Incorporating Multiple Cluster Models for Network Traffic Classification

Anil Kumar, Jinoh Kim, Sang C. Suh
Department of Computer Science
Texas A&M University, Commerce, TX
Email:akatta1@leomail.tamuc.edu,
jinoh.kim@tamuc.edu, sang.suh@tamuc.edu

Ganho Choi
Sysmate Inc.
1290 Dunsan-Dong Seo-Gu, Deajeon, 302-830, Korea
Email: ghchoi@sysmate.com

***Abstract*—Network traffic classification is one of essential functions for local and ISP networks for quality of service, network usage statistics, resource provisioning, and security. With its importance, a substantial number of previous studies have explored various machine learning techniques based on network flow statistics to improve the accuracy of classification and reported promising results with fairly high classification accuracy. However, what we observed from previously proposed network traffic classification techniques with our own data set recently collected are somewhat unacceptable results. In particular, we observed that simply combining flow attributes for classification may lead to unexpectedly poor accuracy in classification (less than xx%). In this paper, we propose a new traffic classification method based on attribute groups, each of which consists of a set of attributes belonging to a single parameter (e.g., packet size, inter-arrival time, etc). Our method then incorporates multiple cluster models obtained from individual attribute groups to reach the final classification decision based on the population of candidate protocols (or applications). From our extensive experiments, we observed that our proposed technique significantly outperforms existing cluster-based classification techniques, showing up to yy% better accuracy.**

I. Introduction

Accurately identifying network-based applications is of major interest for local and ISP networks for various purposes, including quality of service, network usage statistics, resource provisioning, and security [1, 5, 11]. Earlier, network protocols/applications were identified simply based on TCP/UDP port numbers. However, the traditional technique based on port-numbers are proved to be ineffective where accuracy is less than 70% [6] since network applications using random port numbers or non-standard port numbers are increasing day-by-day and also usage of tunneling makes identification of applications more difficult just based on port numbers. Due to this reason, a substantial body of research has been conducted to replace or complement the port-based identification.

One approach to overcome the limitation of the port-based identification is to inspect the packet payload information with template signature sets [12, 9] or machine learning techniques [6]. While highly accurate, drawbacks of the deep packet inspection-based approach include encrypted traffic transformed with cryptographic keys and privacy concerns as many countries do not permit the extraction of full payload information from packets with increasing privacy requirements.

The limitations of the traditional port-based identification and the payload inspection-based classification suggested to utilize transport layer characteristics of the application as the differentiator. From previously proposed techniques [], we can see the combination of transport layer characteristics with machine learning techniques would be an effective alternative for network traffic classification. Several techniques were proposed based on supervised learning [], while some other techniques utilized unsupervised or semi-supervised clustering techniques [], reporting promising accuracy for network traffic classification. However, what we actually observed from previously proposed network traffic classification techniques with our own data set recently collected are somewhat unacceptable results. In particular, one important observation is that simply combining flow attributes for classification leads to unexpectedly poor accuracy in classification (less than xx%), which motivates us to thoroughly examine the impact of flow attributes in this work.[1]

To evaluate the significance of flow attributes to classification accuracy, we use a notion of attribute groups. An attribute group consists of a set of attributes that are derived from a single communication characteristic. For example, the packet size group includes the minimum packet size, maximum packet size, average packet size, standard deviation of packet sizes observed from a single flow. We consider four attribute groups of flow information group, packet size group, packet inter-arrival time group, and relative packet inter-arrival time group, as will be discussed in the Section III in detail. From our preliminary experiments, we observed that some combinations of attributes work quite better than the other combinations. Moreover, simply applying a subset of attributes selected based on the evaluation to supervised techniques significantly improves performance compared to using the entire attributes without selection. With the initial observations, we developed a new semi-supervised learning technique based on the attribute groups. A key challenge for this approach is how to use multiple attribute groups to make a single classification decision. To address this, we establish independent cluster models based on individual attribute groups and incorporate

[1] A (network) flow is defined as a set of packets for a single session of communication with the five tuples of source IP address, destination IP address, source port number, destination port number, and protocol type.

the results collected from multiple cluster models. Although it is known that clustering techniques generally work poorly compared to supervised learning techniques [11], we will present that the proposed technique using multiple cluster models yield comparable classification accuracy.

