CluClas: Hybrid Clustering-Classification Approach for Accurate and Efficient Network Classification

Adil Fahad, Kurayman Alharthi, Zahir Tari, Abdulmohsen Almalawi and Ibrahim Khalil
RMIT University School of Computer Science and IT
Melbourne, Australia
Emails: adil.fahad@rmit.edu.au, kareaman@gmail.com, zahir.tari@rmit.edu.au
abdul.almalawi@student.rmit.edu.au, Ibrahim.khalil@rmit.edu.au,

***Abstract*—The traffic classification is the foundation for many network activities, such as Quality of Service (QoS), security monitoring, Lawful Interception and Intrusion Detection Systems (IDS). A recent statistics-based approach to address the unsatisfactory results of traditional port-based and payload-based approaches has attracted attention. However, the presence of non-informative attributes and noise instances degrade the performance of this approach. Thus, to address this problem, in this paper, we propose a hybrid clustering-classification approach (called CluClas) to improve the accuracy and efficiency of network traffic classification by selecting informative attributes and representative instances. An extensive empirical study on four traffic data sets shows the effectiveness of our proposed approach.**

I. Introduction

The process of classifying network traffic into a set of categories according to the applications which generate them is known as traffic classification. Traffic classification methods are essential tools for improving the quality of service (QoS) and enhancing system security, which have been widely studied in recent years. Traditional network classification methods [21], [14], including port-based methods that directly identify applications by port number, the packet headers and deep packet inspection methods, have shown a number of drawbacks, especially with the rapid evolution of new traffic applications. For example, new applications and encrypted traffic can easily evade detection by using a technique like dynamic port assignment. To address this limitation, a new method based on statistical characteristics of IP flows [32], [13], [27] (e.g. mean and variance of packet size and inter-packet time in traffic flows) and machine learning algorithms, shows promising results.

The classification process for statistics-based classifiers can be divided into two phases including training and testing [32], [13], [27]. The former phase feed the training data to learning algorithms to build classifier models, while the latter phase is used to predict the application types based on the generated model obtained from the training phase. Two types of learning methods can be used for both training and testing phases, depending on whether the class labels are available or not. For example, supervised learning algorithms are used with labelled data; on the other hand, unsupervised learning is used with unlabelled data.

To the best of our knowledge, there are a limited number of studies which combine the advantages of both supervised and unsupervised learning algorithms to improve the performance of network classifiers. Thus, in this paper, we propose a new hybrid clustering-classification approach (namely CluClas) to eliminate noise attributes and instances for better network classification. In particular, our proposed approach first pre-processes the traffic data and removes redundant and irrelevant attributes from the global perspective. Second, we apply a K-means clustering algorithm [6] on the training set to discard noise instances and select the centroid of each cluster as representative training instances. This step is important for some learning algorithms which may be noise-fragile, and also to reduce the amount of computation for such learning algorithms. Finally, using a Hidden Markov Mode (HMM) [19], a network classifier is built on the representative training instances, which can be used for evaluating new traffic in real-time.

Four publicly available traffic data sets [25], [20], [26], [1] are used to evaluate our proposed CluClas approach. The experimental results show that our approach achieved better results in comparison to individual methods, including K-means and HMM.

The rest of the paper is organized as follows. Section II presents related work in the area of network classification and machine learning. Section III describes our hybrid clustering-classification approach. Experimental evaluation and discussion of the results are presented in Section IV. Section V presents the conclusion and outlines of future work

II. Related Work

Classication techniques based on Machine Learning are divide into two categories: supervised and unsupervised. An extensive study of ML and traffic classification can be found in the survey of Nguyen et al. [22]

For supervised algorithms [32], [20], [27], the class of each traffic flow must be known before the learning stage. A classification model is built using a training set of example instances that represents each class. The model is then able to predict class membership for new instances by examining the feature values of unknown flows.

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

Supervised learning creates knowledge structures that support the task of classifying new instances into pre-defined classes [23]. A group of sample instances, pre-classified into classes, are being provided to the learning machine. A classification model is the output of the learning process. A classification model is constructed by analysing and generalizing from the provided instances. As a result, supervised learning's main focus is on modelling the input and output relationships. Its main goal is to find a mapping from input features to an output class. The knowledge learnt can be presented as classification rules, a decision tree, a flowchart, etc. This knowledge will be used later to classify a new instance. There are two major steps in supervised learning: training and testing. Training is a learning phase which analyses the provided data, which is called the training data set, and builds a classification model. Testing is also known as classifying. In this phase, the model that has been built in the training phase is used to classify previously unseen instances.

