A SYSTEMATIC APPROACH TO FEATURE SELECTION FOR ENCRYPTED NETWORK TRAFFIC CLASSIFICATION

Authors

Dr. Don McGaughey

Dr. Ron Smith

Dr. Scott Knight

Trevor Semeniuk

Presented by Trevor Semeniuk

Motivational Scenario

Introduction
Motivation

- Classification of network traffic is becoming more difficult due to the use of encryption, non-standard ports and proprietary protocols

- Many classification methods use statistical features to ascertain the type of data being transferred
  - the choice of what features to use is difficult due to the sheer number of possible combinations

Introduction

Objectives

- The main contribution of this research is:

the development of a general-purpose method of selecting feature subsets with high prediction accuracy for encrypted traffic classification

- This was accomplished by:
  - Generating an expanded set of features based on a primary set
  - Using the fast orthogonal search (FOS) algorithm to select a subset of features from the expanded set for classifying Dropbox traffic
  - Comparing the FOS selected features to those selected by a comparable systematic approach utilizing the Best First Search (BeFS) Algorithm
  - Establishing metrics for determining the prediction accuracy of 3 different classifiers

Introduction

Current Techniques

- Subjective selection
  - Features are chosen using intuition or best guesses using expert knowledge
  - For example, choosing inter-arrival times and packet-sizes to distinguish between large file transfer and banking transfer
- Exhaustive search (Wrapper)
  - a reduced subset consisting of a combination of features is selected, then this subset is evaluated using a specific classifier
  - process is repeated until a given feature set results in acceptable classification of the data
- Systematic Approach (Filter)
  - Instead of performing classification to evaluate the merit of the feature subset, a filter is used internal to the algorithm to evaluate the merit of a chosen subset
  - Examples of filters include mean squared error (FOS) and entropy (correlation based feature selection (CFS))

Introduction

Deficiencies of Current Techniques

- Manual or subjective selection:
  - Feature selection is tedious
  - Only features with a phenomenological basis are likely to be considered
- Exhaustive search
  - The number of combinations that can be tested is limited by computing time
  - For example, selecting 10 features from 40: $\binom{40}{10} = 847,\ 660,\ 528\ \ trials$
  - Biases are introduced due to the same data set being used for feature selection and classification
    - This reduces the robustness of the model when applied to never-before seen datasets
    - Overfitting occurs due to memorization of training data rather than learning to generalize from trends

Introduction
Overview of Developed Method

- The FOS algorithm was used to select statistical-based network flow features with high predictive value for classification of Dropbox transactions
- Why FOS?
  - shown previously to be effective at selecting reduced feature subsets for the application of classification of biological data
  - It is relatively fast and computationally efficient algorithm for seeking a minimum model
  - Allows for systematic feature selection
- Why Best First
  - The BeFS algorithm explores a feature space by expanding the most promising nodes, according to an evaluation function
    - The evaluation function used in this thesis was a correlation-based feature selector (CFS)
  - In this manner, BeFS is a comparable systematic approach to feature selection

Introduction
Overview of Developed Method

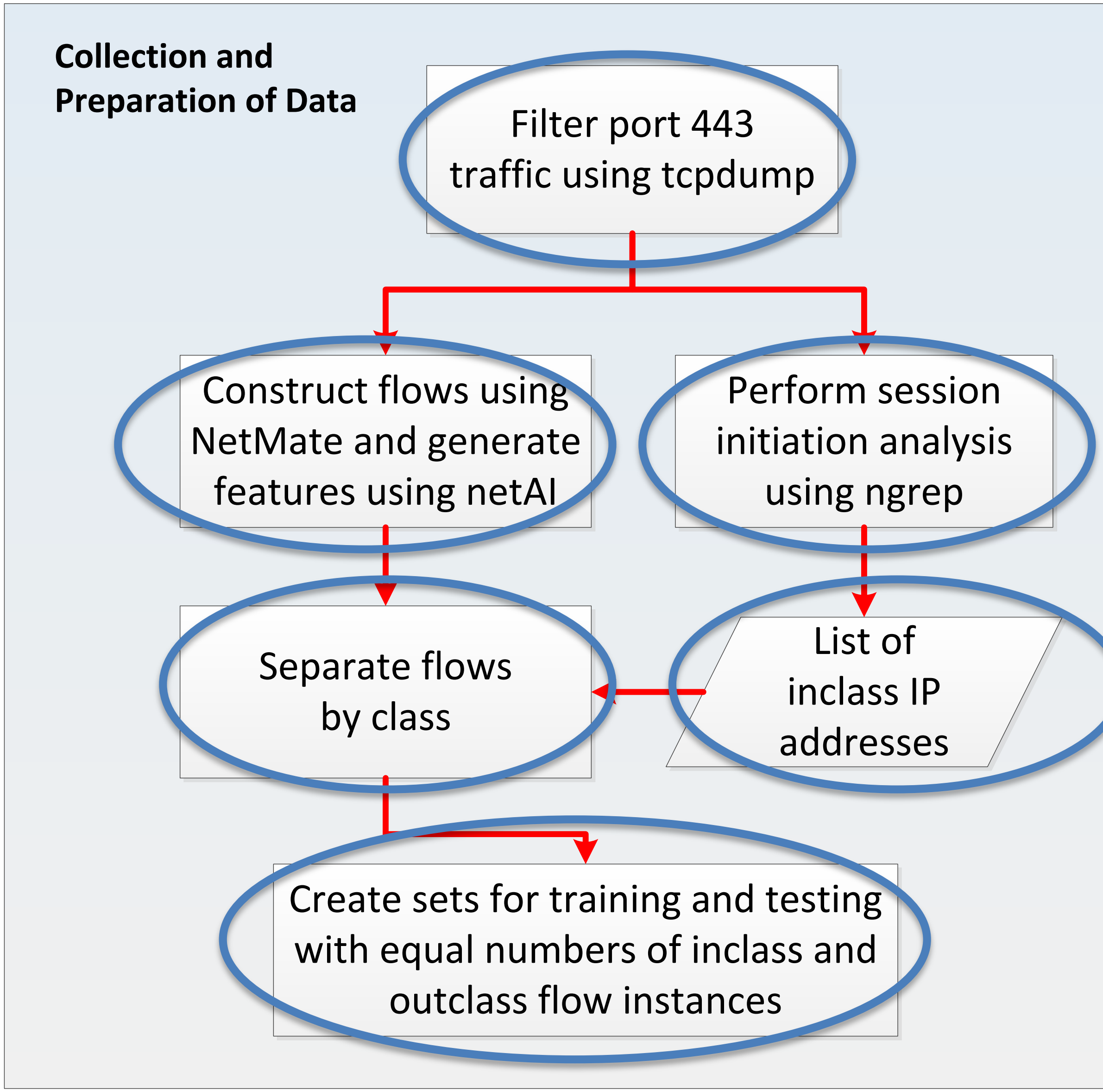
- Why statistical flow-based features?
  - Statistical features are very robust when applied to encrypted traffic with obfuscated IP addresses, port numbers and protocol types
  - User privacy is maintained
  - Flow-based features have been shown greater success for classification compared to packet-based features

- Why Dropbox?
  - Chosen to be a representative type of encrypted network traffic that would typically be present on a corporate network
  - This is a type of traffic that may or may not be permitted on a corporate network depending on IT policies