The key contributions of this paper can be summarized as follows:

- We evaluate the significance of flow attributes to classification accuracy with a notion of attribute group. We used 18 flow attributes in total and four groups are formed, which are flow information group, packet size group, packet inter-arrival time group, and relative packet inter-arrival time group.
- From the evaluation results with the attribute groups, we examine classification accuracy with the entire attributes and with the selected ones using supervised learning methods, to ensure validity of the selection.
- We present a new clustering technique for network traffic classification that utilizes multiple cluster models developed from the attribute groups. A set of heuristic algorithms are also presented to incorporate multiple cluster models.
- We also present experimental results for evaluating the proposed traffic classification technique. Experiments for sensitivity study are also conducted to see the impact of configurable parameters.

The paper organization is as follows. We provide a summary of related studies in Section II. In Section III, we examine the significance of attribute groups to classification accuracy and performance with selected attributes with supervised learning methods. We then present the new clustering technique incorporating multiple cluster models in Section IV and evaluation results are presented in Section V. Finally we conclude our presentation with a summary and future direction in Section VI.

II. Related Work

A substantial body of research has been conducted for network traffic classification with machine learning techniques. The work can broadly be divided into the following three categories:

- *Un-supervised* []: Labeling information of the training data is not available at the time of training. We use various clustering algorithms for the classification of unlabeled data[].

- *Supervised* []: We provide the labels for the flows when we train the model and then use this model to test each incoming flow whether it belongs to any of the application which is provided at the time of training[].

- *Semi-supervised* []: We provide partial labeling information at the time of training and we use clustering algorithms to cluster the training data. We use the partially available labeling information to label each cluster. Heuristics have been proposed on how to label the cluster from partial training information[]

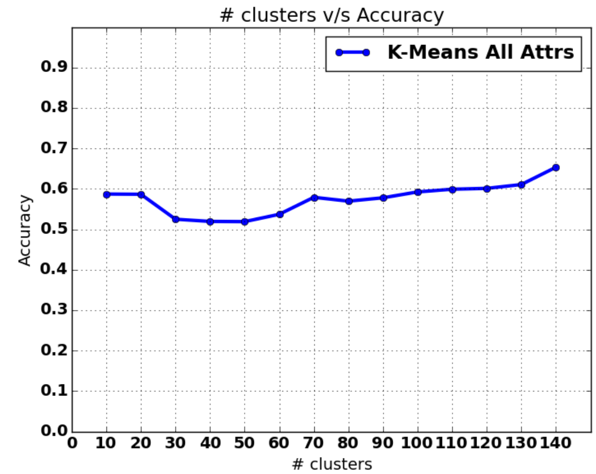

Fig. 1: Number of clusters v/s Accuracy for K-Means algorithm with All

ACAS[6]: ACAS encodes initial *n-bytes* of the payload into space and uses supervised machine learning algorithm as the classification algorithm. We observe this technique is effective in identifying applications with very high accuracy with our data set ($\approx$ 99%). Drawback of this technique is it requires completely labeled data. Which means we should know prior to the start classification it should know all the applications that classifier may encounter in future (which is not a feasible solution). It also As mentioned, however, it requires the access to the payload, which could be limited by laws due to privacy concerns. In addition, encryption plays an important role in classification, which may significantly lower the accuracy.

K-Means classification technique[3]. We ran the K-Means classification technique across by varying the number of clusters from 10 to 140, and Figure 1 show the result. As can be seen from the figure, classification performance is largely not acceptable with quite less than 70% accuracy.

Early Application Identification.[1] We also evaluated this technique. Rather than considering only the first 4 packets in a flow as suggested in the paper, we considered the whole packets to see the maximum performance. Although this technique is based on clustering as the above technique and uses only the average packet size attribute, it yielded quite enhanced results as shown in Figure 2 across diverse ratios between training data set and testing data set.

III. Grouping of Attributes

In this section, we present the impact of flow statistics attributes to classification accuracy based on grouping of attributes. We first describe the flow attributes used in this study and attribute groups, and then discuss classification accuracy for individual groups.

From the network traffic data set, totally 18 flow attributes are considered, as summarized in Table I. The information about the data set used in this study will be described in Section V in detail.