Unlike classification techniques in supervised machine learning, clustering methods [17], [18] do not use any pre-defined training instances; instead, they find natural clusters in the data using internalized heuristics [9]. McGregor et al. in [18] is one of the earliest works to use unsupervised machine learning techniques to group network traffic flows using transport layer attributes with Expectation Maximization (EM) method. Even though the authors do not evaluate the accuracy of the classification as well as which traffic flow attributes produce the best results, this approach clusters traffic with similar observable properties into different application types. In [31], Zander et al. extend this work by using another Expectation Maximization (EM) algorithm called AutoClass and analyse the best set of attributes to use. Both [18] and [31] with Bayesian clustering techniques were implemented by an EM algorithm which is guaranteed to converge to a local maximum. To find the global maximum, AutoClass repeats EM searches starting from pseudo-random points in parameter space, thus it performs much better than the original EM method. Both the early works in [7] and [31] have shown that cluster analysis has the ability to group Internet traffic using only transport layer characteristics. Erman et al in [10] proposed to use K-Mean and DBSCAN clustering methods to evaluate the predicating performance. They also demonstrated that both K-Mean and DBSCAN perform better and work more quickly than the clustering method of AutoClass used in [31]. However, these unsupervised techniques are not as good as supervised techniques. Thus, in this paper, we will exploit the advantages of both supervised and unsupervised techniques for better accuracy of network classifiers.

III. The Proposed Hybrid Clustering-Classification Approach

In this paper, a CluClas approach is proposed to improve the accuracy and the efficiency of network traffic classification. The CluClas is based on combining the advantages of clustering and classification algorithms. The following subsections illustrate the process and details of our CluClas approach.

A. Overview of proposed approach

Fig. 1 illustrates the overall view of the proposed CluClas approach. In particular, the proposed approach is comprised of three phases: (1) the pre-processing of the data to discard and remove irrelevant and redundant attributes of the original data from a global perspective, (2) identifying the most representative instances with the aim of improving the efficiency of the learning process as well as the overall prediction accuracy by partitioning the samples belong to a single class only and extracting the centroid of each cluster to act as a representative instance of that application class, and (3) building a network traffic classification model based on the Hidden Markov Model (HMM).

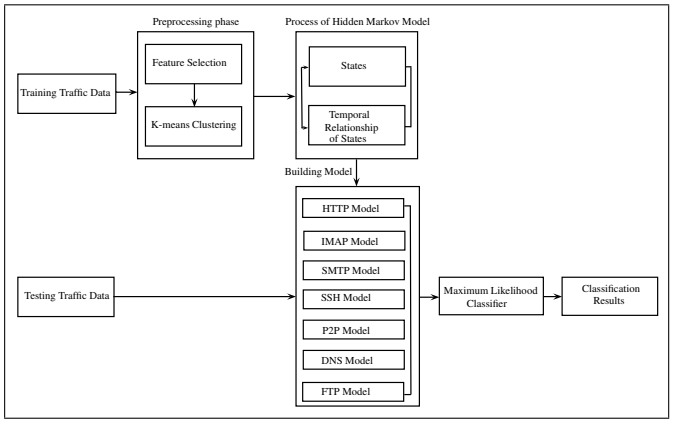


Fig. 1: Overview of the proposed CalClus approach

B. Discarding irrelevant and redundant attributes

The quality of the data always affects the accuracy and execution of the machine learning algorithm during the training phase [28], [12]. This is due to the presence of irrelevant and redundant attributes in the data. Thus, to discard these non-informative attributes, feature selection techniques are used. Feature selection (FS) techniques can be divided into two main categories: the wrapper method and the filter method [16], [3]. The former method [3] employs an existing ML technique [2] as a classifier and uses the classifier's accuracy as the evaluation measure to select the best possible attributes. Such a method tends to be not only computationally expensive, but also inherits a *bias* toward the predetermined learning algorithm. The latter method [3] relies on the natural characteristics of the data (e.g. correlation) and does not require a predetermined mining algorithm to select feature subsets. As a result, this method does not inherit the *bias* of any mining algorithm, and it is also computationally effective. However, the filter techniques eliminate both irrelevant and redundant attributes from a local perspective, and thus it can be tricked in a situation where the dependence between a pair of attributes is weak, but the total inter-correlation of one attribute to the others is strong. Thus, in this paper, we introduce a new FS approach to select informative attributes from a global perspective [5], [11]. The process of discarding

irrelevant and redundant attributes from a global perspective and only keeping the optimal attributes is presented in Table I.

Table I shows the procedure of discarding the irrelevant and redundant attributes in two parts. In the first part, the algorithm eliminates irrelevant attributes by applying the *symmetrical uncertainty* correlation measure. In particular, the *symmetrical uncertainty* correlation evaluates the reliability of each individual attribute for predicting the accurate class label as follows:

$$H = (Y) = -\sum_{y \in Y} p(y) \log_2 p(y), \tag{1}$$

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2 (y|x), \tag{2}$$

$$\begin{aligned} gain &= H(Y) - H(Y|X) \\ &= H(X) - H(X|Y) \\ &= H(Y) + H(X) - H(X,Y) \end{aligned} \tag{3}$$

Information gain is considered to be a bias in attributes with more values. Thus the correlation value in Equation 1 should be normalized to the range $[0, 1]$ as follows:

$$SU(X,Y) = 2.0 \times \left[\frac{gain}{H(Y) + H(X)} \right] \tag{4}$$

Note that attributes whose *symmetrical uncertainty's* value is zero are removed, which means that attributes do not have the power to distinguish between traffic classes. The remaining attributes are then ranked in descending order according to their value of symmetrical uncertainty, and the mean of these attributes is calculated, μ_{rv}. In the second part, the inter-correlation between attributes is computed and the total values of the symmetrical uncertainty related to that attribute are added. The weight factor w is computed (as in line (5.a)) to be used for selecting optimal attributes from a global perspective. Finally, attributes greater than zero are selected, which means that they not only can accurately predict the class, but also have a low correlation to other attributes.