Methodology

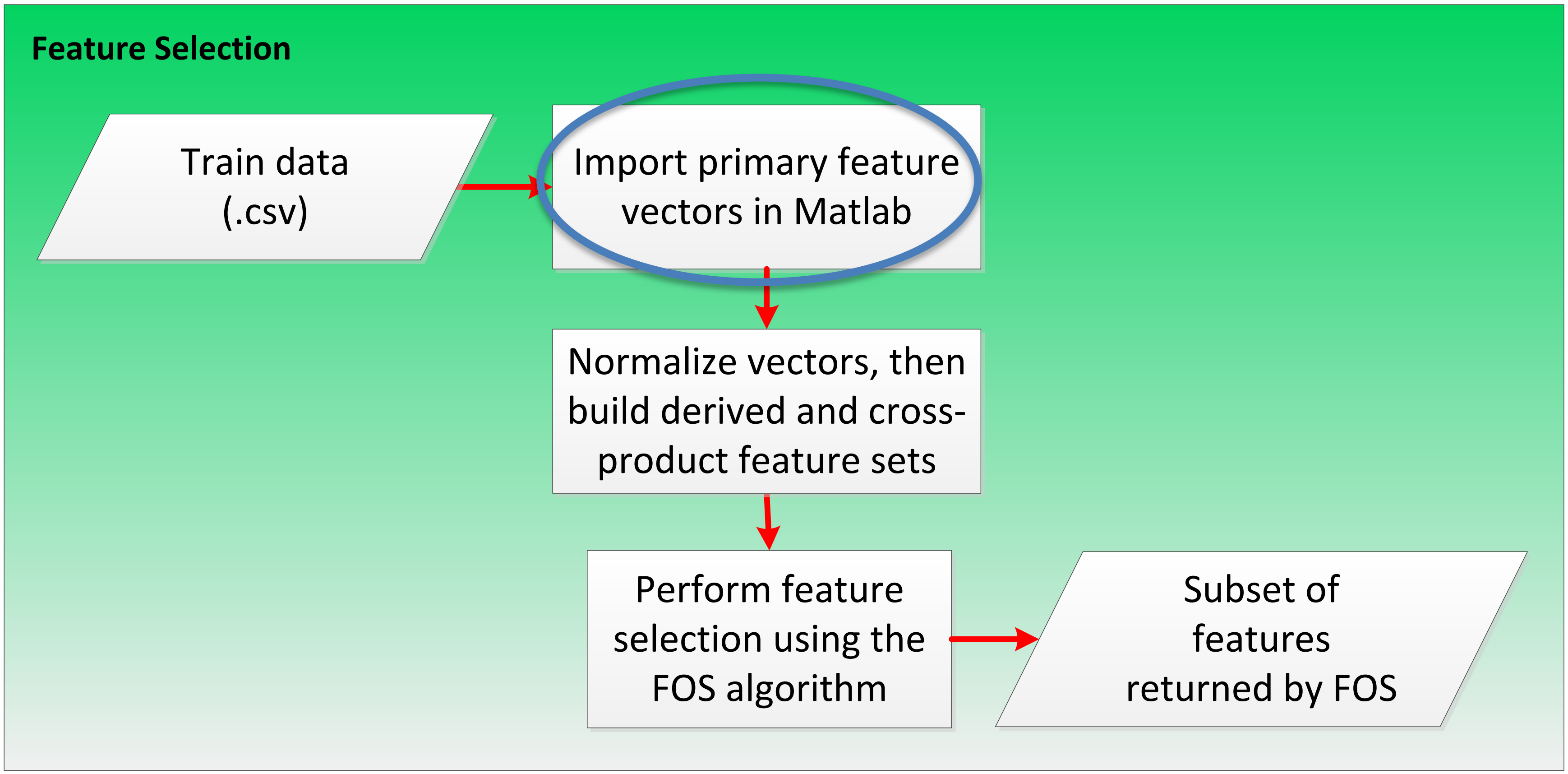
Methodology

Phase 1: Collection and Preparation of Data



Methodology

Phase 2: Feature Selection



Pre-FOS

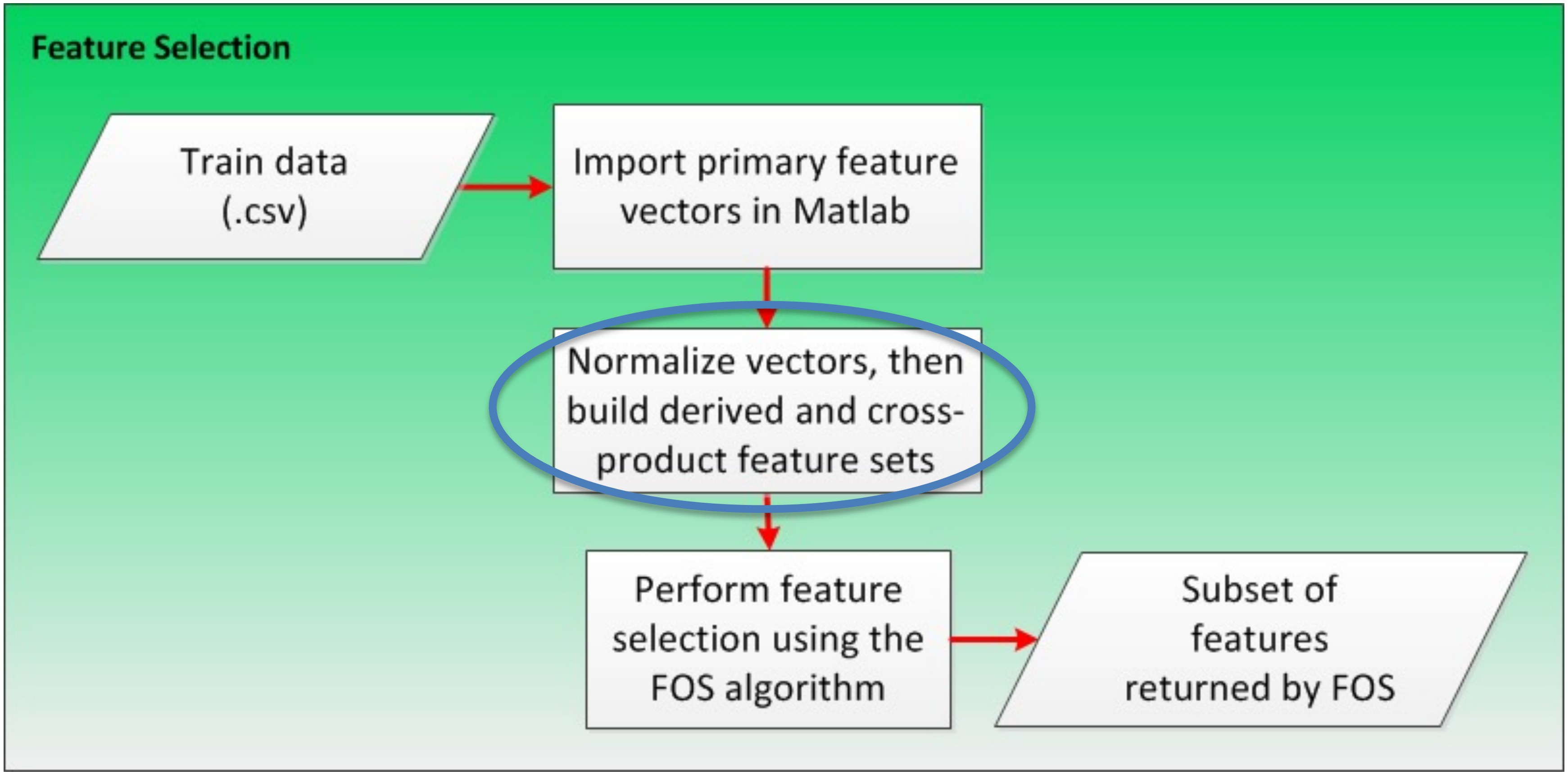
Step 1: Import Feature Vectors

		Features					Truth
		1	2	3	43	44	
	Flow Instance	mean_fpktl	duration	min_active	std_fpktl	std_fiat	
Inclass	1	1.31E+02	6.05E+07	7.07E+02	1.51E+02	9.17E+04	1
	2	8.30E+01	9.42E+07	7.30E+01	7.60E+01	1.52E+05	1
	3	1.42E+02	6.04E+07	9.14E+02	1.66E+02	4.11E+04	1
	4	1.42E+02	6.05E+07	9.77E+02	1.66E+02	4.16E+04	1
	5	1.63E+02	4.89E+05	4.89E+05	1.86E+02	4.39E+04	1
Outclass	49996	4.25E+02	4.53E+07	9.20E+04	5.40E+02	7.64E+04	-1
	49997	1.52E+02	6.40E+05	6.40E+05	1.43E+02	1.12E+05	-1
	49998	1.92E+02	1.09E+07	2.23E+04	3.34E+02	1.57E+05	-1
	49999	1.67E+02	4.08E+05	4.08E+05	2.03E+02	1.54E+04	-1
	50000	3.48E+02	4.78E+04	4.78E+04	5.35E+02	1.30E+04	-1
	mean	**2.26E+02**	**9.16E+08**	**2.17E+05**	**2.62E+02**	**1.24E+05**	**0**
	STD	**2.10E+02**	**3.84E+09**	**4.85E+05**	**1.58E+02**	**9.54E+04**	**1**