TABLE I: Attributes used in out studies

Attribute	Description
Flow size	Total number of bytes in the flow
Flow duration	Duration of the flow from initiation to termination
Number of Packets	Total number of packets inside the flow
Number of packets per second	Number of packets / Flow duration
Number of bytes Per second	Total packet size / Flow duration
Payload size	the payload size in bytes without network and transport layer headers
Packet size	min/max packet size and average and standard deviation of packet sizes in the flow
Packet inter-arrival time (IAT)	min/max packet inter-arrival time and average and standard deviation of inter-arrival times in the flow
Relative IAT (RIAT)	Recomputation of IAT after dividing IAT by round-trip delay: max RIAT and average and standard deviation of RIATs

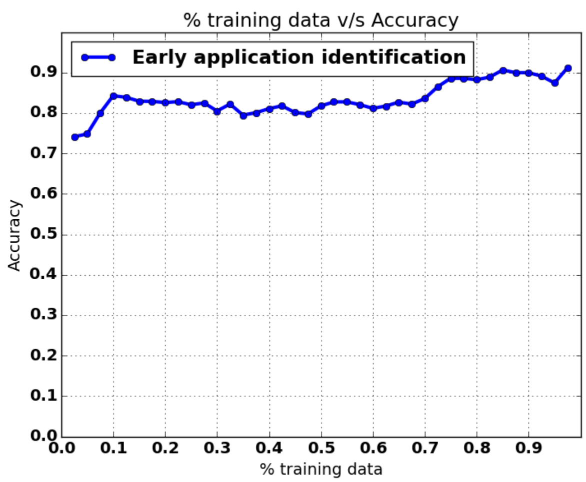

Fig. 2: Percentage of Training Data v/s Accuracy for Early Application Identification

As discussed in the previous section, using only a single attribute of average packet size works much better than using a greater number of attributes for traffic classification. Thus, it would be good to see the significance of the attributes. Since there can exist too many combinations of the attributes, we categorized the attributes into four groups based on their characteristics, as follows:

- *Flow group*: Any flow-level attribute belongs to this group, including flow size, flow duration, and so forth.
- *Packet size group*: Attributes related to packet size belong to this group: *minimum*, *maximum*, *average*, and *standard deviation* packet size
- *Inter-Arrival Time (IAT) group*: Attributes from inter-arrival time: *minimum*, *maximum*, *average*, and *standard deviation* inter-arrival time
- *Relative Inter-Arrival Time (RIAT) group*: Attributes related to relative inter-arrival time: *maximum*, *average*, and *standard deviation*. Relative inter-arrival time is defined as,

$$\tau_i' = \frac{\tau_i}{\min\limits_{k=1..|f|-1} \tau_k}$$

Here, f is a flow with $|f|$ number of packets and τ_i is the time difference between i-th packet and $(i+1)$-th packet in that flow.

TABLE II: Accuracy of flow group

Flow Size	Flow Duration	Number of Packets	Avg Pkts Per Sec	Avg Bytes Per Sec	Payload Size	Accuracy
✓	✓	×	×	×	×	92.92%
✓	×	✓	×	×	×	91.66%
✓	×	×	✓	×	×	91.12%
✓	×	×	×	×	✓	94.53%
×	✓	×	×	×	✓	92.58%
×	×	✓	×	×	✓	93.80%
✓	✓	×	✓	×	×	92.31%
×	✓	✓	×	×	✓	92.02%
✓	✓	✓	✓	×	×	91.86%

To evaluate the attributes' performance impact to network classification, we applied the clustering technique introduced in the previous section with the individual subsets of attributes for each group. To measure classification accuracy, we used a simple metric of true positive, and thus, accuracy indicates true positive rate in this paper. We used our own data set described in the evaluation section in detail (Section V). By default, we used 90%:10% for the ratio of the number of training flows to the number of test flows.

Table II shows accuracy of classification when using a subset of attributes in the flow group. In fact, there can exist many number of combinations with the six attributes in the flow group, but we report the top 10 results only to save the space. Surprisingly, we can see very high accuracy up to 94.5% with only 2–4 attributes in the flow group.

