An Analysis of Clustering Objectives for Feature Selection Applied to Encrypted Traffic Identification

Carlos Bacquet, Nur A. Zincir-Heywood, and Malcolm I. Heywood

Abstract— **This work explores the use of clustering objectives in a Multi-Objective Genetic Algorithm (MOGA) for both, feature selection and cluster count optimization, under the application of flow based encrypted traffic identification. We first explore whether it is possible to achieve the performance of a gold standard model (i.e., classification objectives), using a MOGA based on clustering objectives. Then, we explore the performance gain (if it exists) of applying a logarithmic transformation to the data prior to running the MOGA. Results show that MOGA trained with clustering objectives can closely reproduce the behavior of a gold standard model, not only in terms of the selected features, but also in terms of the achieved detection rate and false positives rate, above 90% and less than 1% respectively. On the other hand, no gain was observed by applying logarithmic transformation to the data.**

I. Introduction

An important part of network management requires the accurate identification and classification of network traffic for decisions regarding both, bandwidth management and quality of service [1], [3]. A particularly interesting area in network traffic identification pertains to encrypted traffic, where the fact that the payload is encrypted represents an additional degree of uncertainty. Many traditional approaches to traffic classification rely on payload inspection, which becomes unfeasible under packet encryption. Alternatively, some approaches have used port numbers to identify application types, however, this practice has become increasingly inaccurate as user applications are now able to arbitrarily change the port number to deceive security mechanisms [2]. In short, the traditional approaches are unable to deal with the identification of encrypted traffic. In this work, Secure Shell (SSH) is chosen as an example encrypted application. While SSH is typically used to remotely access a computer, it can also be utilized for "tunneling, file transfers and forwarding arbitrary TCP ports over a secure channel between a local and a remote computer" [1]. These properties of SSH make it an interesting encrypted application to focus on, given that it shows similar behavior to popular encrypted applications such as Skype. However, unlike Skype, SSH is an open source protocol. This ensures that the ground truth is known regarding the traffic tested.

From the traffic identification perspective, we employ a Multi-Objective Genetic Algorithm (MOGA) that is used for the dual goal of (i) identifying the appropriate (flow) attribute/feature subspace and (ii) identifying traffic types by evolving clustering objectives. We first employed such a MOGA based approach to traffic identification in [5], under the assumption that the resulting clusters partition traffic into encrypted/not encrypted. Then in [4], we benchmark the performance of this MOGA against other unsupervised techniques existing in the literature, like K-Means, DBSCAN, and Expectations Maximization (EM). The results favored the proposed algorithm not only in terms of detection rate and false positives rate, but also in terms of the time required to train and test the models.

Alternatively, instead of using clustering objectives, we could use label information during training to establish a gold standard for cluster purity. Thus, by using Detection Rate (*DR*) and False Positives Rate (*FPR*) as objectives, we would avoid the uncertainty of building clusters based on a fitness function that indirectly pertains to the purpose of traffic identification. I.e., clustering data description is not necessarily the same as classification. However, the use of labels during the learning phase is computationally much more expensive as it involves iterative calculations of *DR* and *FPR*, and implies the availability of labeled training data which is expensive to produce. Thus, the first objective in this work is to explore whether it is possible to mimic the results obtained with the gold standard objectives (label driven learning), with clustering objectives.

The second objective of this work is to establish the significance of a prior attribute normalization on the resulting cluster quality. Network traffic under a flow based representation utilizes attributes representing different properties, such as time features and packet features. As a result, significant variation in attribute ranges is typically observed. Such diversity in attribute ranges is typically considered to have a negative impact on machine learning algorithms in general, resulting in the wide spread use of a prior attribute standardization/normalization to achieve a common variance or dynamic range across all attributes. We analyzed the effects of applying a logarithmic transformation to the data set to increase the homogeneity between the attributes. This is believed [18] to lead to better classification results.

II. Previous Work

Paxson presented in [18] a number of analytic models to describe the features associated with TELNET, NNTP, SMTP, and FTP connections. Three million TCP connections gathered at seven different sites were analyzed, focusing on attributes like bytes transfered and duration. In doing so, it was noticed that in the case of attributes with large ranges of values it is more meaningful to analyze the data

Carlos Bacquet, Nur A. Zincir-Heywood, and Malcolm I. Heywood are with the Faculty of Computer Science, Dalhousie University, 6050 University Avenue, Halifax NS, B3H 1W5, Canada (phone: +1 902 4942093; email: {bacquet, zincir, mheywood}@cs.dal.ca).

after applying a logarithmic transformation. Specifically, the significance of statistics such as mean and standard deviation are skewed towards attributes with the greater ranges. Given that there are several orders of magnitude, applying a log transformation reduces the corresponding dynamic range to the attributes with the most dissimilar ranges.

