle × min_idle
8	std_biat	min_idle	max_fpktl -std_bpktl	proto + bpsh_cnt	min_bpktl × min_fpktl
9	std_bpktl	max_idle	max_bpktl +min_fiat	mean_fpktl + max_idle	std_active × std_bpktl
10	mean_fiat	fpsh_cnt	max_fpktl -min_fiat	max_fpktl+total_bvolume_rate	mean_fpktl - std_fpktl
11	-	-	max_fiat -min_active	max_fpktl + mean_fiat	max_biat × max_bpktl
11	-	-	min_idle -mean_idle	max_fiat - mean_biat	mean_fiat - std_biat

primarily presented results for the kNN classifier, further work can be done measuring the performance of other classifiers (SVM, NN, decision trees) using the FOS selected feature sets.

REFERENCES

[1] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/Realtime Traffic Classification Using Semi-Supervised Learning," *Perform. Eval.*, vol. 64, pp. 1194–1213, 2007.

[2] R. Alshammari and N. Zincir-Heywood, "Automatically Generating Robust Signatures Using a Machine Learning Approach to Unveil Encrypted VoIP Traffic Without Using Port Numbers, IP Addresses and Payload Inspection," Ph.D. Thesis, Dalhousie, Halifax, NS, 2012.

[3] A. Dainotti, F. Gargiulo, L. I. Kuncheva, A. Pescapè, and C. Sansone, "Identification of Traffic Flows Hiding behind TCP Port 80," in *2010 IEEE International Conference on Communications (ICC)*, 2010, pp. 1–6.

[4] D. J. Arndt and A. N. Zincir-Heywood, "A Comparison of three machine learning techniques for encrypted network traffic analysis," in *Computational Intelligence for Security and Defense Applications (CISDA), 2011 IEEE Symposium on*, 2011, pp. 107–114.

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, pp. 18–28, Mar. 2009.

[6] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 6, pp. 1437–1447, Nov. 2003.

[7] C. Schmoll and S. Zander, *NetMate - Version 0.9.5*. Germany: Fraunhofer FOKUS, 2009.

[8] Dropbox Inc, "Dropbox." [Online]. Available: https://www.dropbox.com/. [Accessed: 14-Feb-2013].

[9] C. V. Wright, F. Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *J. Mach. Learn. Res.*, vol. 7, pp. 2745–2769, 2006.

[10] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Proceedings of the 8th international conference on Passive and active network measurement*, Berlin, Heidelberg, 2007, pp. 165–175.

[11] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," *Comput. Netw.*, vol. 53, no. 1, pp. 81–97, Jan. 2009.

[12] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Detecting HTTP Tunnels with Statistical Mechanisms," in *IEEE International Conference on Communications, 2007. ICC '07*, 2007, pp. 6162–6168.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *Proc. ACM SIGCOMM*, vol. 35, pp. 229--240, 2005.

[14] R. Alshammari and A. N. Zincir-Heywood, "Investigating Two Different Approaches for Encrypted Traffic Classification," in *Privacy, Security and Trust. PST '08. Sixth Annual Conference on*, 2008, pp. 156–166.

[15] Skype Communications SARL, "Skype," *About Skype - What is Skype*. [Online]. Available: http://www.skype.com/en/about/. [Accessed: 06-Apr-2013].

[16] M. J. Korenberg and L. D. Paarmann, "Orthogonal approaches to time-series analysis and system identification," *IEEE Signal Process. Mag.*, vol. 8, no. 3, pp. 29–43, Jul. 1991.

[17] E. A. Shirdel, M. J. Korenberg, and Y. Madarnas, "Neutropenia Prediction Based on First-Cycle Blood Counts Using a FOS-3NN Classifier," *Adv. Bioinforma.*, vol. 2011, 2012.

[18] M. Rakoczy, D. McGaughey, M. Korenberg, J. Levman, and A. Martel, "Feature Selection in Computer-Aided Breast Cancer Diagnosis via Dynamic Contrast-Enhanced Magnetic Resonance Images," *J. Digit. Imaging*, pp. 1–11, 2012.

[19] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of A*," *J ACM*, vol. 32, no. 3, pp. 505–536, Jul. 1985.

[20] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

[21] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning," University of Waikato, Hamilton, New Zealand, 1998.

[22] "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[23] V. Jacobson, C. Leres, and S. McCanne, *tcpdump/libcap Public Repository - Version 4.3.0*. http://www.tcpdump.org/: University of California, Berkeley, CA.

[24] R. McCarty, "A Look at \tt ngrep," *J-SYS-ADMIN*, vol. 10, no. 5, pp. 75–76, May 2001.

[25] S. Zander and N. Williams, "netAI - Network Traffic based Application Identification." [Online]. Available: http://caia.swin.edu.au/urp/dstc/netai/. [Accessed: 20-Feb-2013].

[26] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*. Dover Publications, Incorporated, 1994.

[27] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

TABLE V. Prediction metrics for a kNN predictor using features selected from the primary, derived and vector product feature sets

	Primary Feature Set				**Derived Feature Set**				**Vector Product Feature Set**			
	MSE	**AUC**	**DR**	**Phi**	**MSE**	**AUC**	**DR**	**Phi**	**MSE**	**AUC**	**DR**	**Phi**
1	0.8658	0.0595	0.5055	0.1464	0.8261	0.2344	0.5770	0.3725	0.8261	0.2344	0.6235	0.3725
2	0.7746	0.8896	0.8600	0.7426	0.6530	0.8633	0.8111	0.6323	0.6530	0.8633	0.8111	0.6323
3	0.6993	0.9223	0.8877	0.7781	0.6187	0.8699	0.8199	0.6480	0.6020	0.9254	0.8711	0.7468
4	0.6628	0.9376	0.9066	0.8145	0.5916	0.9113	0.8599	0.7248	0.5630	0.9658	0.9237	0.8483
5	0.6310	0.9383	0.9020	0.8060	0.5695	0.9105	0.8729	0.7500	0.5310	0.9721	0.9346	0.8700
6	0.5755	0.9591	0.9229	0.8466	0.5508	0.9209	0.8797	0.7633	0.5020	0.9776	0.9434	0.8872
7	0.5493	0.9638	0.9281	0.8567	0.5351	0.9596	0.9200	0.8414	0.4780	0.9778	0.9448	0.8900
8	0.5351	0.9702	0.9347	0.8697	0.5122	0.9659	0.9295	0.8597	0.4560	0.9788	0.9471	0.8946
9	0.5181	0.9772	0.9437	0.8875	0.5055	0.9657	0.9306	0.8620	0.4350	0.9858	0.9578	0.9156
10	0.5109	0.9783	0.9457	0.8914	0.4996	0.9658	0.9309	0.8626	0.4220	0.9898	0.9643	0.9287
11	-	-	-	-	0.4947	0.9742	0.9407	0.8817	0.4080	0.9893	0.9649	0.9298
12	-	-	-	-	0.4888	0.9773	0.9470	0.8941	0.3960	0.9898	0.9667	0.9334