We next examined the attributes in the packet size group. As shown in Table III, any combination of attributes in the packet

TABLE III: Accuracy of packet size group

Avg Pkt Size	Min Pkt Size	Max Pkt Size	Stddev Pkt Size	Final Accuracy
✓	×	×	×	92.43%
×	✓	×	×	75.95%
×	×	✓	×	92.22%
×	×	×	✓	91.82%
✓	✓	×	×	91.46%
✓	×	✓	×	92.98%
✓	×	×	✓	93.96%
×	✓	✓	×	91.46%
×	✓	×	✓	92.69%
×	×	✓	✓	92.76%
✓	✓	✓	×	93.98%
✓	✓	×	✓	93.62%
✓	×	✓	✓	92.76%
×	✓	✓	✓	91.73%
✓	✓	✓	✓	93.75%

TABLE IV: Accuracy of IAT group

Avg IAT	Min IAT	Max IAT	Stddev IAT	Final Accuracy
✓	×	×	×	47.29%
×	✓	×	×	49.34%
×	×	✓	×	39.20%
×	×	×	✓	42.82%
✓	✓	×	×	45.86%
✓	×	✓	×	44.25%
✓	×	×	✓	47.78%
×	✓	✓	×	39.64%
×	✓	×	✓	42.91%
×	×	✓	✓	40.54%
✓	✓	✓	×	45.20%
✓	✓	×	✓	47.18%
✓	×	✓	✓	42.70%
×	✓	✓	✓	40.10%
✓	✓	✓	✓	42.75%

TABLE V: Accuracy of RIAT group

Avg RIAT	Max RIAT	Stddev RIAT	Final Accuracy
✓	×	×	55.86%
×	✓	×	54.08%
×	×	✓	53.41%
✓	✓	×	55.21%
✓	×	✓	54.28%
×	✓	✓	55.25%
✓	✓	✓	56.41%

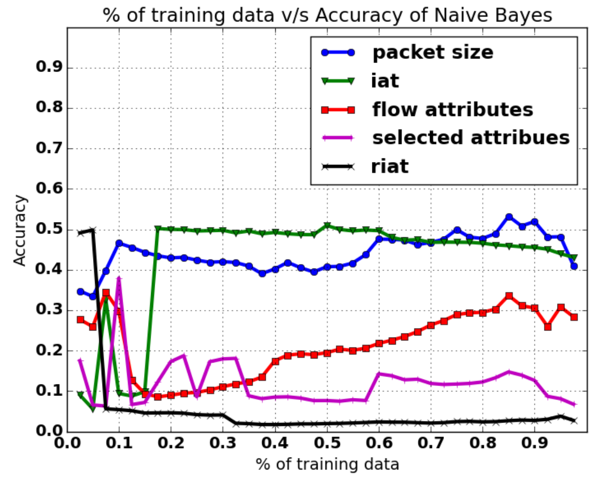

Fig. 3: % of training data v/s Accuracy for Naive Bayes with Groups of Attributes. Plotting Naive Bayes with each group of attributes.

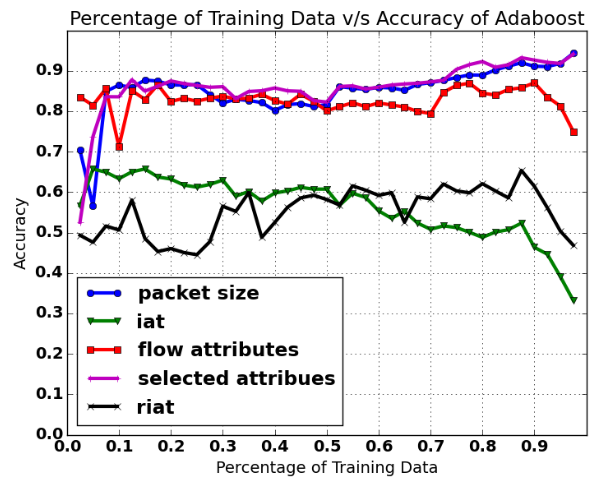

Fig. 4: Percent of training data v/s Accuracy for Adaboost with Groups of Attributes.Plotting Adaboost with each group of attributes.

size group works very good, showing up to 94% accuracy. When using minimum packet size only, it does not work well with 76% accuracy. The results here explain why it worked well only with the attribute of average packet size in [1]. We observed the combination of *Avg Pkt Size* and *Stddev Pkt Size* gives the best accuracy of the group, and the combination of *Min Pkt Size*, *Max Pkt Size* and *Avg Pkt Size* gives the second best accuracy in this group.