- Rows represent flow instances as generated by Netmate
- Columns represent feature vectors generated by NetAI
- Each cell represents the value of the specific feature for the specific flow instance

Methodology

Phase 2: Feature Selection



Pre-FOS

Step 2: Normalize Vectors

$$p_m(n) = \frac{p_m(n) - \overline{p_m(n)}}{\sqrt{\frac{1}{N-1}\sum_{n=0}^{N-1}\left(p_m(n) - \overline{p_m(n)}\right)^2}}$$

		Features						
		1	2	3		43	44	Truth
	Flow Instance	mean_fpktl	duration	min_active		std_fpktl	std_fiat	
Inclass	1	-0.4513	-0.2229	-0.4460		-0.7035	-0.3380	1
Inclass	2	-0.6794	-0.2141	-0.4474		-1.1786	0.2974	1
Inclass	3	-0.3990	-0.2229	-0.4456		-0.6085	-0.8691	1
Inclass	4	-0.3990	-0.2229	-0.4455		-0.6085	-0.8633	1
Inclass	5	-0.2992	-0.2385	0.5606		-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579		1.7605	-0.4985	-1
Outclass	49997	-0.3514	-0.2385	0.8697		-0.7542	-0.1276	-1
Outclass	49998	-0.1613	-0.2358	-0.4015		0.4556	0.3505	-1
Outclass	49999	-0.2802	-0.2385	0.3924		-0.3741	-1.1385	-1
Outclass	50000	0.5801	-0.2386	-0.3491		1.7288	-1.1628	-1
	mean	**0**	**0**	**0**		**0**	**0**	**0**
	STD	**1**	**1**	**1**		**1**	**1**	**1**

Why Normalize?

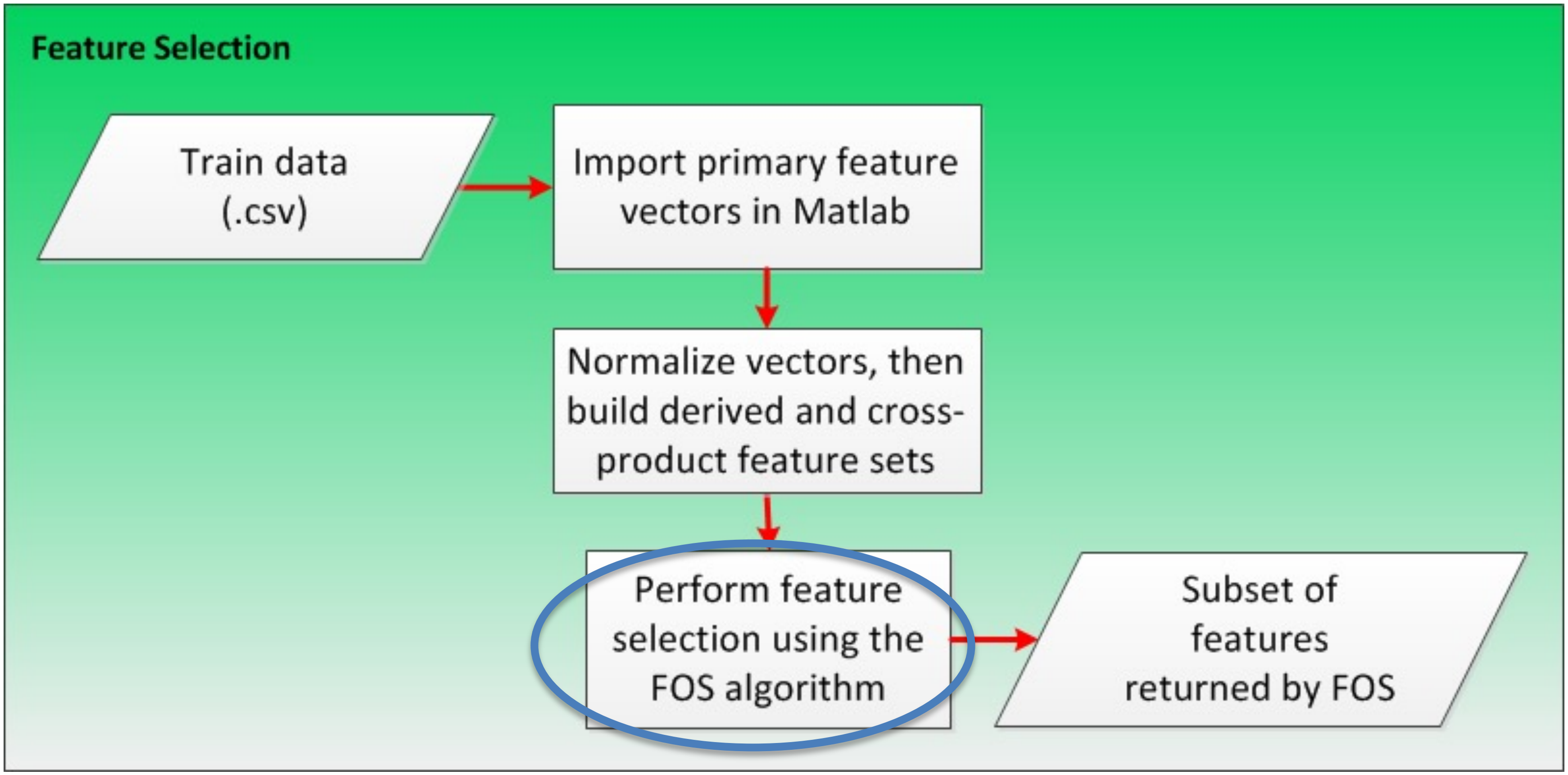
to ensure feature data is of the same order of magnitude to avoid round-off issues caused by numerical precision limitation inherent to computer systems

Methodology
Feature Sets

- The primary set of features:
  - Included 38 flow features generated by NetAI plus 6 rate features calculated using the mean active time
- These 44 features formed the primary set which was used to establish a benchmark for comparison
- Additional feature sets were created based on these primary features consisting of:
  - Derived
    - Comprising of sums and differences of primary features
  - Cross-Products
    - Comprising of 2^{nd} and 3^{rd} order cross-products of all primary feature combinations
    - Calculated as the point-wise vector product

Methodology

Phase 2: Feature Selection



The Fast Orthogonal Search Overview

- The model created by FOS is of the following form:

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

where,

y(n)	output (ground truth) of the system being modelled
M	number of features selected for inclusion in model
a_m	associated weights of the function expansion
p_m (n)	non-orthogonal candidate features selected
e*(n)*	residual model error

- The first step of the FOS algorithm is to build a functional expansion of the input using orthogonal functions of the following form:

$$y(n)=\sum_{m=0}^{M-1} g_m w_m(n)+e(n)$$

FOS Overview

Search Algorithm

		$p_m(n)$					y(n)
		1	2	3	43	44	
	Flow Instance	**mean_fpktl**	**duration**	**min_active**	**std_fpktl**	**std_fiat**	**Truth**
Inclass	1	-0.4513	-0.2229	-0.4460	-0.7035	-0.3380	1
	2	-0.6794	-0.2141	-0.4474	-1.1786	0.2974	1
	3	-0.3990	-0.2229	-0.4456	-0.6085	-0.8691	1
	4	-0.3990	-0.2229	-0.4455	-0.6085	-0.8633	1
	5	-0.2992	-0.2385	0.5606	-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579	1.7605	-0.4985	-1
	49997	-0.3514	-0.2385	0.8697	-0.7542	-0.1276	-1
	49998	-0.1613	-0.2358	-0.4015	0.4556	0.3505	-1
	49999	-0.2802	-0.2385	0.3924	-0.3741	-1.1385	-1
	50000	0.5801	-0.2386	-0.3491	1.7288	-1.1628	-1
	Qm	**1.27E-03**	**3.77E-02**	**7.48E-02**	**1.34E-01**	**4.35E-02**	
	MSE				**8.66E-01**		

