

P2P Traffic Identification and Optimization Using Fuzzy C-means Clustering

Duo Liu

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada K1S 5B6
duoliu@sce.carleton.ca

Chung-Horng Lung

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada K1S 5B6
chlung@sce.carleton.ca

Abstract—Accurate identification of P2P traffic is critical for efficient network management and reasonable utilization of network resources, as P2P applications have been growing dramatically. Fuzzy clustering is more flexible than hard clustering and is practical for P2P traffic identification because of the natural treatment of data using fuzzy clustering. Fuzzy c-means clustering (FCM) is an iteratively optimal algorithm normally based on the least square method to partition data sets, which has high computational overhead. This paper proposes modifications to the objective function and the distance function that greatly reduces the computational complexity of FCM while keeping the clustering accurate. The proposed FCM clustering technology can be incorporated into a Fuzzy Inference System (FIS) to implement real-time network traffic classification by updating the training data set continuously and efficiently.

Keywords—network traffic identification; peer-to-peer communications; machine learning; fuzzy c-means clustering; statistical analysis; data transformation

I. INTRODUCTION

Network traffic that could be administered through traffic measurement, traffic control, and traffic simulation consists of flows from a great variety of users using different applications on diverse utilities. Network traffic management plays a key role in optimizing network performance, improving latency, and scheduling network resources for adjusting bandwidth based on traffic engineering. Network traffic measurement measures the amount of different types of traffic on a specific device, an individual port, or a particular application; network control prioritizes and optimizes traffic. Both network measurement and control are key processes for network traffic management [1].

Internet protocol-based network traffic classification (also known as IP traffic identification or characterization) has become one of the major challenges in network management. The main roles of network management are to provide information for capacity planning and provisioning, traffic engineering, fault diagnosis, application performance optimization, anomaly detection and pricing to Internet Service Providers (ISPs) and their equipment vendors. Based on network traffic measurement, appropriate features selection, and classification methods, traffic classification relies on an in-depth understanding of the contents of Internet traffic dynamically [2].

Peer-to-peer (P2P) networking is a distributed application architecture that mainly relies on the computing power and bandwidth of the participants in the network and uses application software to search for shared files of users connected to the network. Every participant in the network plays both roles of provider (server) and consumer (client) without requiring to be coordinated by a central server or file hosting services. Compared to Web browsing, e-mail or many other non-P2P applications, where data is only transferred in short intervals and with relative small quantities, P2P file-sharing could utilize significant network resources, resulting in network jams and affecting the quality of service (QoS). Classifying P2P traffic could support ISPs to increase overall availability and reliability of connection by offering less bandwidth to specific users or definite P2P applications while maintaining QoS to other users or applications. P2P traffic classification is a key element for network traffic management, network resource scheduling, and performance enhancement.

P2P traffic classification is getting more sophisticated as P2P communications are continuously changing from TCP layers using well known ports to disguising themselves by using port numbers for commonly used protocols such as HTTP and FTP and implementing dynamic port numbers. The accurate identification is proved to be more complicated with the enormous emerging P2P applications. Traditional P2P traffic classification mechanisms depend on mapping applications to well-known port numbers and have been very successful in the past. However, they are becoming obsolete as new P2P applications began using dynamic port numbers. Payload based analyses which inspect application packet payloads to determine if they contain characteristic signatures of known applications overcome the shortcomings of port-based classification. However, their performance is greatly reduced and inefficient with P2P applications using plain-text ciphers, variable-length padding, and encryption. The payload-independent statistical identification based on network layer and data link layer utilizes machine learning techniques such as neural networks, unsupervised machine learning, support vector machine, and hybrid mechanism with much flexibility, self-learning and self-adaptive and is gained much attention in academic and industrial areas [3].

Clustering is one of the machine learning techniques that has been used for network traffic classification. Fuzzy

clustering is more flexible than hard clustering and is practical for P2P traffic identification because of the natural treatment of data [2]. Traditional FCM (Fuzzy c-means) clustering technique has the advantage of quantifying a level of membership of elements to detected clusters. GK-FCM (Gustafson-Kessel FCM) can generate clusters that adapt themselves to the shape and location of data [4]. However, the computational complexity of FCM or GK-FCM is high. Our proposed modification for the objective function and distance can decrease computational complexity greatly while keeping the classification accurate, which is crucial for high volume real-time P2P traffic applications.

The remainder of the paper is structured as follows: Section II discusses various network traffic analysis categories. Different machine learning techniques with supervised, unsupervised, and hybrid learning are described in Section III. Section IV illustrates FCM and the proposed method for P2P traffic identification. Some simulations and comparison of simulating results are provided in Section V. We conclude the research and outline future work in Section VI.