Unlike the flow group and the packet size group, we observed that the attributes related to packet inter-arrival time are not good with clearly low classification accuracy less than 60%, and hence, less attractive to use them for network traffic classification than the attributes in the flow and packet groups.

We also run supervised learning methods (Adaboost and

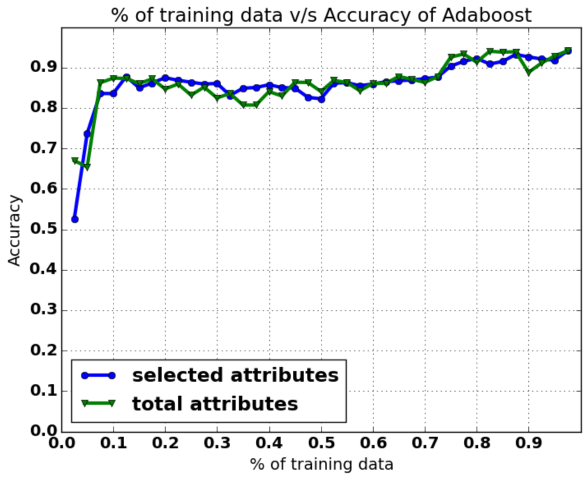

Fig. 5: Percentage of Training data v/s Accuracy for Adaboost

Naive Bayes) with a subset of the attributes to see their impacts. For the experiments with the supervised learning techniques, we considered the following attribute sets:

- *Packet Size* This group comprised of all the packet size related attributes like *average packet Size, minimum packet size, maximum packet size and standard deviation of packet size*.
- *IAT* This group consists of all the attributes related to IAT like *average IAT, minimum IAT, maximum IAT, standard deviation of IAT*
- *Flow Attributes* This group is composed of all flow level attributes like *flow duration, flow size, payload size, average bytes per second, average packets per second, etc*
- *Selected Attributes* This group consists of selected attributes which are resulting in better accuracy than other attributes. Consists of *average packet size, minimum packet size, maximum packet size, standard deviation of packet size, total packet size, payload size and number of packets*.
- *RIAT* This group composed of all the attributes related to relative IAT like *average RIAT, minimum RIAT, maximum RIAT and standard deviation of RIAT*

Figure 3 and Figure 4 show the classification results with Naive Bayes and Adaboost [4], respectively. We observed Naive Bayes works very poorly showing less than 60% accuracy. Unlike this, Adaboost showed over 85% accuracy for some combinations of attributes with 10% or more flows in the training data set (i.e., the ratio of training and testing data sets $\geq$ 10%). For Adaboost, we observed that the two groups of attributes showed outstanding performance compared to the others: packet size and selected attributes.

Figure 5 compares Adaboost performance when using the entire data set and using the selected attributes only. From the figure, we observed that using the selected attributes (7 out the 18 attributes) gives us almost the same accuracy(at some points even more), as when we consider the entire 18 attributes.

In sum, we observed that applying the entire attributes degrades classification accuracy when using cluster-based techniques. For supervised learning techniques, using the entire attributes does not degrade classification accuracy but using only half of the attributes showed the comparable result with the Adaboost classification technique.

IV. Multiple Cluster Models

A. Population Fraction

In this paper, we introduced new term called population fraction, which stands for percentage of dominant application (application with most number of flows in the cluster in consideration.) flows to total flows in a cluster

$$P_{clus} = (\frac{flows_{dominant}}{flows_{total}})_{clus} \tag{1}$$

B. Description of the technique used for classification

Based on II and III we can observe specific combination of attributes results in better accuracy(like *Average Pkt Size* and *Std. deviation Pkt Size*). From fig.1 it can be observed that the accuracy of selected attributes is around 90%, which can be further improved by considering multiple trained models(with different combination of attributes) than with one trained model(which has superset of combination of attributes used in training of multiple models). In a single classifier, supervised or unsupervised, we cannot consider combination of attributes as we have to put every attribute to be considered through one training model(one training model can have one combination of attributes. For example we can have one model based on *packet size* group and *iat* group combined together, but we cannot have *iat* and *packet size* group separately in one training model). So, we advise usage of four different models, each with specific combination of attributes(one based on *payload size, total packet size* other *payload size, number of packets*, etc.). After defining our four models. We have following:

- *Model 1* In this model we considered attributes, Average Pkt Size and Standard Pkt Size.
- *Model 2* We considered Average, Minimum, Maximum Packet size attributes in this model.
- *Model 3* Considered Total Flow Size, Payload Size as the attributes for this model
- *Model 4* Considered Number of Packets, Payload Size as the attributes for this model.

We use results from four models in determining the final classification result of the incoming flow. We tested different strategies in deciding the final result from four intermediate result. Following are the descriptions of the strategies used. *Strategies in deciding final result based on intermediate results*

- *Random* Select classification of any of the results from the four models as the final classification result
- *Greatest* Select classification result from model with highest population fraction as the final classification result

- *Quorum* Select the majority result from trained models. If we have 3 models resulted in one classification result and other resulted in other application then we consider result of 3 as the final classification result. In case of tie we select application randomly
- *Unanimous* Select the final classification only when all the results from the four models exactly results in the same result. Otherwise mark it as unknown
- *Unanimous Greatest* Similar to unanimous but fallback hypothesis is greatest.
- *Unanimous Random* Similar to unanimous but fallback hypothesis is random
- *Unanimous Quorum* Unanimous with fallback hypothesis as Quorum

V. Evaluation

A. Datasets

Data had been collected with full payload in early-2014. It has been gathered on various interfaces 1) Wired 2) WIFI 3) 3G and LTE. Data had been collected for individual application in isolation, by generating requests intentionally and capturing bidirectional data. Considered five tuples (i.e., Source IP, destination IP, source port, destination port and protocol). Used TCP flags to mark the start and end of flow(flow started before the capture, or flows terminating after the capture are not considered in construction of flows).

We selected 5 protocols (Skype, Bittorrent, Http, Edonkey and Gnutella) for further study, criteria for the selection of this protocols is number of flows.

B. Cleaning

We constructed flows from packets, which we read from *pcap* files using *scapy*[2] a python library. Used community maintained signatures[8] available for this protocols from L7 filters. We matched payload of each constructed flow with corresponding signatures from L7, if we find an unmatched flow then it is discarded otherwise labeled as the matched signature.

C. Experimental Setup

We used *sklearn*[10] a python library for machine learning algorithms in our study. *sklearn*[10] is the most used python library for data analysis in python. Used *Numpy*[7] for numerical computations, it is a python library to handle numerical computation in efficient way. Used *Matplotlib* for plotting the results, it is a python library which lets us plot the results.

We divide our data into training and testing partitions. We used training data to train the model and testing data to test the accuracy of the trained model in correctly classifying protocols. During training phase we label clusters in each model(we have 4 of them) based on population fraction(majority based). Each test flow is fed into the trained model and we get the label for that test flow which is compared with the actual label for that flow. Finally