TABLE I: The process of selecting optimal attributes globally.

Input:
Given the input data set D
Specify the number of optimal features K.
Remove irrelevant attributes
1. Compute the mutual information for each attribute, x_i.
1.a $SU(x_i, Y) = 2.0 \times \left[\frac{gain}{H(Y)+H(x_i)} \right]$.
2. Rank the attributes in descending order based on the value of $SU(Y|x_i)$.
3. Select x_i whose relevant score is greater than 0.
3.a If $SU(x_i, Y) > 0$ then $X_{rr} = X_{rr} \cup \{x_i\}$.
4. Compute the mean of relevant scores.
4.a $\mu_{rv} = \frac{\sum_{i=0}^{|X_{rr}|} SU(x_i, Y)}{|X_{rr}|}$.
Remove redundant attributes
5. For each $x_j \in X_{rr}$.
5.a Compute the inter-correlation between attributes, as
$SU(x_i, x_j) = 2.0 \times \left[\frac{gain}{H(x_i)+H(x_j)} \right]$.
6. Compute the mean of the redundance scores as
6.a $\mu_{rd} = \frac{\sum_{i=0}^{|X_{rr}|} SU(x_i, x_j)}{|X_{rr}|}$.
7. Compute the weight value based on both the relevant and redundant scores.
7.a $w = \frac{\mu_{rd}}{\mu_{rr}}$.
8. For each $x_j \in X_{rr}$.
8.a Use the weight value to calculate the importance of attributes from a global prospective.
$S(x_i) = w \cdot x^i_{rv} - x^i_{rd}$.
8.b Select the optimal attributes $S_{optimal}$.
If $S(x_i) > 0$ then $S_{optimal} = S_{optimal} \cup x_i$.
9. Return the final set of optimal attributes, $S_{optimal}$.

C. Identifying representative instances In CluClas

Original training data set may contain noise instances which can affect noise-fragile learning algorithms. Thus, an instance selection step is important to discard as many instances as possible without significantly degradation of reduced data set for learning processes. To do so, we apply a clustering algorithm on the training set and select only the centroid of each cluster, which can act as a representative instance. As a result of selecting only representative instances, we can not only reduce the amount of computation, but also improve the accuracy of a machine learning algorithm such as the k-nearest neighbours, Naive Bayes and so on. To cluster the training data, there are a large number of clustering algorithms [30]. However, in this study, our proposed CluClas approach to finding representative instances is based particularly on a K-means clustering algorithm [6]. In particular, we have chosen a K-means algorithm for a number of reasons, including: (1) it is simple to implement, (2) it does not need to re-compute the centroid, (3) it has a limited number of parameter settings; (4) it can generate tighter clusters in comparison to hierarchical clustering, and (5) it is computationally faster than Expectation Maximization (EM) and hierarchical clustering algorithms, especially with a large number of variables. The K-means clustering algorithm partitions the traffic flows into k disjoint clusters (where k is a predefined parameter). Each cluster represents a region of similar instances based on the Euclidean distance between the instances and their centroids. Table II illustrates the process of selecting the most representative instances based on the concept of the K-means clustering algorithm. Particularly, in this paper, since a traffic data set can contain patterns from different classes, we adapted the concept of *homogeneous clustering* to partition and identify the instances belonging to each class separately. At the completion of the K-means cluster, we select the centroid of each cluster to represent all of the data in the corresponding cluster. Consequently, the number of training sets becomes much smaller than the original instances.

D. Learning process in the CluClas approach

At the completion of the clustering process, we utilize the centroid of each cluster to build a representative training data set to generate the classification model. We have chosen a classification model here instead of clustering as it has better accuracy [8]. Our approach to building a traffic classifier is based on the concept of the Hidden Markov Model (HMM) [19]. This is due to its powerful modeling; it is far more powerful than many statistical methods. Hidden Markov

TABLE II: The process of selecting the most representative instances.

Input:
Given the input data set D to be clustered.
specify the number of clusters k.
Building Cluster:
1. Select k random instances from the training set as an initial centroid.
2. For each training instance $X \in D$, do the following.
a. Compute the Euclidean distance as.

$\arg\min \sum_{\mathbf{i=1}}^{\mathbf{k}} \sum_{\mathbf{x_j} \in \mathbf{C_i}} \|\mathbf{x_j} - \mathbf{m_i}\|$.

b. Find and assign X to the closest cluster C.

$C_i^t = \{x_j : \|x_j - m_i^t\| \leq \|x_j - m_l^t\| \text{ for all } l = 1, \cdots, k\}$.

c. Update the centroid of C.