McGregor *et al.* [15] presented an unsupervised approach using Expectation Maximization (EM) clustering to classify network traffic represented by a set of flow attributes. Attributes that did not have an impact on the classification were removed. Using the Auckland VI trace they observed that the clustering showed some capabilities in grouping flows together by traffic type, but that more work needed to be done to derive better features to increase their performance. Zander *et al.* [22] also proposed traffic identification by using an unsupervised machine learning technique (autoclass). They employed the Auckland VI data set, and the NZIX-II and Leiozig traces. The feature selection was based on a sequential forward selection. The quality of the resulting classes was evaluated in terms of intra-class homogeneity, aiming to achieve good separation between different applications. They were able to achieve an average accuracy across all traces of 86.5%. Bernaille *et al.* [6] proposed an unsupervised clustering (K-means) online approach, based solely on the packet size of the first p packets. The best results were obtained with the first 5 packets, being able to correctly classify more than 80% of the flows of almost all of the tested applications. The authors did mention, however, that this approach is sensitive to the arrival order of the packets. Siqueira Junior *et al.* [12] focused on P2P traffic, presenting an unsupervised approach in which 249 features were analyzed, including statistics about the length of the packets and time between the packets. The feature selection was based on the Ratio F, which uses a ratio of two estimates, "dividing the variance mean of intra-group elements" and "the mean variance of inter-group elements". They were able to achieve 86.12% in trust, and 96.79% in accuracy. It is important to notice, however, that the Port server was included as one of the selected features. Yang *et al.* [20] applied a DBSCAN clustering algorithm, using 5 flow features. Feature selection was based on the effect that different features have on the classification accuracy, i.e., a wrapper methodology. They were able to achieve an accuracy of about 87%. None of these authors mention any logarithmic transformation to deal with the skewness of the data as described by Paxson. Moreover, most of them used both time, and packet related features.

Erman *et al.*, on the other hand, presented several unsupervised approaches for traffic classification in which the data was logged to deal with the "heavy tail distribution" of the features employed [7], [8], [9], [10]. In [8], the authors presented a comparison of an unsupervised machine learning approach, EM, to previously obtained results with a supervised learning method. They found the unsupervised technique outperformed the supervised technique by up to 9%, achieving an overall accuracy of up to 91%. Then in [7], they compared three unsupervised techniques: EM, K-Means, and DBSCAN, using the Auckland IV trace as well as traffic collected at the University of Calgary. The authors concluded that K-Means accuracy is only marginally lower than AutoClass (EM), but with a much faster training time. In [9] they proposed a semi-supervised method, in which the training was done with only a small percentage of labeled and a high percentage of unlabeled flows. They obtained high flow and byte accuracy, greater than 90%, using a backward greedy feature selection method. They observed that flow features that have time components should be avoided as "they are less likely to be invariant across different networks" [9]. Finally, Yingqiu *et al.* [21] also applied a logarithmic transformation to the data. The authors presented an unsupervised K-means approach, combined with feature reduction techniques such as CFS, Consistency-based subset evaluation, backward and forwards greedy search, among others. With traffic collected at a research facility, they obtained improvements of at least 10% accuracy after applying log transformation, with an overall accuracy of up to 90%.

To the best of our knowledge, this is the first work that, without using any payload information or port number, identifies encrypted traffic with a multi-objective genetic algorithm, and compares its performance to that of an equivalent gold standard model. Furthermore, this is also the first work that assesses the suitability of a logarithmic transformation to the data for a genetic algorithm applied to encrypted traffic identification.

III. Methodology

This section first discusses the employed data set, followed by a description of the flow features obtained from it. We then explain the multi-objective genetic algorithm, MOGA, which leads to an analysis of its clustering objectives, detailed in the fitness subsection. This section is concluded with an outline of the logarithmic transformation used for attribute wise standardization.

A. Data Set

The employed data set was captured by the Dalhousie University Computing and Information Services Centre (UCIS) in January 2007 on the campus network between the university and the commercial Internet. Dalhousie University is one of the largest universities in the Atlantic region of Canada, with more than 15,000 students and about 3,300 faculty and staff. Data privacy related issues required that the data was filtered to scramble the IP addresses and each packet was further truncated to the end of the IP header so that all the payload was excluded. Furthermore, the checksums were set to zero since they could conceivably leak information from short packets. However, any length information in the packet was left intact. Dalhousie traces were labeled by UCIS with a commercial classification tool, PacketShaper, which is a deep packet analyzer, i.e., it analyzes the packet payload [17]. Given that the handshake part of SSH protocol is not encrypted, we can confidently assume that the labeling of the data set is completely accurate and provides the ground truth for testing purposes. Again, we emphasize that our work did not consider any information from the handshake phase nor any part of the payload, IP addresses, or port numbers. Also, we focus on SSH as a case study, there is nothing in the approach that ties us to the SSH protocol specifically. However, the fact that the SSH's handshake is not encrypted, allowed us to compare our obtained results with those obtained through payload inspection. In order to build training data we sampled the Dalhousie traces. The training data for all experiments consisted of 12250 flows, including SSH, MSN, HTTP, FTP, and DNS. The test data, on the other hand, was the entire data set (more than 18,500,000 flows) and consisted of flows from each of those applications, plus flows that belonged to any of the following additional applications: RMCP, Oracle SQL*NET, NPP, POP3, NETBIOS Name Service, IMAP, SNMP, LDAP, NCP, RTSP, IMAPS and POP3S.