II. NETWORK TRAFFIC ANALYSIS

Network traffic analysis is the process of observing and identifying network traffic. The ultimate goal of traffic analysis is to classify the traffic into definite groups, which is defined by network administrators. The classification could be based on users recognized by sources or destinations, or applications identified by a protocol. Traffic classification typically starts with network measurements and extracts traffic attributes later, and finally groups the traffic based on the selected strategies. The following sub-sections briefly describe various techniques that have been used for traffic analysis.

A. Traffic Measurement

Network traffic measurement measures traffic flows of observing devices such as routers, switches and dedicated probes to capture traffic attributes such as port numbers, payload and statistical characteristics. Traffic measurements are conducted on a continuous basis and the results are used for network management and temporary routing, maintenance, or long-term network deployment depending on the sampling period. They can be categorized into active and passive measurements. External traffic is injected to the network during the measurement in active measurements, whereas no enhanced injection is adopted for passive measurements. Active measurements are mainly used to detect the vulnerability of a specific network or a device, and examine the performance of a network to a particular application. Active measurements are not suitable to check the influence by users because user-independent packets are injected to network continuously. Passive measurements based on users connected to a specific network observe network connections, packets and flows to find source and destination addresses, port numbers and transport protocols. Passive measurements are the fundamentals for traffic identification and optimization [2, 5, 6]. This paper focuses on passive network flow-level-based measurements.

B. Port Number Analysis

Port-based analysis rests on the fact that many traditional P2P applications have default ports (assigned by the Internet Assigned Numbers Authority (IANA) and can be classified by looking up the TCP SYN packets for TCP traffic). This is the most straightforward and most successful method in the early time for identifying P2P network traffic. This method is easy and fast because it only accesses the header of the packets to get the data required for classification. The port-based classification system does not apply sophisticated strategy and can be easily extended by adding new application-port match pairs to its database. For example, HTTP traffic uses port 80, FTP uses 20, 21, DNS uses 53, and E-mail uses 25, 110, and etc.

However, for P2P applications, the accuracy of port-based classification is decreased abruptly as modern networks exceedingly carry more and more traffic that use dynamically allocated port numbers. Some applications may not have their ports registered with IANA or use ports other than well-known ports to prevent them from being detected. IP layer encryption may confuse the traffic headers also, making it impractical to identify network traffic through port-based approaches [7].

C. Payload-based Analysis

To avoid total reliance on the port numbers, a more reliable approach, payload-based analysis, a.k.a. DPI (Deep Packet Inspection), is to inspect the packet payload for characteristic signature of known applications. Payload-based identification based on some knowledge about the payload formats of specific applications checks particular features present in the application traffic to see whether it contains some characteristic signatures or not.

Payload-based classification works well for Internet traffic, including P2P traffic, when the payload is not encrypted. But payload-based classification is gradually becoming ineffective or impossible as new P2P applications use encryption, variable-length padding and other more advanced routing techniques to avoid being detected. The lack of openly reliable protocol specifications, especially for enormous emerging applications, the demand for high computing and storage capacities, the application signatures changing with the corresponding applications, and their protocol evolution are key challenges for developing application signatures [8, 9].

D. Statistical Analysis

To overcome the drawbacks of heavily relying on fixed port numbers or deep inspection of packet contents of network traffic, the approaches that take advantage of the statistical properties at the network layer and data link layer are attracting much attention to academic researchers, industrial developers and network administrators. The statistical analysis relies on the fact that traffic at the network layer has statistical properties (such as the distribution of flow duration, flow idle time, packet inter-arrival time and packet lengths, and etc.) and that similar applications typically have similar behavior patterns. For example, FTP normally used for large file transferring have a longer connection duration and larger average packet size, while some P2P applications such as BitTorrent typically are persistent and send data bi-directionally.

Exploiting distinct characteristics of traffic flow for specific applications, network traffic classifier can identify network traffic using only flow statistics. P2P nodes are possibly located within a domain of common data link layer switch; however, detecting the traffic is a minor concern because it does not consume network resources outside the local switch.

Based on the statistical parameters, machine learning which automatically builds a classifier by learning the inherent structure of network traffic dataset can automatically identify traffic flows. The classifier has the ability of continually gaining new knowledge based on the data newly acquired and modifies the model structure to adapt to the whole data and get the overall performance optimized [10, 11].

E. Feature selection and reduction

Feature selection techniques normally based on the actual performance of the learning algorithms are critical to build a machine learning classifier by which the network traffic can be identified using the smallest necessary set of features to reduce the demand for storage capacity and computational complexity. It can be broadly classified into filter method based on general characteristics of traffic data and wrapper method based on the comparison of different subsets and selection of the best one using machine learning techniques. Transformations of features such as Log or Gaussian are sometimes very effective to get higher levels of classification accuracy.