- Algorithm searches all Q values and selects maximum
- Feature is chosen by maximum Q
- MSE is calculated using:

$$mse = \overline{\left(y(n) - \sum_{m=0}^{M} g_m w_m(n) \right)^2}$$

- This feature is now selected for inclusion in the model

FOS Overview

2nd Term Search

		$p_m(n)$						y(n)
		1	2	3		43	44	
	Flow Instance	mean_fpktl	duration	min_active		std_fpktl	std_fiat	Truth
Inclass	1	-0.4513	-0.2229	-0.4460		-0.7035	-0.3380	1
	2	-0.6794	-0.2141	-0.4474		-1.1786	0.2974	1
	3	-0.3990	-0.2229	-0.4456		-0.6085	-0.8691	1
	4	-0.3990	-0.2229	-0.4455		-0.6085	-0.8633	1
	5	-0.2992	-0.2385	0.5606		-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579		1.7605	-0.4985	-1
	49997	-0.3514	-0.2385	0.8697		-0.7542	-0.1276	-1
	49998	-0.1613	-0.2358	-0.4015		0.4556	0.3505	-1
	49999	-0.2802	-0.2385	0.3924		-0.3741	-1.1385	-1
	50000	0.5801	-0.2386	-0.3491		1.7288	-1.1628	-1
	Qm	**9.12E-02**	**3.29E-02**	**7.35E-02**		**0.00E+00**	**2.38E-02**	
	MSE	**7.75E-01**						

- Algorithm searches remaining Q values
- Feature is chosen by maximum Q
- *New* MSE is calculated
- This additional feature is now included in the model

FOS Overview

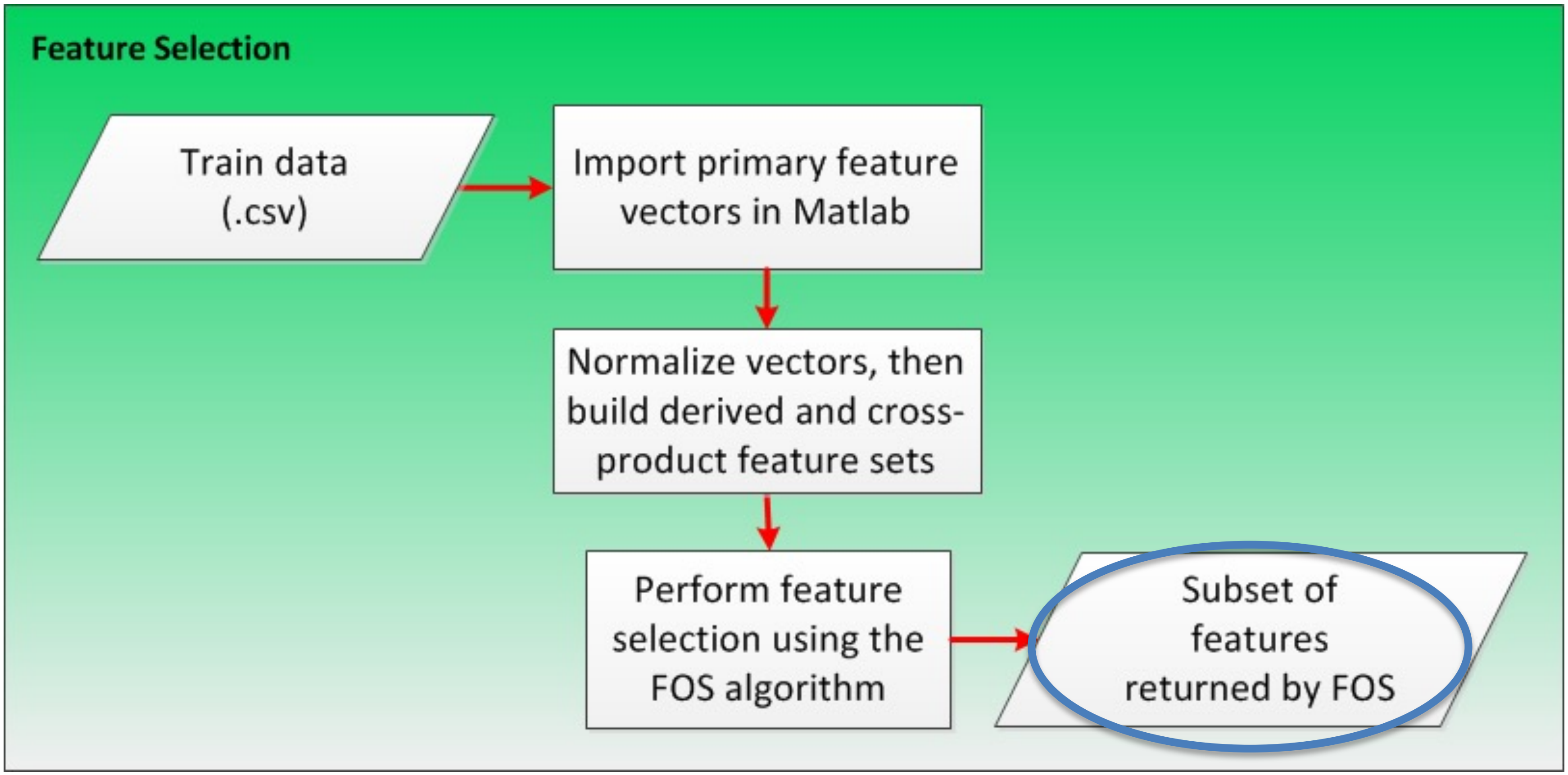
Stopping Criteria and Model Building

- Once any one of stopping criterion (TH_{MSE}, TH_{WGN}, TH_{Qm}, TH_{tta}) have been met, the FOS algorithm returns a model consisting of the subset of selected features

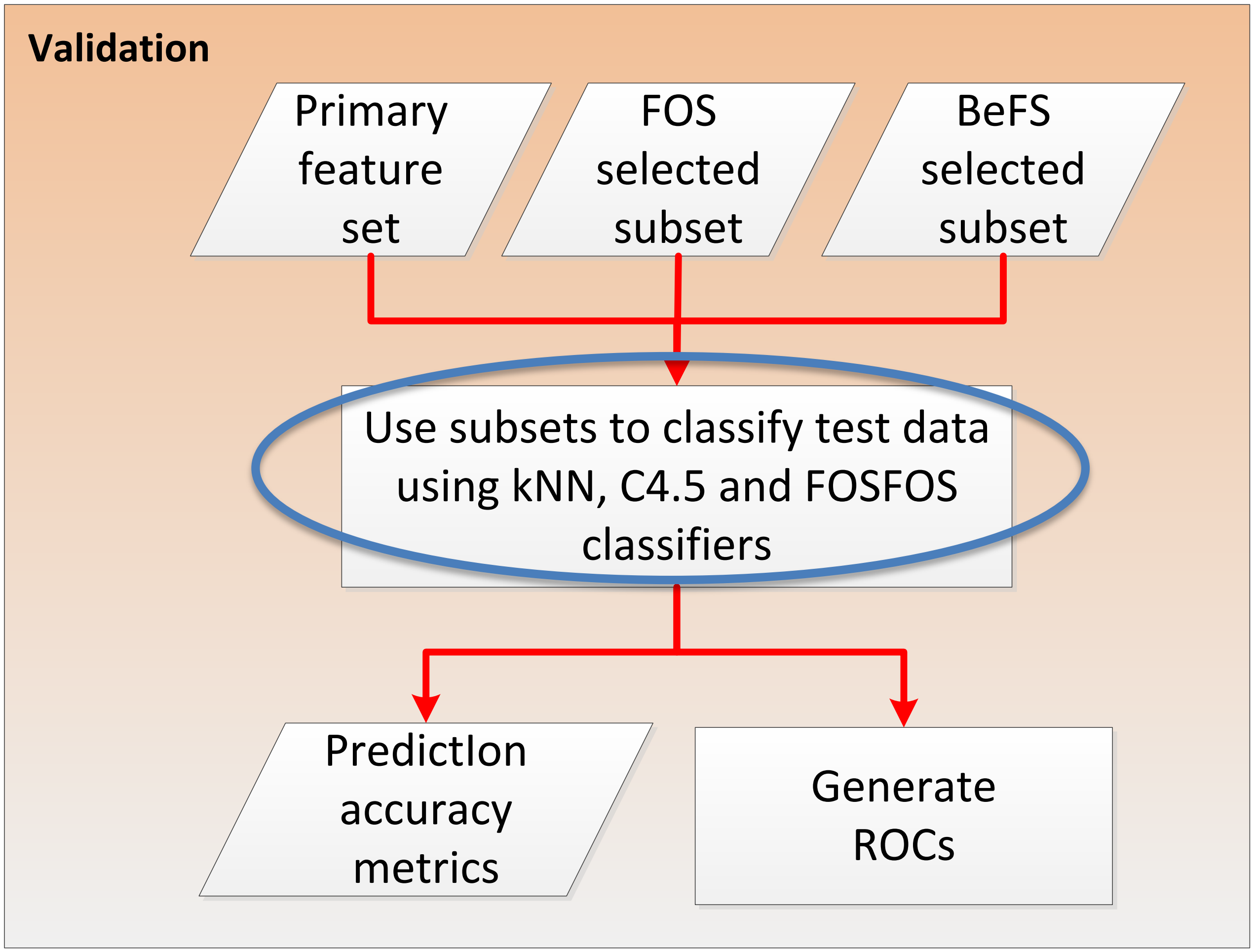
- A number of other parameters are also returned:
  - The non-orthogonal weights a_m of the functional series expansion
    - These weights are not used for feature selection, but are required to use FOS as a classifier
  - The mean and STD's for the training data
    - permits the test data to be normalized against the training data
  - The final MSE of the resulting model and the MSE reduction at each step (Q_m)