$m_i^{t+1} = \frac{1}{|C_i^t|} \sum_{x_j \in C_i^t} x_j$.

3. Repeat step 2 until the centroids of clusters stabilize based on the mean square error.
4. Return the centroid of each cluster as a representative instance.

Model (HMM) is one of the most popular statistical methods widely applied in pattern recognition [15], [4]. This is due to its good capability to grasp temporal statistical properties of stochastic processes. The basic idea of the HMM process is to construct an optimal model which can explain in a time sequence the occurrence of observations, which can be then used to identify other observation sequences. In particular, for modelling the data distribution, our approach is based on a finite mixture model for the probability of the cluster labels. The main assumption is that the traffic applications y_i are modelled as random variables drawn from a probability distribution described as a hidden Markov model:

$$p(x) = \sum_{i=1}^{M} p(x|\theta_i) p_i \tag{5}$$

where x denotes the observation, p_i denotes the weight of the i-th model, and $p(x|\theta_i)$ is the density function for the observation x given the component model p_i with the parameters θ_i Here we assume that the models $p(x|\theta_i)$ are HMMs; thus, the observation density parameters θ_i for the i-th component are the transition matrices. $p(x|\theta_i)$ can be computed via the forward part of the forward backward procedure. By applying the Hidden Markov Model (HMM) on the representative training sets, it would create a set of hidden states Q and a state transition probability matrix A which includes the probability of moving from one hidden state to another. In general, there are at most M^2 transitions among the hidden states, where M denotes the number of states.

E. Classification/Prediction process in CluClas approach

Fig. 1 illustrates also the process of classification phase. Given a new statistical flow instance of TCP, the log-likelihood for this observation is calculated based on all generated Markov models M^k, with $\prod^k = (\pi_1^k, \cdots, \pi_n^k)$ and $A^K = \left\{a_{\sigma_i,\sigma_j}^k\right\}$ as:

$$\log Pr(O|M^{(k)}) = \log\left(\pi_k^{o_1} + \sum_{i=1}^{n} \log a_{o_i,o_i+1}^k\right) \tag{6}$$

Hence, each flow will be assigned to its application type for which the log-likelihood is the largest.

IV. Experimental evaluation

The aim of this section is to comprehensively evaluate the proposed CluClas approach by performing a large number of experiments. In this section, we present our experimental evaluation in four parts. In Section IV-A, we describe the four traffic data sets. In Sections IV-B and IV-C, we present the evaluation metric and discuss the experimental setting. In Section IV-D, we present the results of the CluClas approach and compare it to individual K-means and HMM classification.

A. Experimental setting

To get robust results, we have repeated the experiments ten times with the same set of parameters. For these experiments, we have set the value of k for K-means clustering algorithm to 400, since the large value of k can result in better performance. For the CluClas approach, we used K-means clustering to partition instances of flow from the same class into k clusters and the centroid of each cluster was then selected as training instances for HMM model to build an accurate classification model. For the traditional HMM approach, we built each model individual application type and then used the built model to classify the testing traffic data.

To test the effectiveness of each approach, the data set was divided into two sets. In particular, 75% of the data was selected randomly as training set, and the remaining data was considered as testing set. All experiments were performed on a 64-bit Windows-based system with 4-duo and Core(i7), 3.30 GHz Intel CPU with 8-GB of memory. For the implementation, we used Java and integrated it with Weka software [29].

B. Traffic data sets

TABLE III: Summery of data sets used in the experiments

Data sets	Features	Classes	Training instances	Testing instances
ITD	149	12	15750	5250
DARPA	41	2	7500	2500
wide	20	6	10030	3100
isp	20	14	10500	3500

To evaluate the performance of the CluClas approach, we conducted the experiments using four publicly available traffic data sets. In what follows, we will discuss the characteristics of the four traffic data sets.

- *Internet Traffic Data (ITD)*: the traffic data sets collected by the high-performance network monitor (described in [20]) are some of the largest publicly available network traffic traces used in our experiment. These data sets are based on traces captured using its loss-limited, full-payload capture to disk, where timestamps with a resolution of better than 35 nanoseconds are provided. The data were taken for several different time periods

from one site on the Internet. This site is a research-facility which hosts up to 1,000 users connected to the Internet via a full-duplex Gigabyte Ethernet link.

- *DARPA* data sets: Since 1999, the DARPA99 data have been the most widely used data set for IDS evaluations that use machine learning techniques. This data set was prepared by Stolfo et al [25] and is built based on the data captured in the *DARPA* cup99 IDS evaluation program. This data set contains raw traffic flow records, each with an associated label to indicate whether the record was labeled as either normal or an attack. In particular, the simulated attacks fall in one of the most common types of attacks, including: Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L) and a Probing Attack.
- *wide* data set [1]: This is a real network traffic data set randomly selected from 182 *wide traces*. This data set is annotated by using a deep packet inspection (DPI) tool and manual inspection to assist researchers in evaluating their intrusion detection systems (IDS) models. In particular, the flows in this data set are categorized into 6 type of traffic applications including HTTP (dominate application), P2P, DNS, FTP, CHAT, and MAIL.
- *isp* data set [26]: This annotated data set is obtained from *isp* traces. The *isp* traces is a full payload traffic data set collected from a medium-sized Australian *isp* network which hosts few hundred users and internal servers for web, mail and name services. The *isp* data set consists of 30k flows randomly sampled from 14 types of traffic applications, inclusing BT, DNS, eBuddy, FTP, HTTP, IMAP, MSN, POP3, RSP, SMTP, SSH, SSL, XMPP, and YahooMsg.