Appropriate feature reduction algorithms can reduce the number of features needed to identify traffic flows drastically, which in turn can greatly reduce the computational complexity without affecting classification accuracy apparently. Feature reduction can be achieved by using machine learning techniques to get the best combination of attributes that provide the most contrasting application classes. Although the total number of packets in the flow, the average packets size of a flow, and the duration of the flow are all valid features, there is still a need to transform some features to increase classification accuracy and reduce feature sets as much as possible using correlation-based or consistency-based approaches to decrease the computation overhead [12].

III. MACHINE LEARNING TECHNIQUES

Machine learning techniques can map instances of network traffic flows into different traffic classes based on selected features. Features are descriptive statistic variables that can be calculated from one or more packets. For example, mean packet length is a scientific discipline that focuses on the design and development of solutions. The discipline allows computers to evolve characteristics based on empirical data. As a subset of the wider Artificial Intelligence, it is closely related to probability theory, statistics, and fuzzy systems and neural networks.

The application of machine learning approaches usually starts by defining features based on calculation of multiple packets to identify and distinguish future unknown IP traffic. The next step is to train the established classifier to match traffic classes and feature sets. Finally, the last step is to classify unknown traffic using previously learned rules. Machine learning techniques can be categorized into three subsets: *supervised learning* which generates a function that

maps inputs to desired outputs, *unsupervised learning*, such as clustering which seeks the structure of data and organizes them orderly, and *hybrid* also called semi-supervised which combines supervised and unsupervised machine learning algorithms. The following sub-sections describe the three subsets briefly.

A. Supervised machine learning

In network traffic classification, supervised machine learning generates knowledge structures to classify new traffic flow into pre-defined classes based on the training of the classifier for the matches of known traffic to definite classes. The classifier is provided with a collection of traffic datasets and their pre-identified classes. The result of classifying the known traffic into specific classes becomes more accurate with the progress of training and the ultimate goal for the classifier is to classify the unknown traffic into the suitable classes as accurately as possible. Two major steps in supervised machine learning are matching training datasets with classes and classifying unknown new captured datasets into classes. Artificial neural network and Naïve Bayes classifier are two important algorithms used in traffic classification.

Based on independent presence and absence of distinct features, the Naïve Bayes classifier greatly simplifies the training procedure [12], although independence is generally a poor practice. The Naïve Bayes method calculates the Gaussian distribution of the features for each class based on labeled training dataset. The maximum likelihood algorithm classifies data into definite classes based on conditional probabilities.

Neural networks have attracted more attention compared to other approaches in many application areas, including network traffic classification. Neural networks with a built-in ability to modify their synaptic connections and weights to adapt in surrounding environment have the potential to address many of the problems encountered by rule-based algorithms. Their attributes of non-linearity and adaptiveness are especially desirable for P2P traffic identification. However, neural networks, like other supervised machine learning approaches, normally require a large diversity of training for real world operations since they need sufficient representative examples in order to capture the underlying structure in which generalization of cases is occurred. The complexity of calculation may also limit its applications in some categories, especially in data pre-processing while clustering techniques have advantages on those [13].

B. Unsupervised machine learning (clustering)

Compared to supervised machine learning techniques in which training datasets match pre-defined classes before testing, unsupervised machine learning algorithms explore natural classes using internalized heuristics to group data. Clustering methods classify data based on the similarity which can be measured by distance functions such as Euclidean, Tchebychev, or City Block. A good set of clusters should have intra-similarity and inter-dissimilarity. As an algorithmic framework for data analysis and interpretation, clustering has been widely used in understanding data, revealing fundamental phenomenon, and visualizing major tendencies. Clustering can be categorized into two types: *hierarchical clustering* uses previously established clusters to find successive clusters; and

partitioning clustering determines all clusters at once based on iterative procedures. Subspace clusters, correlation clustering and bi clustering are new emerging algorithms and have been adapted to practical applications. Density-based algorithms, such as DBSCAN (Density Based Spatial Clustering of Applications with Noise), probabilistic model-based techniques, for instance, AutoClass and K-means clustering have become popular.

DBSCAN is based on density-reachability. Density-connectivity is an example of density-based algorithms which find clusters starting from the estimated density distribution of corresponding nodes. In this approach, all points in a specific cluster are mutually density-connected. It requires two parameters: epsilon (the distance around an object that defines its eps-neighborhood) and minPts (minimum number of points to classify data into distinct classes) [14].

Expectation Maximization (EM) approach focuses on finding maximum likelihood. EM is used to group data into a small number of classes and then generates classification rules from distinct classes. The probabilistic model-based AutoClass allows automatic selection of the number of classes and soft clustering of the data, which allows data to be fractionally assigned to more than one cluster. It is effective for first step of classification or individual applications identification in network traffic engineering [3].

K-means clustering approaches partition data into classes based on distances and try to maximize the homogeneity intra-cluster by minimizing the error overall and heterogeneity inter-cluster by maximizing the error. The distance and objective functions are fundamental elements for clustering. The iteration changes the clusters' centers and alters the belongingness of data to distinct clusters until specifications are met. It can classify different types of TCP-based applications using the first few packets of the traffic flow [11].