Methodology

Phase 2: Feature Selection



Methodology
Phase 3: Validation



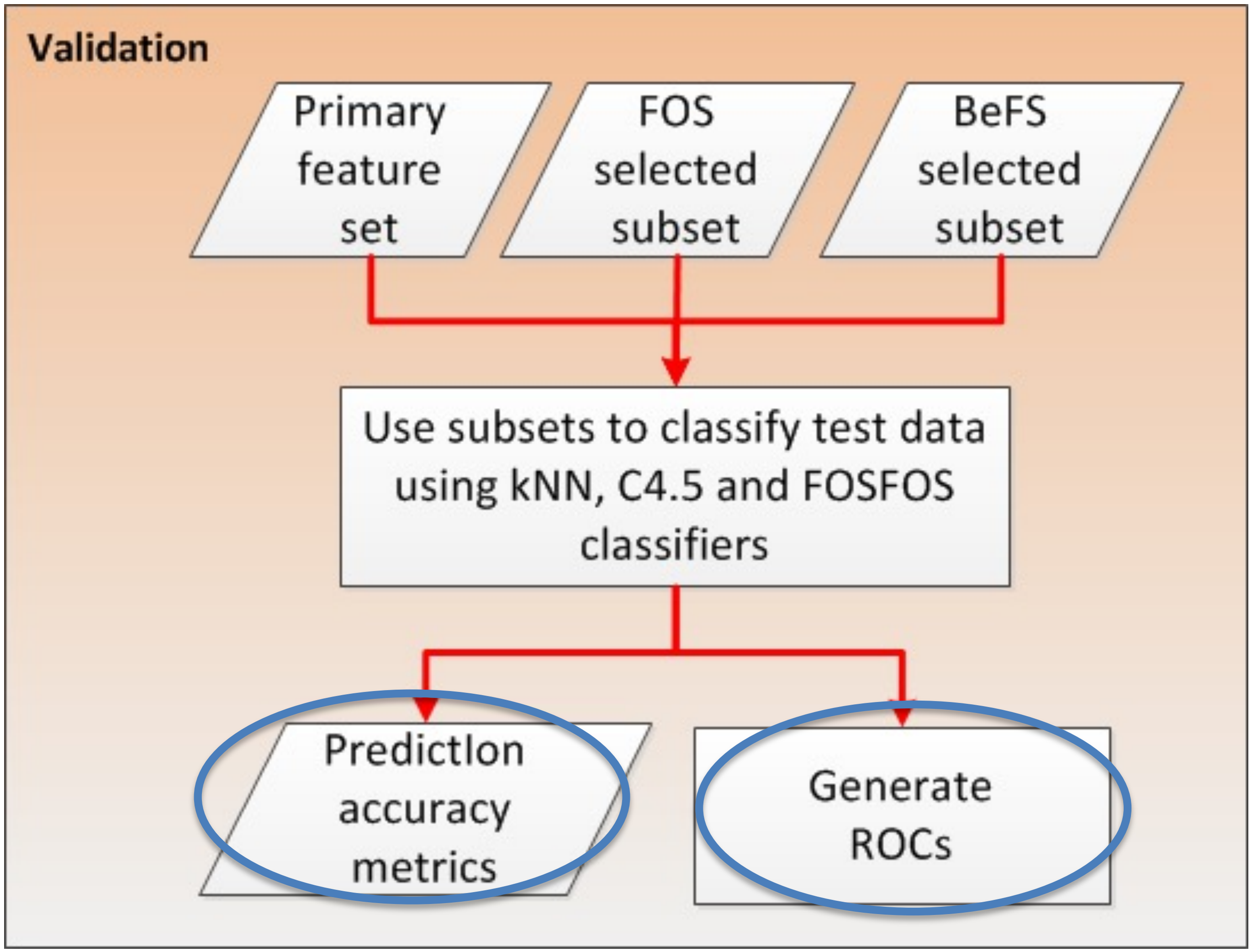
Methodology

Classifiers

- kNN
  - A general and proven method for classifying data based on their closest training examples in the feature space
- C4.5
  - A binary decision tree classifier, in which merit of splits are determined by an entropy evaluation
- FOSFOS classifier
  - The FOSFOS classifier uses the non-orthogonal weights a_m returned by the FOS algorithm to determine the class of each instance
- Thresholds
  - For the 3 classifiers used in this thesis a threshold of zero was used for class assignment

Methodology

Phase 3: Validation



Methodology

Validation

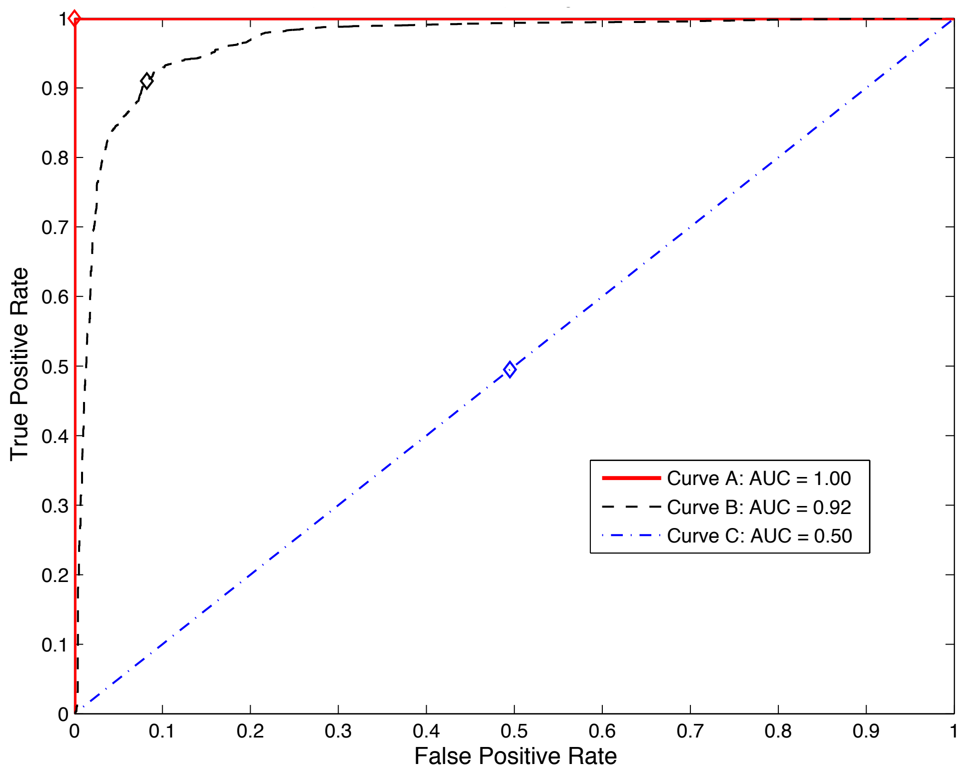
- Comparisons were made between the prediction accuracy using the Primary feature set against reduced subsets selected by:
  - the FOS algorithm
  - the BeFS algorithm

- Prediction accuracy of the developed method was evaluated using the following metrics
  - Phi Coefficients
    - a measure of association for two binary variables (value between −1 and +1)
  - Detection Rate
    - Total correct predictions over total test cases
  - Receiver Operating Characteristics (ROC) Curves

Methodology

ROC Curve Overview

- ROCs provide a graphical means of visualizing the ratio of FP_{rate} to TP_{rate} as the detection threshold varies