In general, we focused our study on TCP flows (as did most of the pervious works [12], [32]). This is due to the clear start-end information for TCP flows. Table III summarizes the number of features, the number of classes in the four data sets, and the proportion of training and testing instances.

C. Evaluation metrics

In this section, we investigate the performance of our proposed CluClas approach. To do so, we used well-known confusion metrics. These metrics include Classification Accuracy (CA), Precision (PR), Recall (RC) and F-measure. Below each metrics is explained:

TABLE IV: Standard Confusion Metrics for Evaluation of Attack Classification

Actual label of flows	**Predicted label of flows**	
	Normal	**Attack**
Normal	**True Negative (TN)**	**False Positive (FP)**
Attack	**False Negative (FN)**	**True Positive (TP)**

- **overall accuracy**: is the percentage of all normal and anomaly instances that are correctly classified, which is defined as follows in terms of the metrics defined in Table IV: CA is defined as

$$CA = \frac{TP + TN}{|\Omega|} \tag{7}$$

where Ω is the total number of instances in the data set.

- **Recall**: is the percentage of anomaly instances correctly detected, which is defined as follows in terms of the metrics defined in Table IV:

$$Recall = \frac{TP}{TP + FN}$$

- **Precision**: is the percentage of correctly detected anomaly instances over all the detected anomaly instances, which is defined as follows in terms of the metrics defined in Table IV:

$$Precision = \frac{TP}{TP + FP}$$

- **F-measure** is the equally-weighted (harmonic) mean of precision and recall, which is defined as follows:

$$F - measure = 2.\frac{Recall \times Precision}{Recall + Precision}$$

For the theoretical basis of F-measure, please refer to [24] for details.

D. Result and discussion

In the following subsection, we present the results of our CluClas approach and compare them against each individual approach.

1) Accuracy Performance: Here, we present the accuracy of the K-means, HMM and CluClas approaches on *DARPA*, *isp*, *wide* and *ITD* data sets. Since the evaluation process of clustering algorithms is totally different from classification algorithms, we adapted the evaluation concept of the classification to evaluate the clustering outputs. To do so, we labeled the cluster based on the dominant application in each cluster. In particular, the labeling function (*LF*) assigns a class label to each cluster as:

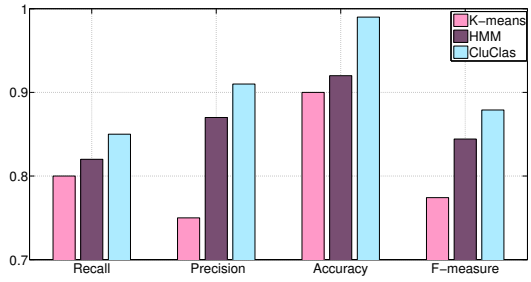
$$LF = \arg\max_{\mathrm{A_i} \in \mathrm{A}} \sum_{\mathbf{x_j} \in \mathbf{C}} \mathbf{\Psi}(\theta(\mathbf{x_j}), \mathbf{A_i}) \tag{8}$$

where A and C denote the actual application class label and the cluster respectively. $\Psi(\theta(x_j), A_i)$ returns the actual class label of a flow instance x, and it can be defined as follows:

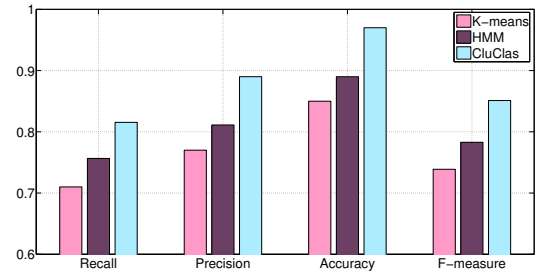
$$\Psi(\theta(x_j), A_i) = \begin{cases} 1 & \text{if } \theta(x_j) = A_i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

After labeling the cluster, we can use the standard confusion metrics to evaluate the quality of K-means clustering.