C. Hybrid approaches

Semi-supervised machine learning approaches (also called Hybrid approaches) combine unsupervised and supervised learning techniques. Hybrid approaches use both a small quantity of labeled and a large amount of unlabelled data. This approach partitions training dataset into classes first using clustering algorithms, then maps unlabelled traffic data into identified classes with the existence of empty classes for further use. The approach is flexible, especially for new emerging unknown network traffic classification [15].

IV. FUZZY C-MEANS CLUSTERING AND PROPOSED MODIFICATIONS

Conventional hard clustering methods classify each data point into one cluster. Nevertheless, the observation of P2P traffic characteristics could be imprecise because of the ever-changing nature of IP network traffic. Fuzzy clustering, however, can classify a piece of data point, such as network traffic data information, into two or more clusters with different belongingness. Based on the fuzzy logic, which has a great resemblance with human decision making ability, FCM generates precise solutions and results from either a certain or an approximate set of input information. It can achieve optimal through iterations with the simultaneous change of cluster

centers and the membership values which are definitions of one data belonging to one class. Fuzzy clustering is more flexible than hard clustering and is practical for P2P traffic identification because of the natural treatment of data [2].

This section first describes FCM and GK FCM clustering methods. Both FCM and GK FCM methods can effectively generate clusters, but with high overhead. The modifications to FCM are then proposed to simplify the calculation and complexity, which is vital for real-time P2P applications that could consist of a large number of real-time traffic flows.

A. Fuzzy c-means clustering

Fuzzy c-means clustering has three requirements [16]:

1. Membership degree (pattern k belongs to cluster i):

$$0 \leq \mu_{ik} \leq 1, i = 1, \dots, C(\text{clusters}), k = 1, \dots, N(\text{patterns})$$

2. Total Membership (the sum of all membership degrees is constrained to 1):

$$\sum_{i=1}^C \mu_{ik} = 1, k = 1, \dots, N \quad (1)$$

3. No empty clusters are allowed and no cluster may contain all the data examples.

$$0 < \sum_{k=1}^N \mu_{ik} < N, i = 1, \dots, C \quad (2)$$

The procedure of clustering is an iterative process of minimizing the objective function,

$$J(X; U, V) = \sum_{i=1}^C \sum_{k=1}^N (\mu_{ik})^m d^2(x_k, v_i) \quad (3)$$

Where $m: [1, \infty]$ is a weighting exponent that determines the fuzziness of the clusters (fuzziness parameter). If $m=1$ the sets are crisp not fuzzy. $d^2(x_k, v_i)$ is the distance measurement.

The data are typically in geometric forms, such as lines or points. When points are used, the general form of the distance measurement between a point x_k and a cluster center v_i is given by:

$$d_{ik}^2 = (x_k - v_i)^T A_i (x_k - v_i) \quad (4)$$

When $A_i = I$, the measurement is based on the Euclidean distance and the clusters are with the spherical shape (traditional FCM or FCM).

When $A_i = F_i^{-1}$, the measurement is based on the Mahalanobis distance and the clusters are with the ellipsoidal shape where F_i is the Fuzzy Covariance Matrix of cluster i . The method is called GK FCM [4].

The objective of the FCM clustering is to find spherical clusters of approximately the same size. Its input is a data matrix $X = (N, M)$ and the number of clusters C ; its outputs are cluster centers V and fuzzy partition matrix U . Cluster centers V are initialized randomly.

The iterative procedure of the FCM algorithm is as follows,

1) Compute distances from x_k to each cluster center v_i :

$$d_{ik}^2 = (x_k - v_i)^T (x_k - v_i) \quad (5)$$

2) Compute partition matrix U (membership values):

$$\mu = \frac{1}{\sum_{i=1}^C (d_{ik}^2 / d_{jk}^2)^{1/(m-1)}} \quad (6)$$

3) Compute new cluster centers V based on U :

$$v_i = \frac{\sum_{k=1}^N (\mu_{ik})^m x_k}{\sum_{k=1}^N (\mu_{ik})^m} \quad (7)$$

These three steps will be repeated until a criterion is satisfied, typically a pre-configured threshold is reached.

B. Gustafson-Kessel FCM clustering

The Gustafson-Kessel (GK) clustering algorithm [4] is commonly used in various domains including image processing, classification and system identification. Contrast to the FCM algorithm, its main feature is the local adaptation of the distance metric to the shape of the cluster. In addition, it is relatively insensitive to the data scaling and initialization of the partition matrix.

The objective of the GK FCM clustering is to find ellipsoidal clusters of approximately the same size. Its inputs and outputs are exactly same with traditional FCM except the distance function.