B. Flow Generation

Flows are defined by sequences of packets that present the same values for source IP address, destination IP address, source port, destination port and type of protocol. In this work, each flow is described by a set of statistical features and associated feature values. A feature is a descriptive statistic that can be calculated from one or more packets. NetMate [16], an open source tool, was used to generate flows, and compute feature values. Table 1 shows the 38 features obtained from NetMate. Flows are bidirectional with the first packet determining the forward direction. Since flows are of limited duration, in this work UDP flows are terminated by a flow timeout, and TCP flows are terminated upon proper connection teardown or by a flow timeout, whichever occurs first. A 600 second flow timeout value was employed here; where this corresponds to the IETF Realtime Traffic Flow Measurement working groups architecture [11]. It is important to mention that only UDP and TCP flows are considered. Specifically, flows that have no less than one packet in each direction, and transport no less than one byte of payload. Again, payload data and features like IP addresses and source/destination port numbers were excluded from the feature set to ensure that the results were not dependent on such biases.

C. Genetic Algorithm for Feature Selection and Clustering

In this work we took the MOGA framework proposed by Kim *et al.* [13] for feature selection and cluster count optimization, but adapted the MOGA as proposed by Kumar *et al.* [14]. The latter converges towards the Pareto-front (set of non-dominated solutions) without any complex sharing/niching mechanism. One specific property of this Genetic Algorithm (GA) is the utility of a steady-state GA, thus, only two members of the population are replaced at a time under an elitist replacement model. Like most GA's, MOGA starts with a population of individuals (potential solutions to a problem), and incrementally evolves that population into better individuals, as established by the fitness criteria. Fitness is naturally relative to the population.

TABLE I

FLOW FEATURES

ind.	Feature Name	Abreviation
1	protocol (tcp, udp)	*proto*
2	total forward packets	*total_fpackets*
3	total forward volume	*total_fvolume*
4	total backward packets	*total_bpackets*
5	total backward volume	*total_bvolume*
6	min forward packet length	*min_fpktl*
7	mean forward packet length	*mean_fpktl*
8	max forward packet length	*max_fpktl*
9	std dev forward packet length	*std_fpktl*
10	min backward packet length	*min_bpktl*
11	mean backward packet length	*mean_bpktl*
12	max backward packet length	*max_bpktl*
13	std dev backward packet length	*std_bpktl*
14	min forward inter arrival time	*min_fiat*
15	mean forward inter arrival time	*mean_fiat*
16	max forward inter arrival time	*max_fiat*
17	std dev forward inter arrival time	*std_fiat*
18	min backward inter arrival time	*min_biat*
19	mean backward inter arrival time	*mean_biat*
20	max backward inter arrival time	*max_biat*
21	std dev backward inter arrival time	*std_biat*
22	duration of the flow	*duration*
23	min active	*min_active*
24	mean active	*mean_active*
25	max active	*max_active*
26	std dev active	*std_active*
27	min idle	*min_idle*
28	mean idle	*mean_idle*
29	max idle	*max_idle*
30	std dev idle	*std_idle*
31	sub flow forward packets	*sflow_fpackets*
32	sub flow forward bytes	*sflow_fbytes*
33	sub flow backward packets	*sflow_bpackets*
34	sub flow backward bytes	*sflow_bbytes*
35	forward push counter	*fpsh_cnt*
36	backward push counter	*bpsh_cnt*
37	forward urg counter	*furg_cnt*
38	backward urg counter	*burg_cnt*

In order to model the problem of feature selection, each individual in the population represents a subset of features *f* and a number of clusters *K*. Specifically, an individual is a 120 bit binary string, where bits between the first bit and the 38th bit represent the features to include, and the remaining bits represent the *K* number of clusters. Bits of the individuals in the initial population are initialized with a uniform probability distribution. For feature selection, a "one" implies that the feature at that index (from Table 1) is included, and a "zero" ignores the feature. The *K* number of clusters, on the other hand, is obtained from the number of "ones" (as opposed to "zeros") contained between the 39th bit and the 119th bit. Clusters are identified using the standard K-means algorithm, using the subset of features *f*, and the number of clusters *K*, as the input for the K-means algorithm. We used the K-means algorithm provided by Weka [19]. The fitness of the individual will then depend on how well the resulting clusters perform in relation to four predefined clustering objectives: *Fwithin*, *Fbetween*, *Fclusters*, and *Fcomplexity* (see section III. D). Fitness evaluation assumes a multi-objective approach, typically resulting in the