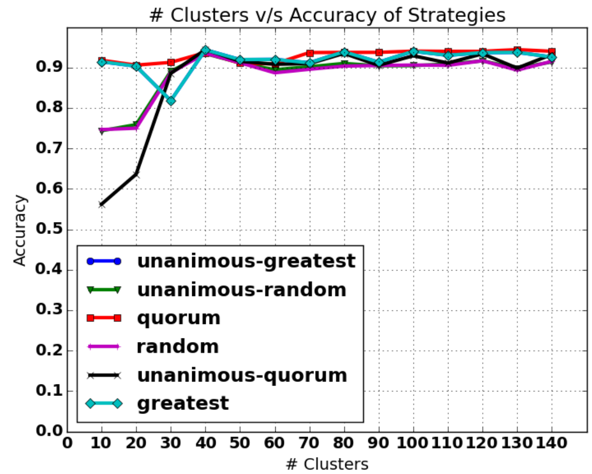

Fig. 6: # Clusters v/s Accuracy of various strategies

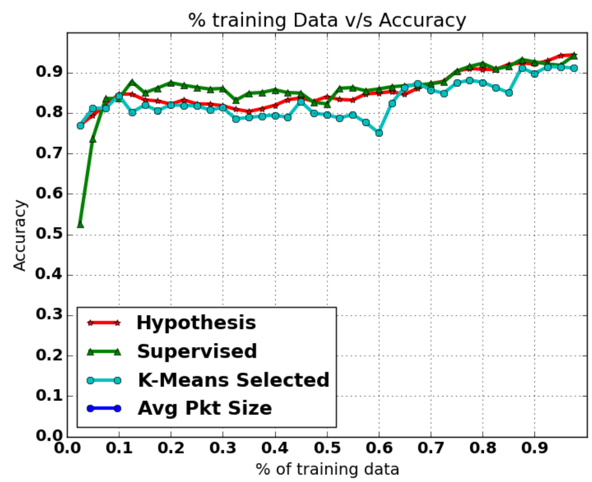

Fig. 7: Percent of training data v/s Accuracy Comparing all the existing and proposed technique

D. Results

From fig.6, it can noticed that accuracies are very high for *Quorum*, *Greatest* and *Unanimous Greatest* strategies.

We chose *Unanimous Greatest* for further study. Results for proposed classification technique, Supervised(Adaboost) and K-Means are used for comparison. From fig.7 we notice that the accuracy is higher than traditional K-Means clustering algorithm at almost all the ratios of the training set. Proposed technique accuracy is comparable to that of the supervised(Adaboost) technique, at times it is higher than supervised too.

VI. Conclusion

By considering all the attributes of flow in classifying the application doesn't give better results. Selected attributes with combinations gives better accuracy. Accuracy doesn't continually increases along with number of clusters, as we get better accuracy when we are around 40 clusters.

Considering IAT and RIAT for classification gives us the least accuracy when compared with other attributes of the flow. Even considering all attributes of packet size and flow also doesn't give better accuracy.

Population fraction will be very good parameter in classification of flow using clustering techniques.

ACKNOWLEDGMENT

This work was in part supported by the IT R&D program of MOTIE/KEIT [10041548, "240Gbps realtime automatic signature generation system for application traffic classification supporting over 95% completeness and accuracy"]

REFERENCES

[1] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proceedings of the 2006 ACM CoNEXT Conference*, ser. CoNEXT '06. New York, NY, USA: ACM, 2006, pp. 6:1–6:12. [Online]. Available: http://doi.acm.org/10.1145/1368436.1368445

[2] P. Biondi, "Scapy: Scapy is a packet manipulation tool for computer networks," 2011–. [Online]. Available: http://www.secdev.org/projects/scapy/

[3] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, ser. MineNet '06. New York, NY, USA: ACM, 2006, pp. 281–286. [Online]. Available: http://doi.acm.org/10.1145/1162678.1162679

[4] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[5] L. Grimaudo, M. Mellia, and E. Baralis, "Hierarchical learning for fine grained internet traffic classification," in *8th International Wireless Communications and Mobile Computing Conference, IWCMC 2012, Limassol, Cyprus, August 27-31, 2012*, 2012, pp. 463–468. [Online]. Available: http://dx.doi.org/10.1109/IWCMC.2012.6314248

[6] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "Acas: automated construction of application signatures," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, ser. MineNet '05. New York, NY, USA: ACM, 2005, pp. 197–202. [Online]. Available: http://doi.acm.org/10.1145/1080173.1080183

[7] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[8] E. S. Justin Levandoski, A. E. Matthew Strait, with support from Sebastian Celis, and L. Kittredge, "L7 signatures for various protocols." [Online]. Available: http://l7-filter.sourceforge.net/protocols

[9] B. Park, Y. J. Won, M. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubioquitous Networks and Services, NOMS 2008, 7-11 April 2008, Salvador, Bahia, Brazil*, 2008, pp. 160–167. [Online]. Available: http://dx.doi.org/10.1109/NOMS.2008.4575130

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci, "Subflow: Towards practical flow-level traffic classification," in *Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25-30, 2012*, 2012, pp. 2541–2545. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2012.6195649

[12] M. Ye, K. Xu, J. Wu, and H. Po, "Autosig-automatically generating signatures for applications." in *CIT (2)*. IEEE Computer Society, 2009, pp. 104–109.