The iterative procedure of the GK FCM algorithm is the same as the traditional FCM, except that the distance computation uses Equation (4) instead of (5). A_i used in Equation (4) is defined as:

$$A_i = \left| F_i^{1/M} \right| F_i^{-1} \quad (8)$$

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (x_k - v_i)^T (x_k - v_i)}{\sum_{k=1}^N (\mu_{ik})^m}$$

The procedure repeats the computations with the simultaneous change of cluster centers, membership values, and belongingness until a criterion is satisfied.

C. Modification to FCM clustering

The FCM has been shown to be an effective data clustering algorithm [13]. It is an iteratively optimal algorithm usually based on the least square method to make a fuzzy partition of the data set. The objective function of the traditional FCM is described in (3). However, it was found out that the modified linear distance, suggested by Bezdek [17], and as shown in (9) produced almost the same results in practice, with the

clustering center values identical to those from the least square method [18].

$$J(X; U, V) = \sum_{i=1}^C \sum_{k=1}^N (\mu_{ik})^m d(x_k, v_i) \quad (9)$$

Additionally, in this paper, the linear distance measurement using City Block distance instead of Euclidean distance is adopted to reduce the computational overhead. City Block distance is calculated as:

$$d_{ik} = |x_k - v_i| \quad (10)$$

for a data point x_k and a cluster center v_i .

The proposed modifications to FCM reduce the complexity of the objective function from quadratic in (3) to linear in (9). Moreover, the complexity of the distance function also is reduced from quadratic plus square root calculations used in Euclidean distance to simply linear used for City Block distance, as shown in (10). The calculation is needed for each data point. Therefore, the total computation overhead for the entire dataset could be significant. The reduction of calculation time is vital for real-time P2P traffic applications, as the number of P2P flows could be huge and the response time needs to be very short.

V. SIMULATIONS AND COMPARISONS

The major role of FCM is to cluster data into different groups (classes or clusters). When the objects in the same clusters share more similar characteristics and the objects in different clusters are with more different features, the clustering result is considered to be better. Our experiments showed that the proposed method using simplified objective function and distance function can generate results and classification accuracy similar to that of the conventional FCM, while the method improves the computational speed. For the traditional FCM, the time complexity is $O(NDC^2I)$, where N is the number of patterns and C is the number of clusters (see section IV), D stands for data dimensions, and I equals the number of iterations of calculations. The modified FCM has time complexity of $O(NDCI)$. With more emerging P2P applications, the complexity reduction from proportional of C^2 to C would greatly enhance the efficiency of this approach. As a result, the proposed method has the potential to be used for online network traffic monitoring or pre-processing for network management [19].

The following two sub-sections demonstrate the results generated by traditional FCM and the modified FCM, respectively, for a comparison. The application of the proposed FCM to P2P traffic classification was then demonstrated.

A. Results of FCM clustering

The results produced by traditional FCM clustering are presented in Fig. 1. The quasi-random two-dimensional data, as depicted in Fig. 1(a), were used for demonstration. Fig. 1(b) shows the result for two clusters, while Fig. 1(c) present the result for three clusters. The black labels ($\times$, $\circ$, and Δ) in the figures are cluster centers.

The membership values of the data points to the cluster centers are shown in Fig. 2. The vertical axis illustrates the

membership values for data presented on the horizontal 2-dimensional plane. Fig. 2(a) presents the membership values for cluster 1 after those three clusters, as shown in Fig. 1(c), were generated; Fig. 2(b) is for cluster 2, and Fig. 2(c) is for cluster 3. Threshold values were to determine which cluster each data point belongs to.

B. Results of modified FCM

As stated in the previous section, this paper modifies the objective function from quadratic to linear, and changes the distance function from Euclidean distance to City Block. The modifications were implemented in Matlab by changing the related M-functions.

Using the same data as the FCM, the clustering results with three clusters for the modified FCM methods are shown in Fig. 3. In Fig. 3(a), only the objective function was changed from quadratic to linear; Fig. 3(b) only the distance function was altered from Euclidean to City Block; and Fig. 3(c) both the objective function and the distance function have been modified.

Compared with Fig.1(c) which depicts results generated from the traditional FCM, there is not much clustering difference. The cluster centers for the modified FCM remain unchanged, the belongingness of objects to clusters are almost the same as the traditional FCM. Only a few boundary points were grouped into different clusters. In addition, Fig. 3 also reveals that the difference between the variations (a), (b), and (c) is also small.

The experiments were used to study the feasibility of the proposed changes to FCM and to compare the results for both approaches. The results demonstrated that the proposed FCM is feasible from the clustering accuracy perspective; the next step is to apply the proposed FCM to P2P traffic classification.