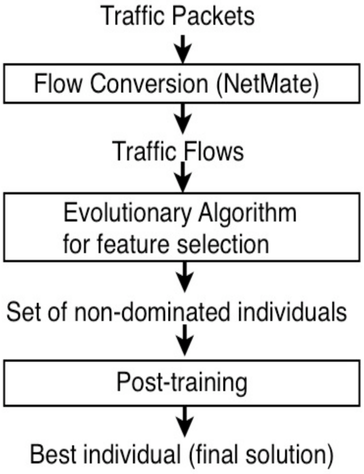
Traffic Packets
Flow Conversion (NetMate)
Traffic Flows
Evolutionary Algorithm for feature selection
Set of non-dominated individuals
Post-training
Best individual (final solution)

Fig. 1. System Diagram

identification of the Pareto front, or a set of non-dominated solutions. Informally, a solution is said to dominate another if it has higher values in at least one of the objectives, and is at least as good in all the others. After the objective values for each individual have been assessed, individuals are assigned with ranks, which indicate how many individuals dominate that particular individual. Thus, fitness of the individuals is inversely proportional to their ranks, which is used to build a roulette wheel that is ultimately used for parental selection.

A population of 250 individuals is evolved for 5000 epochs, with a mutation rate of 0.6% and a uniform crossover operator. The evolutionary component of the algorithm is then terminated and the best individual (the one that better identifies SSH traffic) out of the set of non-dominated solutions (individuals whose ranks equal to 1) is identified in the post-training phase. We take each individual from the set of non-dominated solutions, apply K-Means with its proposed set of features *f* and number of clusters *K*, and label its clusters as SSH or NON-SSH. If the majority of the flows in a cluster have SSH labels, then that cluster is labeled as SSH, otherwise it is labeled as non-SSH. The post-training phase is then entered and consists of testing each of the non-dominated individuals in our training data (used to build the clusters on), to identify the solution with best classification rates. The entire system is displayed in Figure 1, and its evolutionary component in Figure 2.

D. Fitness Function

The fitness of the individual depends on how well the resulting clusters perform in relation to the following four predefined clustering objectives:

1) *Fwithin*: Measures cluster cohesiveness, the more cohesive the better. For this purpose the average standard deviation per cluster is assumed. That is, the sum of the standard deviations per feature over the total number of employed features. Then *Fwithin* will be the number of clusters in a solution, *K*, over the sum of all the clusters' average standards.

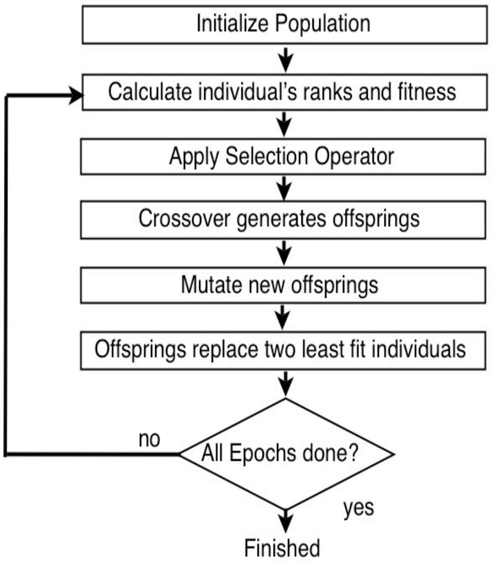
Initialize Population
Calculate individual's ranks and fitness
Apply Selection Operator
Crossover generates offsprings
Mutate new offsprings
Offsprings replace two least fit individuals
All Epochs done?
no
yes
Finished

Fig. 2. Evolutionary Component Diagram

2) *Fbetween*: Measures how separate the clusters are from each other, the more separated the better. For each pair of clusters *i* and *j*, we calculate their average standard deviations and we also calculate the euclidean distance between their centroids. Then, *Fbetween* for clusters *i* and *j* is:

$$Fbetween(i,j) = \frac{EuclideanDistanceFrom\_i\_to\_j}{\sqrt{(AveStdDev_i)^2 + (AveStdDev_j)^2}}$$

Thus, *Fbetween* will be the sum of all pairs of cluster's *Fbetween(i,j)*, over *K*.

3) *Fclusters*: Measures the number of clusters *K*, "Other things being equal, fewer clusters make the model more understandable and avoid possible over fitting" [13].

$$Fclusters = 1 - \frac{K - Kmin}{Kmax - Kmin}$$

Kmax and *Kmin* are the maximum and minimum number of clusters.

4) *Fcomplexity*: Measures the amount of features used to cluster the data, this objective aims at minimizing the number of selected features.

$$Fcomplexity = 1 - \frac{d-1}{D-1}$$

D is the dimensionality of the whole dataset and *d* is the number of employed features.

In short, this model assumes that building fewer high quality clusters in terms of low intra-cluster distance and high inter-cluster distance, and selecting fewer features, will lead to a better data description. Conversely, post training performance evaluation is based on detection rate (*DR*) and false positives rate (*FPR*) defined by:

$$DR = 1 - \frac{\#false\_negatives}{total\_number\_of\_SSH\_flows}$$

$$FPR = \frac{\#false\_positives}{total\_number\_of\_non\_SSH\_flows}$$

where false_negatives means SSH traffic incorrectly classified as non-SSH traffic, and false positives means non SSH traffic incorrectly classified as SSH traffic.