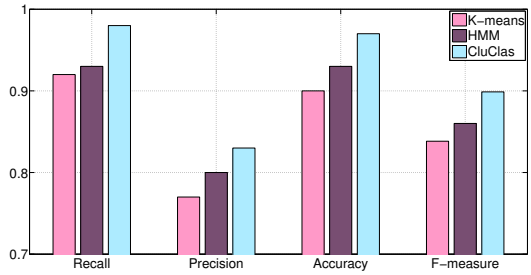
Figures 2a, 2b, 2c and 2d illustrate the accuracy value obtained by each individual approach. For the K-means and CluClas the k value was set to 400 (Fig. 3 justifies the setting of the k value). In general, we can observe that CluClas has better performance than the K-means and HMM approaches in terms of TPR, FPR, precision, accuracy and f-measure on all data sets. In particular, the average F-measure scores



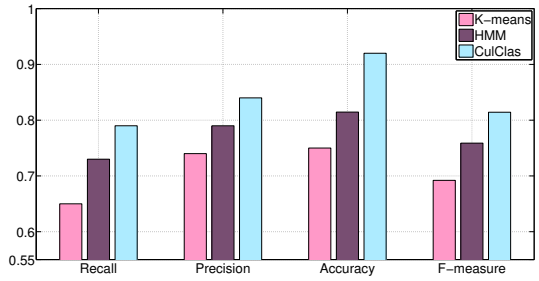
(a) Comparing the Accuracy Performance of K-means, HMM and CluClas Approaches on *DARPA* Data Set



(b) Comparing the Accuracy Performance of K-means, HMM and CluClas Approaches on *isp* Data Set



(c) Comparing the Accuracy Performance of K-means, HMM and CluClas Approaches on *wide* Data Set



(d) Comparing the Accuracy Performance of K-means, HMM and CluClas Approaches on *ITD* Data Set

Fig. 2: The Accuracy Performance of K-means, HMM, CluClas Approaches on Four Different Traffic Data Sets

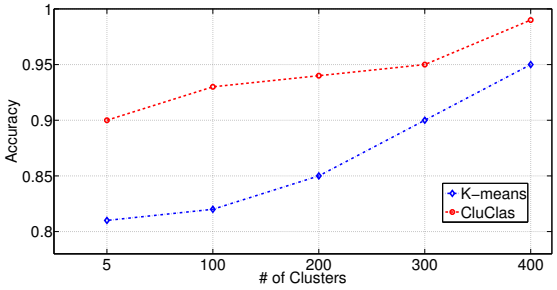
of CluClas are always higher than K-means and HMM by about 3.47-12.21 percent on all four traffic data sets, and the overall accuracy is always higher than K-means and HMM by about 7.19-17.48 percent on all four traffic data sets. This can be explained by the fact that the CluClas approach discards the irrelevant and redundant attributes, chooses informative sample flows which can represent the whole data fairly well, and combine the advantages of both K-means and HMM to efficiently build the classification model. It can be seen from Figures 2a, 2b, 2c and 2d that the performance of the HMM approach outperforms the K-means on all four data sets by about 2.01-6.45 percent and 2.17-7.06 percent with respect to the overall accuracy and F-measure values.

Figures 3a, 3b, 3c and 3d show the overall accuracy of K-means and CluClas approaches with respect to the number of clusters, k. It can be seen that the performance of both K-means and CluClas keet improving as the number of clusters increased on all four data sets. For example, on the *isp* data set, it can be seen as the number of cluster gradually increased from 5 to 400, the average accuracy of CluClas also keeps improving from 70.011 to 97.016 percent and from 51.12 to 85.01 percent for K-means clustering. This can be explained by the fact that setting the number of clusters to a low value would underestimate the natural grouping within the traffic data and thus force samples from different applications to be a part of the same cluster.

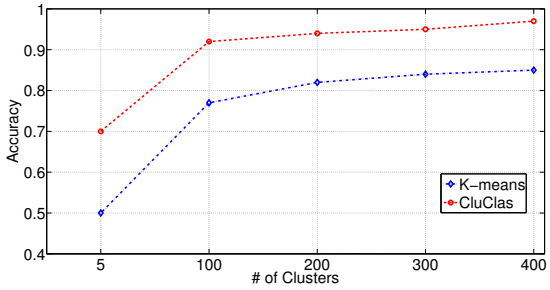
2) Runtime Performance: Another key motivation of the CluClas approach is to improve the runtime performance of network classification. Thus, in this section, we compare the runtime performance of the CluClas approach against the each individual approach. For each approach, the test was repeated ten times to give the average execution time and to have greater confidence in the obtained results.

Fig. 4a shows the normalized training time for the K-means, CluClas and HMM approaches on all four data sets. This is particularly important because the model building phase is computationally time consuming. Note the value of 1 represents the slowest training time. It can be seen from Fig. 4b that K-means has the fastest training time in comparison to both the CluClas and HMM approaches. In particular, the average building time for the K-means approach is only 30.12 percent of the building time of CluClas and 8.34 percent of the runtime of HMM. Also, it can be seen from Fig. 4b that the CluClas approach achieved the second fastest training time in comparison to HMM, with an average of 27.65 percent. A promising future research direction would be to reduce the execution time of these three approaches by using parallel computing, such as multi-core CPUs or Graphics Processing Units (GPU).