C. P2P traffic classification using modified FCM clustering

We used the data captured in an empirical platform and extracted the features from traffic flows. Many features can be obtained directly from packet headers or their statistic characteristics such as average, min, max, and standard deviation. The dataset consists of 41 features. Some features are correlated with each other, while others are independent. Feature reduction was adapted to closely related features. Feature selection is crucial for clustering techniques. For P2P traffic clustering, after feature reduction, we grouped features into two categories: traffic flow and packet statistics.

From the traffic flow perspective, features that are essential for flow identification and flow characteristics include:

- number of packets sent for a flow
- number of packets received for a flow
- duration of the flow
- protocol
- source and destination ports (2 features)

From the perspective of packet statistics, we consider:

- total number of packets
- mean packet length
- mean payload length

- mean packet inter-arrival time
- the average sent packet size
- average received packet size
- variances of average sent packet size
- variances of average received packet size
- byte ratio of sent packets
- byte ratio of received packets

We examined those 15 features listed above and applied the clustering to the experimental data using those 15 features. We did not find significant clustering difference using the modified distance function from Euclidean to City Block and the changed objective function. This demonstrates that the proposed modifications to FCM would generate clusters that are comparable with those created with the traditional FCM. However, the computational speed has improved significantly.

On the other hand, when we used different feature combinations, we obtained apparently different clustering results, so feature selection can have direct impact not only on the efficiency for network traffic data clustering, but also clustering accuracy as well. This is also true for other clustering methods.

For Fuzzy clustering, the selection of the number of clusters is also crucial; otherwise the clustering results may be absurd. We applied the modified FCM on simulated P2P network traffic, as illustrated in Fig. 4 (a). The data consist of four simulated P2P traffic types: PPLive, BitTorrent, QQ, and Skype.

Fig. 4(b) shows the clustering results for four clusters, each representing one P2P traffic type. The results for four clusters were consistent for 8-10 experiments. Running the same data using the traditional FCM method generates similar results, i.e., there was no significant difference between the traditional and the modified approaches. This demonstrates that the modified FCM generates very comparable results to the traditional FCM.

On the other hand, the experiments also demonstrate that the selection of the number of clusters has a direct effect on the clustering outcome. For instance, Fig. 4(c) and 4(d) illustrate results of selecting five clusters for the simulated P2P traffic which has only four different traffic types: PPLive, BitTorrent, QQ, and Skype.

As presented in Fig. 4(c) and Fig. 4 (d), the results are very different if the number of clusters is set to five for the clustering process. Obtaining very different results using the same number of clusters is an indication that the number of clusters is incorrect or does not represent the actual data.

The effect of the number of clusters is further explained here. If Fig. 4(d) represents the actual traffic data and Fig. 4(b) is the results when we choose four clusters in the beginning of the clustering, then only the data in the yellow region of Fig. 4(d) would be misclassified. If Fig. 4(c) represents the actual traffic data and we obtain the results shown in Fig. 4(b) when four clusters are chosen, it means that more data have been misclassified. In both cases, the clustering errors are caused by

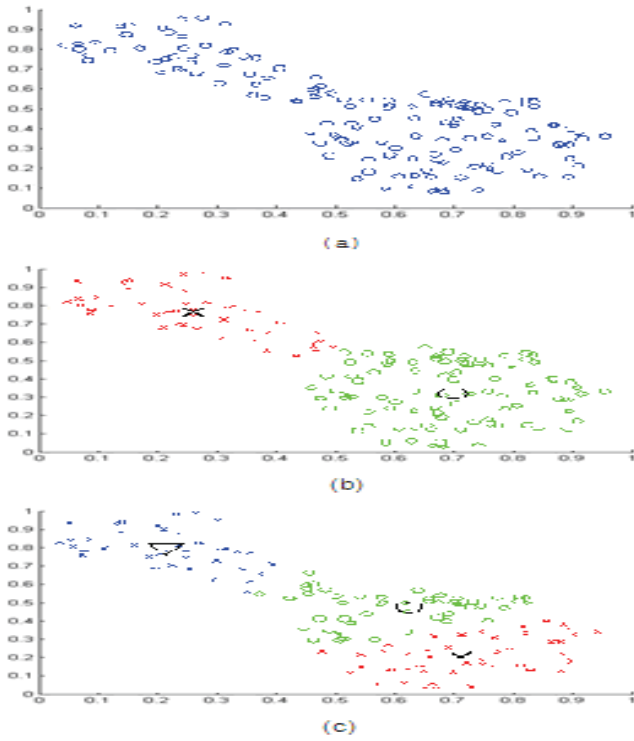


Figure 1. Clustering results of traditional FCM: (a) quasi-random data to be clustered, (b) clustering result with 2 clusters, (c) clustering result with 3 clusters