In order to test whether these clustering objectives can indeed optimize the system, and still achieve good classification rates, a second set of objectives is defined, in which *Fwithin* and *Fbetween* are replaced with *DR* and *FPR* as clustering objectives, but keeping the *Fcomplexity* and *Fclusters* objectives. By doing so, the learning is guided towards the classification objectives, rather than towards generating high quality clusters. However, this second approach has the disadvantage of (i) being computationally much more expensive, as the calculation of *DR* and *FPR* requires a test run over the entire training data for each individual evaluation, and (ii) being label dependent, requiring a labelled training data that is expensive to produce. We are, therefore, interested in identifying under what conditions (if any) the purely cluster style objectives are able to approach the performance of the explicitly label driven approach to cluster identification. Such an analysis will consider both classification performance and attribute support.

E. Log Transformation

It is important to notice that not all 38 features from Table 1 belong to the same type. Instead, there is a mix of time features, with packet features, and others, resulting in significant range differences between their values. These differences can account for up to seven orders of magnitude between their average values. Presumably, differences of this order could bias the design of clusters towards some of the features, not because of class discriminating characteristics, but because of their range in values. I.e., clustering is a data description process, thus will be biased to modeling the most frequent/dominant properties in the data. Whether this is an appropriate bias for traffic discrimination is unknown. To test the effect of a logarithmic transformation to reduce the effect of these range differences, we conducted a second set of experiments in which a log transformation was applied to each attribute. The logged data will observe much lower inter attribute variation, thus less bias towards any feature in particular, potentially resulting in different features being identified as appropriate support for clustering.

IV. Results

We conducted a total of four sets of experiments. The first set consisted of running the MOGA with the original clustering objectives, without applying a logarithmic transformation. The second set consisted of running MOGA with the original objectives, but after applying a logarithmic transformation to the data. The third and fourth sets of experiments consisted of running MOGA with the gold standard objectives, without applying logarithmic transformation in the third set, and after applying logarithmic transformation in the fourth set. Each set of experiments consisted of 25 independent runs, from which we took the non-dominated individuals. We then combined those non-dominated individuals, and took a subset that we consider the best individuals, in the case of the runs with the original objectives, the best individuals were those with the highest intra and inter cluster distances, and in the case of the gold standard model, the best individuals were those with *DR* above 90% and *FPR* under 0.6%.

Figures 3 to 6 show the features selected by the best individuals in each set of experiments. The vertical axis contains the 38 available features from Table 1, and the horizontal axis represents the percentage of best individuals that employed each feature. Figure 3 demonstrates that MOGA with the original objectives selects a subset of the features identified by the gold standard model, Figure 5, both without logarithmic transformation. Both time and packet related features are selected by the best individuals. It appears that the standard deviation forward inter arrival time, *std_fiat*, and the minimum backward inter arrival time, *min_biat*, were selected by almost all of the individuals in both sets of experiments. We can also see how with the exception of the features between *duration*, and *std_idle*, the remaining 26 features are selected with a very similar frequency. On the other hand, once the logarithmic transformation is included, we observe that the features selected with the original objectives significantly differ from the features selected by the gold standard model; compare Figures 4 and 6. With the exception of the standard backward inter arrival time, *std_biat*, the other selected features seems to be the opposite with the gold standard model.

To further explore these observations, we conducted a post training evaluation of classification performance, as described in the methodology section. We test the best non-dominated individuals per set of experiments in the training data, in order to identify the ones with the best classifications rates. The individual with the best classification performance in post training becomes the final solution for that set of experiments, and it is then tested in the entire data set.

Figures 7 to 10 show the plot of the best individuals performances during the post training phase. The vertical access represents the *DR* and the horizontal axis represents the *FPR*. The final solution per experiment is marked with a darker square on each plot. From Figure 7, we can observe that with the original objectives and without the logarithmic standardization, the final solution achieves a *DR* of 93.5%

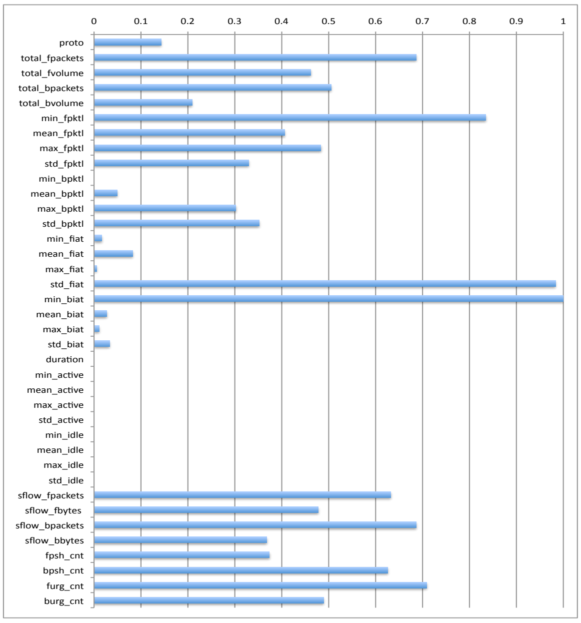

Fig. 3. Features selected with original objectives without log. standardiz.