- The points indicated above show operating points of the classifiers for a given threshold (ie, FP and TP at threshold)
  - In this manner, the ROC curve can be used to choose the best operating point when evaluating different threshold values

Results

Results

Feature Sets

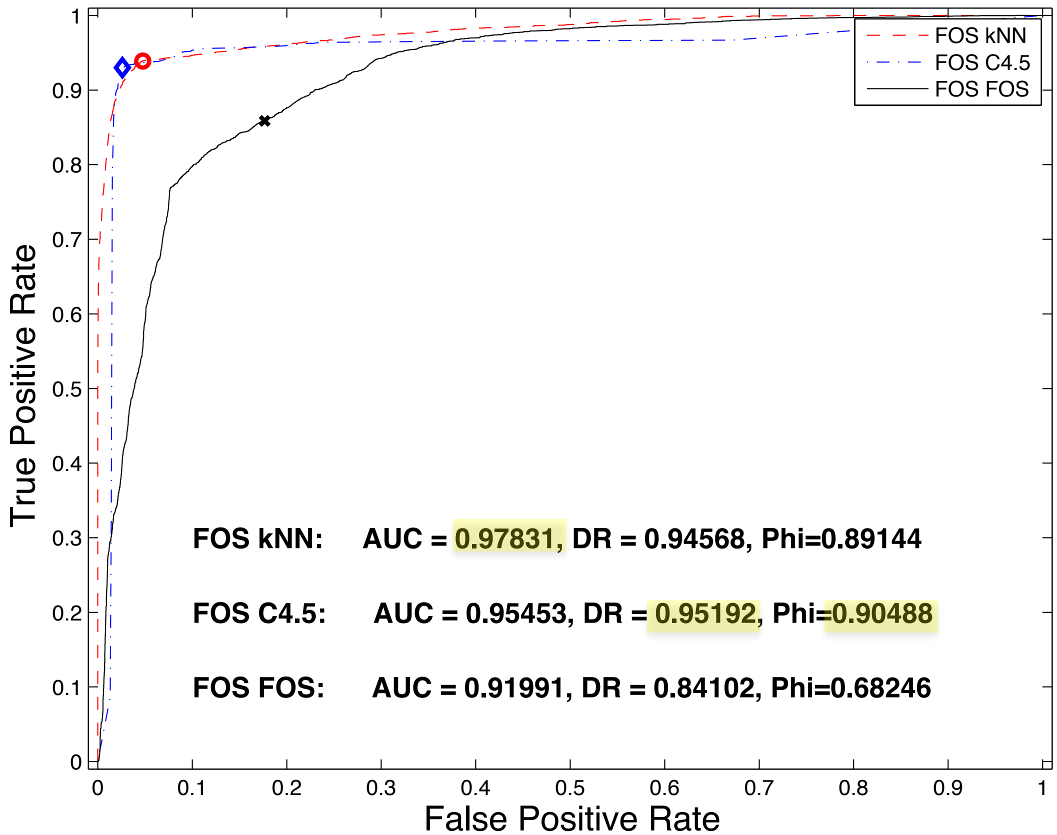
- Using the methodology developed, feature selection was performed using the FOS and Best First algorithms from the following 4 feature sets:

Set Name	Number of Features	Comprising
Primary Set	44	38 netAI features 6 rate features
Derived Set	1893	Primary Set (44) 1849 sum and difference features
2^{nd} Order Cross-Product Set	2839	Derived Set (1893) 946 2^{nd} order cross-product features
3^{rd} Order Cross-Product Set	16,083	2^{nd} Order Cross-Product Set (2839) 13,244 3^{rd} order cross-product features

- The primary feature set is used as the basis for all subsequent sets that are build upon the primary features
- The FOS and BeFS algorithms were used to build reduced subsets from the above feature sets
    - these reduced subsets were subsequently used to classify traffic

Results

Comparison of Classifiers



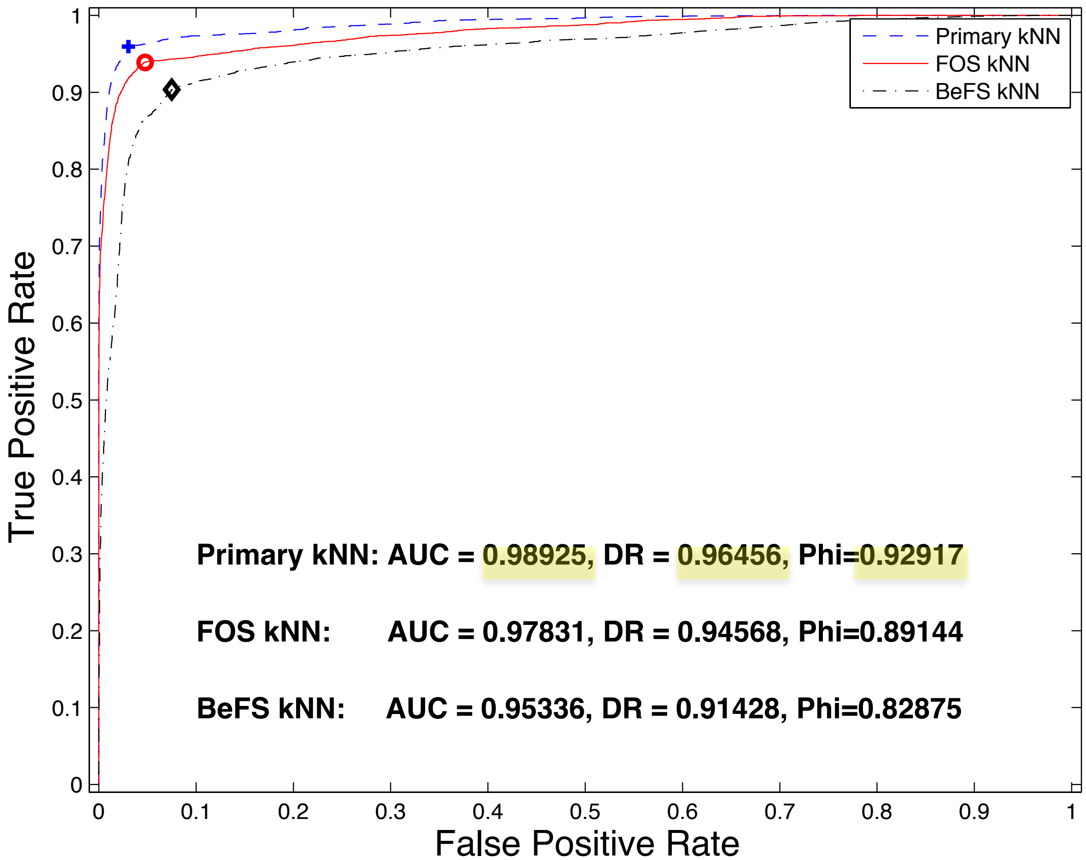
- 3 classifiers (kNN, C4.5, FOSFOS) were trialed with a 10-feature subset built by FOS using Primary Feature Set
- kNN was shown to have the best overall performance in terms of AUC
  - subsequently chosen for all further experiments

Results

Primary Features Only

Total Features	Comprising
44	38 netAI features 6 rate features

	Selected Features		
	FOS Selected	Resulting MSE	BeFS Selected
1	std_fpktl	0.8658	total_bvolume
2	mean_fpktl	0.7746	max_fpktl
3	min_active	0.6993	max_bpktl
4	duration	0.6628	min_fiat
5	std_fiat	0.6310	max_biat
6	mean_biat	0.5755	*min_active*
7	std_active	0.5493	*std_active*
8	std_biat	0.5351	min_idle
9	std_bpktl	0.5181	max_idle
10	mean_fiat	0.5109	fpsh_cnt