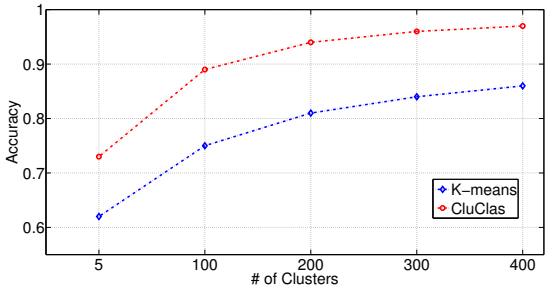
Fig. 4b compares the normalized classification time of the K-means, CluClas and HMM approaches on all four data sets. This is particularly important when considering real-time classification of potentially thousands of simultaneous network flows. Note the value of 1 represents the slowest classification time. From Fig. 4b, it can be seen that the K-means algorithm has the best classification time, while HMM has the worst clas-



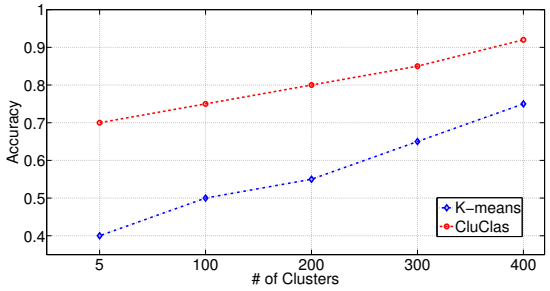
(a) The Influence of k Value Variation on Accuracy of K-means and CluClas Approaches on *DARPA* Data Set



(b) The Influence of k Value Variation on Accuracy of K-means and CluClas Approaches on *isp* Data Set

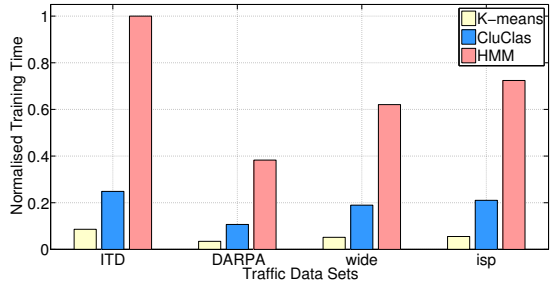


(c) The Influence of k Value Variation on Accuracy of K-means and CluClas Approaches on *wide* Data Set

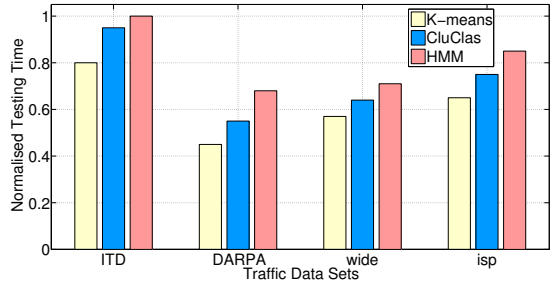


(d) The Influence of k Value Variation on Accuracy of K-means and CluClas Approaches on *ITD* Data Set

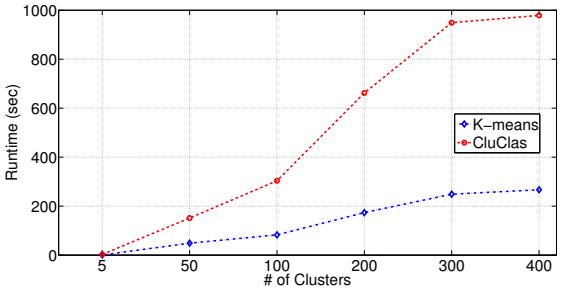
Fig. 3: The Influence of k Value Variation on Accuracy of K-means, HMM, CluClas Approaches on Four Different Traffic Data Sets



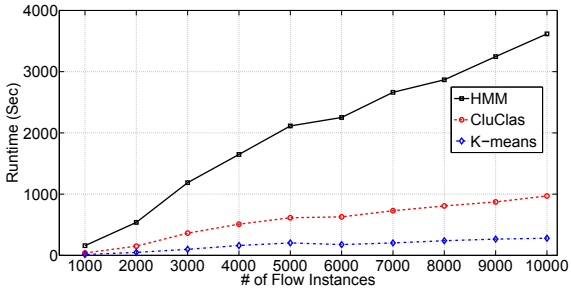
(a) Comparing the Building Model Time of K-means, HMM and CluClas Approaches on all Traffic Data Sets



(b) Comparing the Classification Time of K-means, HMM and CluClas Approaches on all Traffic Data Sets



(c) Runtime Performance of K-means and CluClas Approaches With respect to the different K value on *wide* Data Set



(d) Comparing the Scalability of K-means, HMM and CluClas Approaches on *wide* Data Set

Fig. 4: The Runtime and Scalability Performance of K-means, HMM, CluClas Approaches on Four Different Traffic Data Sets

sification time. Generally, we observed marginal differences between these three approaches in terms of classification time. To make the result more precise, the average classification time of K-means was about 85.46 percent of CluClas and 76.23 percent of HMM. On the other hand, it is notable that the average classification time of CluClas was only 89.19 percent of HMM.

Fig. 4c analyses the runtime the behaviour of CluClas and the K-means approaches when the k value was varied. Note number of clusters, k, varies from 5 to 400. Not surprisingly, it can be observed from Fig. 4c that the computational time of both approaches is affected as the k value increases. However, the runtime of K-means generally is better than CluClas as the value of k increases.