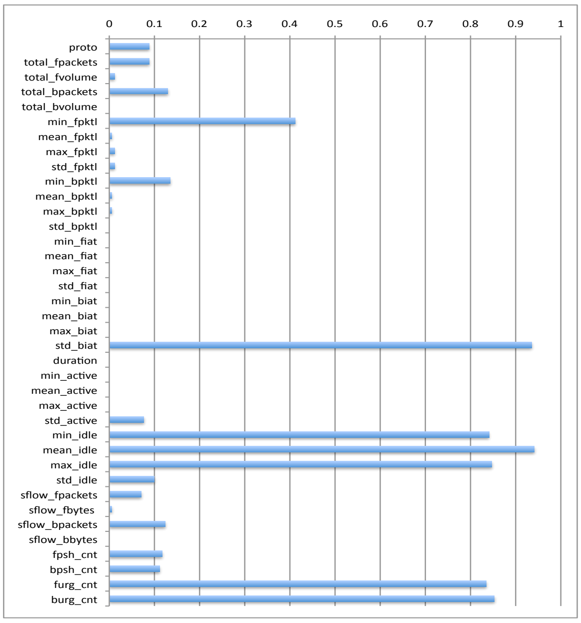

Fig. 4. Features selected with original objectives with log. standardiz.

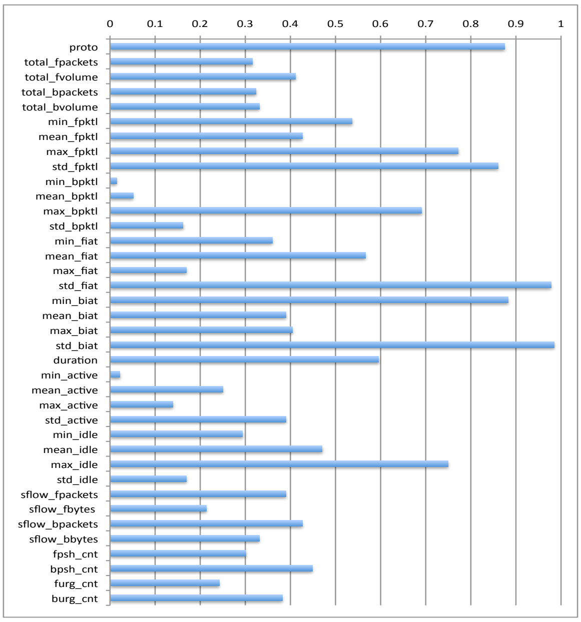

Fig. 5. Features selected with gold standard without log. standardiz.

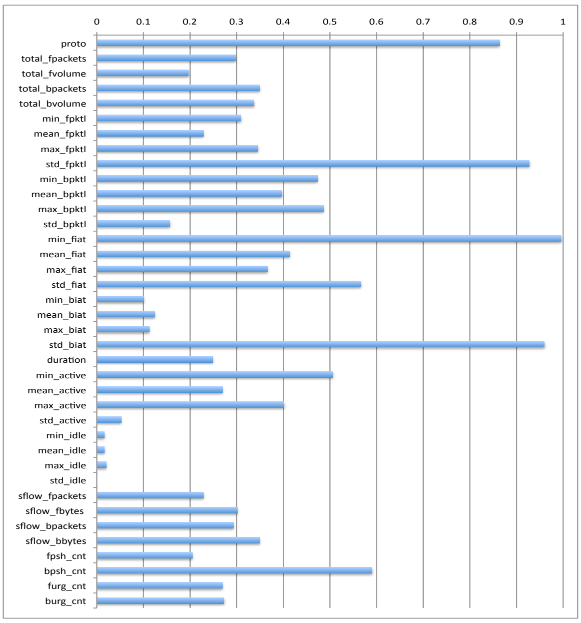

Fig. 6. Features selected with gold standard with log. standardiz.

and a *FPR* of 0.25% in post training. That same individual achieves a *DR* of 90.0% and a *FPR* of 0.4% when tested on the entire data set. In comparison, the gold standard model achieves a *DR* of 93.9% and a *FPR* of 0.22% in post training (Figure 9), and a *DR* of 90.0% and a *FPR* of 0.8% when tested on the entire data set. We can conclude that the original objectives are capable of closely mimicking the performance of the gold standard model. This does not come as a surprise, as we already observed that the original model generally selects a similar set of features as the gold standard model.