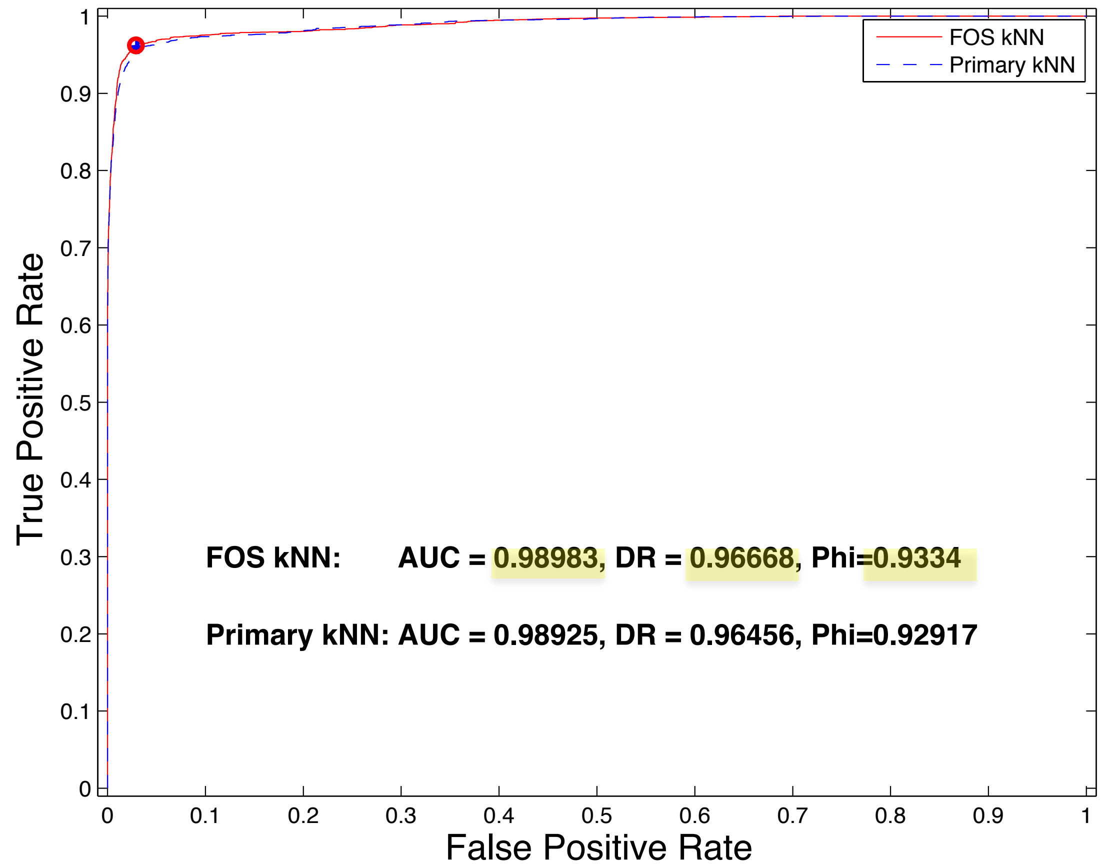
- FOS and BeFS were used to select 10 features from the 44-feature primary set
    - The primary feature set resulted in slightly increased kNN prediction accuracy compared to the 10-feature FOS subset

Results

2nd Order Cross-Product Features

Total Features	Comprising
2839	Derived Set (1893) 946 2nd order cross-product features

	FOS Selected Features	Resulting MSE
1	mean_fpktl - max_fpktl	0.8261
2	std_fiat - mean_biat	0.6529
3	max_bpktl x mean_bpktl	0.6020
4	max_idle x std_active	0.5630
5	mean_bpktl - max_bpktl	0.5312
6	min_active - min_idle	0.5021
7	max_idle x min_idle	0.4778
8	min_bpktl x min_fpktl	0.4563
9	std_active x std_bpktl	0.4355
10	mean_fpktl - std_fpktl	0.4219
11	max_biat x max_bpktl	0.4080
12	mean_fiat - std_biat	0.3960



- FOS was used to select 12 features from the 2839-feature cross-product set
  - Based on all three measures of prediction accuracy the FOS selected subset outperforms the benchmark primary feature set

Results

Summary

Feature Set	Feature Selection			kNN Classification Results				
	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 Cross-Product (2839)	FOS	12	118s	0.9898	0.9667	0.9334	1666	
3rd Order Cross-Product (16,083)	FOS	10	~4min	0.9861	0.9568	0.9137	2160	3

- Best performance for a systematically selected subset was from the 2nd order cross-products using the FOS algorithm
- Compared to the benchmark primary feature set:
  - The FOS selected subset resulted in 106 fewer errors
  - used 32 fewer features
  - Took 14.5 less minutes to classify using the kNN classifier

Conclusion

Conclusion
Future Work

- Analysis of Other Traffic Types
  - Classification of Dropbox traffic was performed, but it is desirable to try other transmission types, such as banking, VoIP, and SSH
- Expanded Feature Space
  - FOS is capable of operating in a feature rich environment
  - As shown by the cross-product results, it is desirable to created as large a features set as possible
  - This feature rich environment should consist of entropy, directionality of packets, encryption schemes, or network conversations

Conclusion
Future Work

- Variance of False Positive and False Negative Thresholds
    - By modifying the threshold criterion in the kNN and FOSFOS classifiers it is possible to vary the balance between FP and FN predictions
    - The ability to adjust the FP/FN balance would be particularly advantageous to an analyst who must direct focus to either positive or negative results.

Conclusion

Contributions and Significant Results

- First body of work to study the application of FOS to network flow feature selection for encrypted traffic classification
- Developed a general-purpose technique (non-data type specific) for building reduced feature sets with high prediction accuracy for classification of encrypted traffic
- Best performance was achieved using a 12-feature subset selected by the FOS algorithm from a 2nd order cross-product set

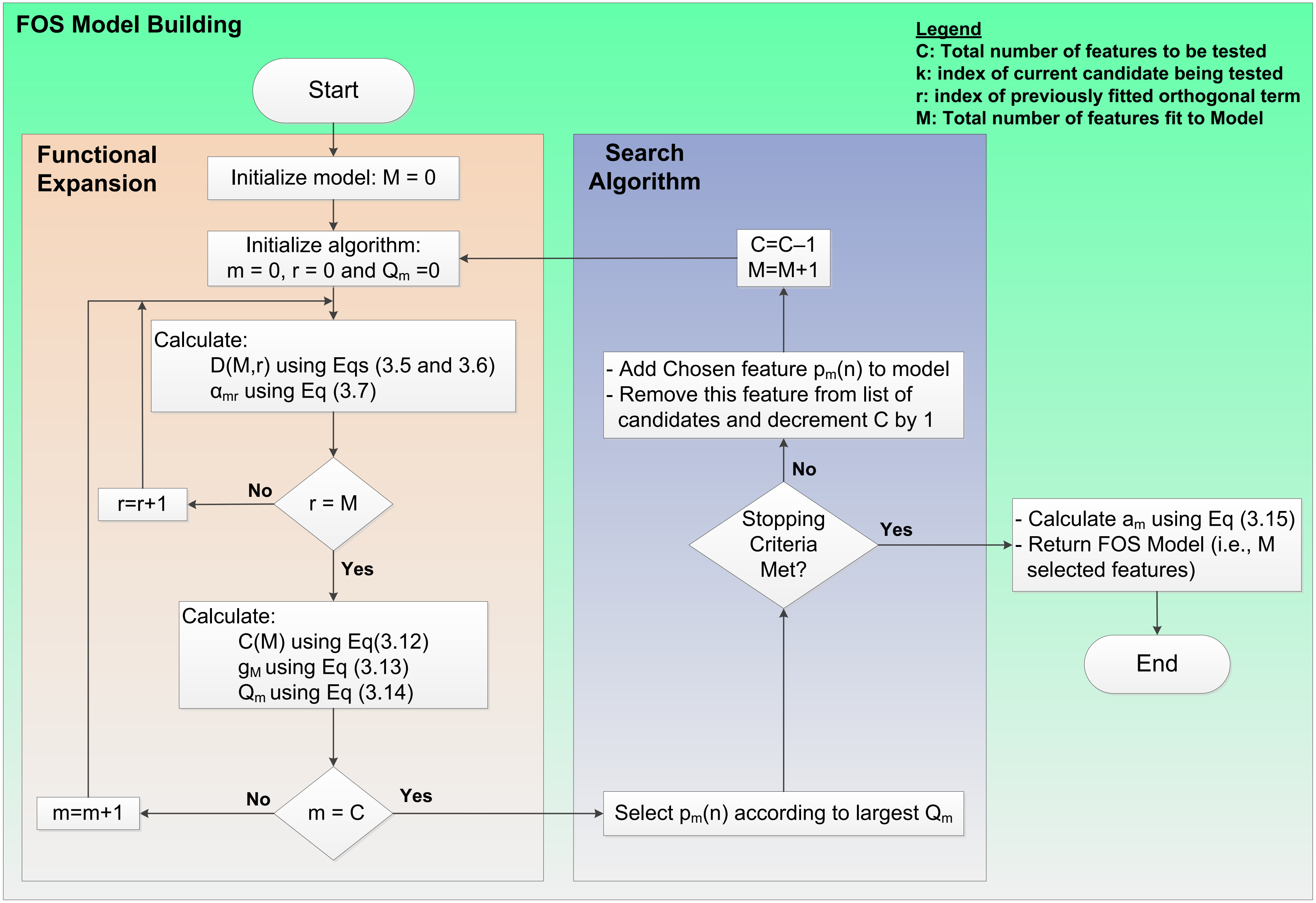
Questions

Extra Slides

The Fast Orthogonal Search Pseudo Code

- while (not stopping criteria)
  - for each remaining candidate function
    - find orthogonal weight, g_m
    - find mse reduction (Q)
  - end for
  - fit candidate with highest Q_{max}
- end while
- build model
  - coefficients, a_m, and selected features, $p_m(n)$