In this section, we also examine the scalability of our proposed CluClas approach against individual approaches, including K-means and HMM. With ever-looming page limitations, it was felt sufficient to evaluate the scalability of the CluClas and the other two approaches only on the *wide* data set. For the scalability analysis, the performance of each approach was evaluated with traffic samples varying from approximately 1000 to 10000 traffic samples (the size of the samples in the training data set is limited by the amount of memory since these approaches need to load the entire training data into memory before building the model). As can be seen from Fig. 4d, that K-means scales better than CluClas and HMM respectively. On the other hand, CluClas obtains better scalability results than HMM; this can be explained by the fact that our CluClas worked only on small representative samples, while the traditional HMM was required to process all of the instances.

V. Conclusion

In this paper, we developed a CluClas approach for network traffic classification. First, the traffic data was preprocessed by applying a new feature selection method on the training data to identify informative attributes and remove irrelevant and redundant attributes. Second, representative instances from the training data set are selected to improve the accuracy and efficiency of the learning capability. In particular, we select these representative instances by applying a K-means clustering algorithm to partition the training instances into k disjoint clusters and then select only the centroid of each cluster. Third, the hidden Markov Model (HMM) is built on the representative training data to improve overall classification accuracy. We compared our approach with the individual K-means and HMM in terms of classification accuracy and runtime performance over four traffic data sets. The experimental results show that our CluClas approach achieved higher classification accuracy compared to K-means and HMM. On the other hand, while the CluClas approach has improved the accuracy and runtime of network classification, future work devoted to improve the scalability of ClaClus approach by using (i) the GPU environment and/or (ii) parallel computing.

A future direction of this research is evaluating our approach over different types of clustering and classification methods. Developing a theoretical proof also is left for future work.

References

[1] Mawi working group traffic archive. [ONLINE]. 2009. Available:http://mawi.wide.ad.jp/mawi/.

[2] T. AULD, A. MOORE, and S. GULL. Bayesian neural networks for internet traffic classification. *IEEE transactions on neural networks*, 18(1):223–239, 2007.

[3] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.

[4] D. Bouchaffra. Conformation-based hidden markov models: Application to human face identification. *Neural Networks, IEEE Transactions on*, 21(4):595–608, 2010.

[5] T. Chou, K. Yen, and J. Luo. Network intrusion detection design using feature selection of soft computing paradigms. *International Journal of computational intelligence*, 4(3):196–208, 2008.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

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

[8] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286. ACM, 2006.

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

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

[11] A. Fahad, Z. Tari, I. Khalil, A. Almalawi, and A. Y. Zomaya. An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion. *Future Generation Computer Systems*, 36:156–169, 2014.

[12] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri. Toward an efficient and scalable feature selection approach for internet traffic classification. *Computer Networks*, 57(9):2040–2057, 2013.

[13] T. Karagiannis, A. Broido, M. Faloutsos, et al. Transport layer identification of p2p traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134. ACM, 2004.

[14] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: multilevel traffic classification in the dark. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 229–240. ACM, 2005.

[15] L. Khan, M. Awad, and B. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB JournalThe International Journal on Very Large Data Bases*, 16(4):507–521, 2007.

[16] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[17] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. In *Passive and Active Network Measurement*, pages 205–214. Springer, 2004.

[18] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. In *Passive and Active Network Measurement*, pages 205–214. Springer, 2004.

[19] D. R. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221. ACM, 1999.

[20] A. Moore, J. Hall, C. Kreibich, E. Harris, and I. Pratt. Architecture of a network monitor. In *Passive & Active Measurement Workshop 2003 (PAM2003)*. Citeseer, 2003.

[21] A. Moore and K. Papagiannaki. Toward the accurate identification of network applications. *Passive and Active Network Measurement*, pages 41–54, 2005.

[22] T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *Communications Surveys & Tutorials, IEEE*, 10(4):56–76, 2008.

[23] Y. Reich and S. J. Fenves. The formation and use of abstract concepts in design. In *Concept formation: knowledge and experience in unsupervised learning*. Citeseer, 1991.

[24] W. M. Shaw Jr. On the foundation of evaluation. *Journal of the American Society for Information Science*, 37(5):346–348, 1986.

[25] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan. Cost-based modeling for fraud and intrusion detection: Results from the jam project. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, volume 2, pages 130–144. IEEE, 2000.

[26] Y. Wang, Y. Xiang, J. Zhang, and S. Yu. A novel semi-supervised approach for network traffic clustering. In *Network and System Security (NSS), 2011 5th International Conference on*, pages 169–175. IEEE, 2011.

[27] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. T. Yang. Internet traffic classification using constrained clustering. *IEEE Transactions on Parallel and Distributed Systems*, page 1, 2013.

[28] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16, 2006.

[29] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham. Weka: Practical machine learning tools and techniques with java implementations. 1999.

[30] R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

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

[32] J. Zhang, C. Chen, Y. Xiang, and W. Zhou. Internet traffic classification by aggregating correlated naive bayes predictions. *Information Forensics and Security, IEEE Transactions on*, 8(1):5–15, 2013.