On the other hand, we can observe from Figures 8 and 10 that the results achieved in post training when applying the logarithmic standardization, differ between the original objectives and the gold standard objectives. With the original objectives, the final solution achieves a *DR* of 95.4% and a *FPR* of 1.1% in post training (Figure 8). Notice that the rest of the individuals do not appear on the plot for being out of scale with much larger *FPR*. That same individual achieves a *DR* of 91.4% and a *FPR* of 17.0% when tested in the entire data set, which is a prohibitory high *FPR*. The gold standard model, Figure 10, achieves a *DR* of 94.4% and a *FPR* of 0.16% in post training, and a *DR* of 91.0% and a *FPR* of 0.2% when tested in the entire data set. Thus, we can conclude that because the original objectives under a logged

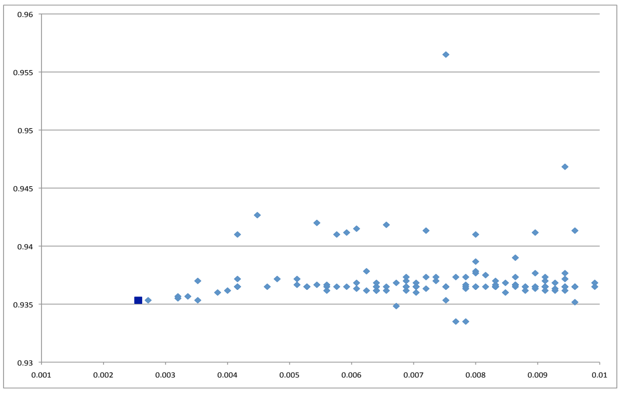

Fig. 7. Post training with original objectives without log. standardization. *DR* (y-axis) over 93 to 96% range; *FPR* (x-axis) over 0.1 to 1% range.

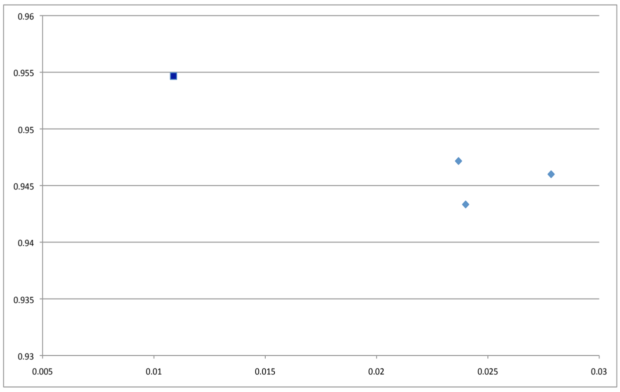

Fig. 8. Post training with original objectives with log. standardization. *DR* (y-axis) over 93 to 96% range; *FPR* (x-axis) over 0.5 to 3% range.

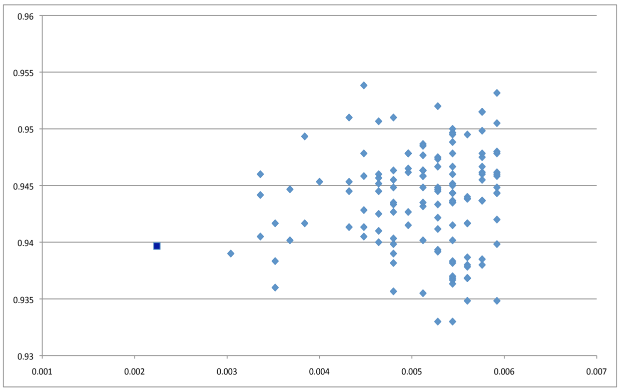

Fig. 9. Post training with gold standard without log. standardization. *DR* (y-axis) over 93 to 96% range; *FPR* (x-axis) over 0.1 to 0.7% range.

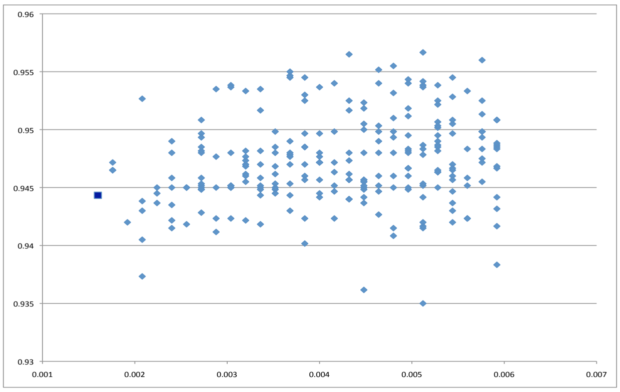

Fig. 10. Post training with gold standard with log. standardization. *DR* (y-axis) over 93 to 96% range; *FPR* (x-axis) over 0.1 to 0.7% range.

data do not select the same features as the gold standard, the performance of the MOGA with the original clustering objectives is inferior to that of the gold standard model. All the test results are summarized in Table 2.

TABLE II

FINAL SOLUTIONS TEST RESULTS

Experiment	DR	FPR
Original Objectives	90.0%	0.4%
Gold Standard	90.0%	0.8%
Original Objectives logged	91.4%	17.0%
Gold Standard logged	91.0%	0.2%

V. CONCLUSIONS

With regards to our first goal, we observe that the original cluster style objectives can quite closely mimic the behavior of the gold standard classifier style objectives in the non logged experiments. In that case, we observe that not only both sets of experiments selected a similar set of features, but also that the results from testing both models' best individuals on the entire data set show very similar performances. The gold standard model achieves a *DR* of 90.0% with a *FPR* of 0.8% when tested on the entire data set, and the original objectives also achieve a *DR* 90.0% with a *FPR* of 0.4%.