The Fast Orthogonal Search



The Fast Orthogonal Search Overview

- The model created by FOS is of the following form:

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

where,

y(n)	output (ground truth) of the system being modelled
M	number of features selected for inclusion in model
$\boldsymbol{p_m(n)}$	non-orthogonal candidate features selected
$\boldsymbol{a_m}$	associated weights of the function expansion
e*(n)*	residual model error

- The first step of the FOS algorithm is to build a functional expansion of the input using orthogonal functions of the following form:

$$y(n) = \sum_{m=0}^{M-1} g_m w_m(n) + e(n)$$

FOS Overview

Step 1: Gram-Schmidt Orthogonalization

- FOS uses the Gram-Schmidt Orthogonalization (GS) method to implicitly transform the set of arbitrary candidate functions $p_m(n)$ into orthogonal functions, $w_m(n)$
- The implicit orthogonal expansion is calculated recursively by defining the correlation of the m^{th} candidate function and the r^{th} orthogonal function to be:

$$D(m,r) = \overline{p_m(n) w_r(n)}$$

where,

$w_r(n)$ are the previously fitted orthogonal terms, and the overbar represents the time average

FOS Overview

Step 1: Gram-Schmidt Orthogonalization

- The orthogonal expansion can be found recursively, eliminating the requirement to calculate and store the orthogonal functions as an intermediate step. D(m,r) is calculated iteratively using:

$$D(m,r) = \overline{p_m(n)p_r(n)} - \sum_{i=0}^{r-1} \alpha_{ri} D(m,i)$$

- Where $p_r(n)$ are the previously fit functions and the GS weights are calculated using:

$$\alpha_{ri} = \frac{D(m,r)}{D(r,r)}$$

For m=1..M
and r = 1..m

FOS Overview

Step 1: Gram-Schmidt Orthogonalization

		$p_m(n)$					$y(n)$
		1	2	3	43	44	
	Flow Instance	mean_fpktl	duration	min_active	std_fpktl	std_fiat	Truth
Inclass	1	-0.4513	-0.2229	-0.4460	-0.7035	-0.3380	1
	2	-0.6794	-0.2141	-0.4474	-1.1786	0.2974	1
	3	-0.3990	-0.2229	-0.4456	-0.6085	-0.8691	1
	4	-0.3990	-0.2229	-0.4455	-0.6085	-0.8633	1
	5	-0.2992	-0.2385	0.5606	-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579	1.7605	-0.4985	-1
	49997	-0.3514	-0.2385	0.8697	-0.7542	-0.1276	-1
	49998	-0.1613	-0.2358	-0.4015	0.4556	0.3505	-1
	49999	-0.2802	-0.2385	0.3924	-0.3741	-1.1385	-1
	50000	0.5801	-0.2386	-0.3491	1.7288	-1.1628	-1
	Alpha	6.92E-01	3.49E-02	6.92E-03	0.00E+00	-1.53E-01	
	D(m,r)	6.92E-01	3.49E-02	6.92E-03	0.00E+00	-1.53E-01	

where, $\alpha_{ri} = \dfrac{D(m,r)}{D(r,r)}$ & $D(m,r) = \overline{p_m(n)p_r(n)} - \sum_{i=0}^{r-1} \alpha_{ri} D(m,i)$

FOS Overview

Step 2: Functional Expansion

- Due to the fact that $w_m(n)$ was calculated implicitly, a new function must be introduced as follows:

$$C(m) = \overline{y(n)w_r(n)}$$

- C(m) can now be calculated without reference to $w_m(n)$ using:

$$C(m) = \overline{y(n)p_m(n)} - \sum_{i=0}^{m-1} \alpha_{mi} C(i)$$

and the orthogonal weights are found from:

$$g_m = \frac{C(m)}{D(m,m)}$$

FOS Overview

Step 2: Functional Expansion

		$p_m(n)$					y(n)
		1	2	3	43	44	
	Flow Instance	**mean_fpktl**	**duration**	**min_active**	**std_fpktl**	**std_fiat**	**Truth**
Inclass	1	-0.4513	-0.2229	-0.4460	-0.7035	-0.3380	1
	2	-0.6794	-0.2141	-0.4474	-1.1786	0.2974	1
	3	-0.3990	-0.2229	-0.4456	-0.6085	-0.8691	1
	4	-0.3990	-0.2229	-0.4455	-0.6085	-0.8633	1
	5	-0.2992	-0.2385	0.5606	-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579	1.7605	-0.4985	-1
	49997	-0.3514	-0.2385	0.8697	-0.7542	-0.1276	-1
	49998	-0.1613	-0.2358	-0.4015	0.4556	0.3505	-1
	49999	-0.2802	-0.2385	0.3924	-0.3741	-1.1385	-1
	50000	0.5801	-0.2386	-0.3491	1.7288	-1.1628	-1
	C(m)	-3.57E-02	-1.94E-01	-2.74E-01	-3.66E-01	2.09E-01	
	gm	-2.71E-01	1.56E-01	-1.82E-01	4.18E-01	-1.94E-01	

Where, $C(m) = \overline{y(n)p_m(n)} - \sum_{i=0}^{m-1} \alpha_{mi} C(i)$ & $g_m = \dfrac{C(m)}{D(m,m)}$

FOS Overview

Search Algorithm

		$p_m(n)$					y(n)
		1	2	3	43	44	
	Flow Instance	**mean_fpktl**	**duration**	**min_active**	**std_fpktl**	**std_fiat**	**Truth**
Inclass	1	-0.4513	-0.2229	-0.4460	-0.7035	-0.3380	1
	2	-0.6794	-0.2141	-0.4474	-1.1786	0.2974	1
	3	-0.3990	-0.2229	-0.4456	-0.6085	-0.8691	1
	4	-0.3990	-0.2229	-0.4455	-0.6085	-0.8633	1
	5	-0.2992	-0.2385	0.5606	-0.4818	-0.8392	1
Outclass	49996	0.9461	-0.2269	-0.2579	1.7605	-0.4985	-1
	49997	-0.3514	-0.2385	0.8697	-0.7542	-0.1276	-1
	49998	-0.1613	-0.2358	-0.4015	0.4556	0.3505	-1
	49999	-0.2802	-0.2385	0.3924	-0.3741	-1.1385	-1
	50000	0.5801	-0.2386	-0.3491	1.7288	-1.1628	-1
	Qm	**1.27E-03**	**3.77E-02**	**7.48E-02**	**1.34E-01**	**4.35E-02**	
	MSE				**8.66E-01**		

- Algorithm searches all Q values and selects maximum
- Feature is chosen by maximum Q
- MSE is calculated using:

$$mse = \overline{\left(y(n) - \sum_{m=0}^{M} g_m w_m(n) \right)^2}$$

- This feature is now selected for inclusion in the model

FOS Overview

Stopping Criteria and Model Building

- Once a stopping criterion has been met the FOS algorithm calculates the non-orthogonal weights a_m of the functional series expansion are calculated recursively using:

$$a_m = \sum_{i=m}^{M} g_i v_i$$

where,

$$v_m = 1 \quad \& \quad v_i = -\sum_{r=m}^{i-1} \alpha_{ir} v_r \qquad m < i < M$$

- These weights are not used for feature selection, but are required to use FOS as a classifier
- FOS also returns the mean and STD's for the training data to permit the test data to be normalized against the training data

The FOSFOS Classifier