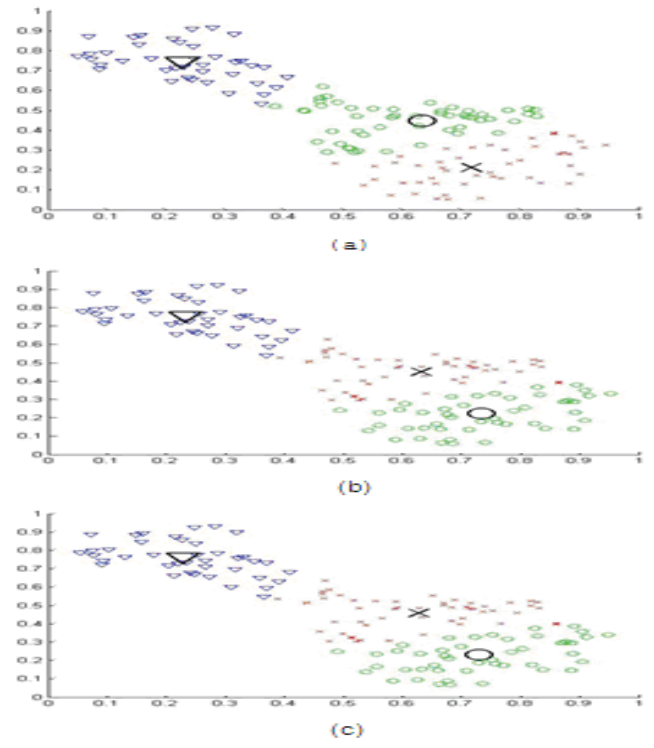


Figure 3. Clustering results of modified FCM: (a) modified objective function, (b) modified distance function, (c) modified objective and distance functions

the inappropriate selection of the number of clusters, which is crucial for FCM-based clustering.

If we obtain different results, e.g., results illustrated in Fig. 4(c) and Fig. 4(d), for different runs using the same number of clusters, cross-examination is needed. In this case, more data may be needed to validate the classification. Normally FCM does not generate reasonable or correct results if the number of clusters is wrong or inappropriate. We can also examine the membership values to determine if the clustering results should be accepted or rejected, or if more data are needed for verification.

In real-time classification for P2P traffic in which new applications are emerging, re-clustering is likely needed. With the modified FCM method, the speed is much faster than the traditional FCM, which is another important factor to consider for comparison for frequent re-clustering.

VI. CONCLUSIONS AND FUTURE WORK

The feasibility of automatic P2P traffic classification using machine learning techniques was explored in this paper. Classification accuracy and computational overhead are critical factors to consider for the high dimensional network traffic dataset. The fewer the features we select, the lower the computational overhead is with possibly lower accuracy. We compared different feature combinations for feature selection and reduced the feature set to 15 critical ones for analysis. Further, this paper modified the distance function from Euclidean (involving square and square root calculations) to City Block (linear calculations only) and changed the objective

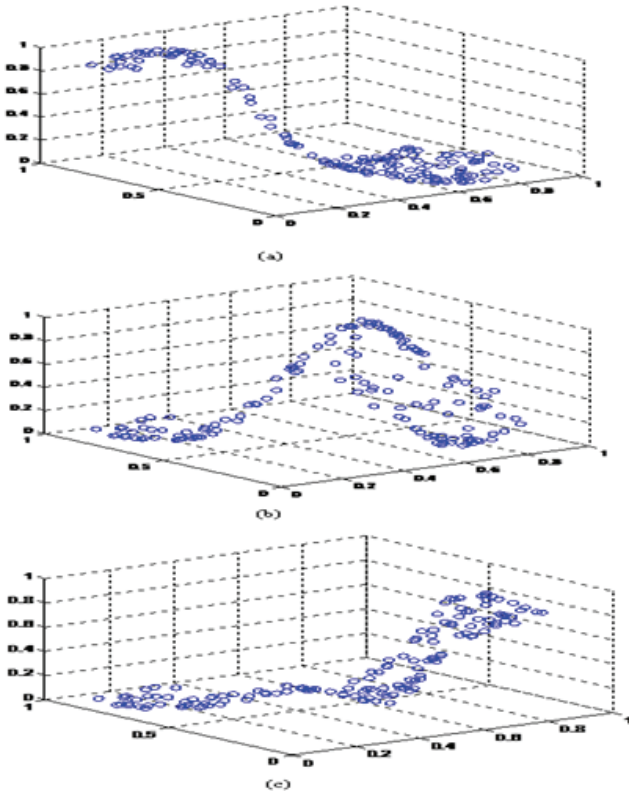


Figure 2. Membership values for: (a) cluster 1, (b) cluster 2, (c) cluster 3 after being grouped into 3 clusters

function from quadratic to linear to decrease the computational complexity. Computational overhead is crucial for a large volume of real-time P2P traffic monitoring. With the proposed modifications to FCM, the efficiency is improved, while the results obtained from our data were very similar to the traditional FCM.

Performance estimation of the modified FCM clustering based on overall accuracy, precision, and recall can be conducted in the future. We are also working on getting real traffic data using tcpdump on campus for further investigation.