With regards to our second goal, on the other hand, we observe that after applying a log transformation to the attributes, the original objectives do not mimic the behavior of the gold standard model. Both sets of experiments do not select similar features. Moreover, the classification performance achieved with the original objectives is considerably lower than the logged gold standard model. The gold standard model achieved a *DR* of 91.0% with a *FPR* of 0.2% when tested on the entire data set, whereas the original model achieved a *DR* of 91.4% with a *FPR* of 17.0%, which is prohibitory high.

Finally, we observe that the best results are achieved consistently with a mix of time and packet related features. In particular, the features *std_fiat* and *min_biat* were employed by almost all the best individuals in the non logged experiments.

For future work we consider to modify the MOGA into a hierarchical clustering approach, and also to focus on other types of encrypted traffic, such as Skype.

ACKNOWLEDGMENT

This work is supported by ISSNet - NSERC Strategic Network and the CFI new opportunities program. Our thanks to John Sherwood, David Green and Dalhousie UCIS team for providing us the anonymozied Dalhousie traffic traces. All research was conducted at the Dalhousie Faculty of Computer Science NIMS Laboratory, http://www.cs.dal.ca/projectx.

REFERENCES

[1] R. Alshammari and A. Zincir-Heywood. A flow based approach for ssh traffic detection. In *Systems, Man and Cybernetics, IEEE International Conference on*, pages 296–301, Oct. 2007.

[2] R. Alshammari and A. Zincir-Heywood. Investigating two different approaches for encrypted traffic classification. In *Privacy, Security and Trust, 2008. PST '08. Sixth Annual Conference on*, pages 156–166, Oct. 2008.

[3] R. Alshammari and A. Zincir-Heywood. Generalization of signatures for ssh encrypted traffic identification. In *IEEE Symposium Series on Computational Intelligence on Cyber Security*, 2009.

[4] C. Bacquet, K. Gumus, D. Tizer, A. Zincir-Heywood, and M. Heywood. A comparison of unsupervised learning techniques for encrypted traffic identification. *Journal of Information Assurance and Security*, 5:464–472, 2010.

[5] C. Bacquet, A. Zincir-Heywood, and M. Heywood. An Investigation of Multi-objective Genetic Algorithms for Encrypted Traffic Identification. In *Computational Intelligence in Security for Information Systems: Cisis' 09, 2nd International Workshop Burgos, Spain, September 2009 Proceedings*, pages 93–100. Springer, 2009.

[6] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36(2):23–26, 2006.

[7] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *MineNet '06: Proceedings of the SIGCOMM workshop on Mining network data*, pages 281–286, New York, NY, USA, 2006. ACM.

[8] J. Erman, A. Mahanti, and M. Arlitt. Qrp05-4: Internet traffic identification using machine learning. In *Global Telecommunications Conference. GLOBECOM '06. IEEE*, pages 1–6, Dec 2006.

[9] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/realtime traffic classification using semi-supervised learning. *Perform. Eval.*, 64(9-12):1194–1213, 2007.

[10] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson. Identifying and discriminating between web and peer-to-peer traffic in the network core. In *Proceedings of the 16th international conference on World Wide Web*, pages 883–892. ACM, 2007.

[11] IETF. http://www.ietf.org/.

[12] G. Junior, J. Maia, R. Holanda, and J. De Sousa. P2P Traffic Identification using Cluster Analysis. In *Global Information Infrastructure Symposium. GIIS 2007. First International*, pages 128–133, 2007.

[13] Y. Kim, W. N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 365–369, New York, NY, USA, 2000. ACM.

[14] R. Kumar and P. Rockett. Improved sampling of the pareto-front in multiobjective genetic optimizations by steady-state evolution: a pareto converging genetic algorithm. *Evol. Comput.*, 10(3):283–314, 2002.

[15] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. *Lecture Notes in Computer Science*, 3015:205–214, 2004.

[16] NetMate.
http://www.ip-measurement.org/tools/netmate.

[17] PacketShaper.
http://www.packeteer.com/products/packetshaper.

[18] V. Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking (TON)*, 2(4):316–336, 1994.

[19] WEKA. http://www.cs.waikato.ac.nz/ml/weka/.

[20] C. Yang, F. Wang, and B. Huang. Internet traffic classification using dbscan. *Information Engineering, International Conference on*, 2:163–166, 2009.

[21] L. Yingqiu, L. Wei, and L. Yunchun. Network traffic classification using k-means clustering. In *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pages 360–365, Aug. 2007.

[22] S. Zander, T. Nguyen, and G. Armitage. Automated traffic classification and application identification using machine learning. In *Conference on Local Computer Networks*, volume 30, pages 250–257. IEEE, 2005.