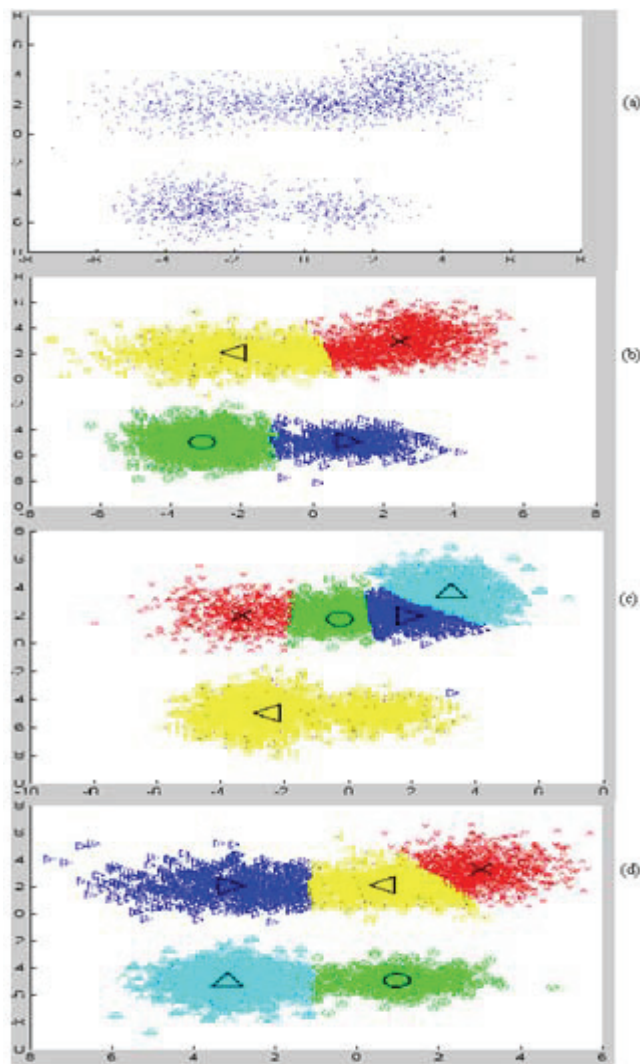


Figure 4. Simulated P2P traffic classification using modified FCM: (a) traffic data 2D visualization, (b) results for 4 clusters, (c)-(d) results for 5 clusters

REFERENCES

- [1] T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 56-76, 2008.
- [2] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, and D. Sadok, "A survey on Internet traffic identification," *IEEE Communication Surveys & Tutorials*, vol. 11, no. 3, pp. 37-52, 2009.
- [3] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," *Proc. of SIGCOMM'06 Workshops*, 2006.
- [4] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with fuzzy covariance matrix," *Proc. of IEEE CDC*, pp. 761-766, 1979.
- [5] WIKIPEDIA, Traffic measurement (telecommunications), Dec. 2010, [http://en.wikipedia.org/wiki/Traffic_measurement_\(telecommunications\)](http://en.wikipedia.org/wiki/Traffic_measurement_(telecommunications))
- [6] M. Crovella and B. Krishnamurty, *Internet Measurement: Infrastructure, Traffic and Applications*, Wiley, 2006.
- [7] IANA, Port Numbers, <http://www.iana.org/assignments/port-numbers>
- [8] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated construction of application signatures," *Proc. of SIGCOMM'05 Workshops*, 2005.
- [9] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," *Proc. of WWW 2004*.
- [10] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," *Proc. of the IEEE Conference on Local Computer Networks*, 2005.
- [11] Y. Liu, W. Li, and Y. Li, "Network traffic classification using K-means clustering," *Proc. of the 2nd International Multisymposium on Computer and Computational Sciences*, IEEE 2007.
- [12] 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*, vol. 36, no. 5, pp. 7-15, October, 2006.
- [13] W. Pedrycz, *Knowledge-Based Clustering*, John Wiley & Sons. 2005.
- [14] M. Ester, H. Kriegel, J. Sander, and X. Xu, "Adensite-based algorithm for discovering clusters in large spatial database with noise," *Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.
- [15] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson, "Semi-supervised network traffic classification," *Proc. of SIGMETRICS'07*, June 12-16, 2007.
- [16] J. V. Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*, John Wiley & Sons, 2007.
- [17] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Pattern Analysis and Machine Intelligence* 2 (1), 1980.
- [18] L. Ma, R.C. Staunton, "A modified fuzzy c-means image segmentation algorithm for use with uneven illumination patterns," *Pattern Recognition*, Vol. 40, 2007.
- [19] V. S. Rao, "Comparative Investigations and Performance Analysis of FCM and MFPCM Algorithms on IRIS Dats," *Indian Journal of Computer Science and Engineering*, vol. 1, no. 2, pp. 145-151, 2010.