An Improved Classification Model for Wide Area Networks with Low Speed Links

Kate Takyi, Amandeep Bagga

Abstract*: *The task of network administrators to identify and determine the type of traffic traversing through the network is very critical with the rapid growth of new traffic each day. Considering wide area networks with limited resources in terms of low speed links, quantified amount of packets are likely to be lost which lowers the quality of service. The classification procedure in such scenarios can also be affected due to the limited features extracted from the various fragments of packets that will successfully get to the destination node or server. We propose a hybrid cluster and label algorithm, which is able to classify application traffic or packets, utilizing restricted traffic features, few packets and at the same time maintains a low complexity and good classification accuracy. A wide area network exposed to extreme packet loss scenario is designed and implemented using OMNET ++ simulation to generate a dataset. The proposed model is built and tested in MATLAB simulation environment. Evaluation results shows that our proposed semi-supervised algorithm achieves an accuracy of 92.4% in classification with lower error rates of 7.4% and 2.9839 seconds processing time.

Index Terms*: *Clustering Techniques, K-Medoids, Packet Loss, Support Vector Machines

I. INTRODUCTION

Data transmission from source to destination networks can be intercepted and changed with important information extracted by people with malicious intentions. Also, malicious content can be hidden in traffic flows which can cause harm or deplete network resources. It is therefore a necessity for network administrators to accurately distinguish and recognize traffic allowed in and out of their network making the need of packet classification a requirement. Clustering techniques, a part of machine learning approach over the decades have been validated to classify traffic types accurately. Clustering techniques has been applied to network systems in various ways to classify traffic including intrusion detection systems for anomaly detection [1] [2] [3] [4]. Furthermore, other application areas in network security, network management and quality of service purposes in networks have been achieved through clustering techniques [5] [6] [7] [8]. Before clustering approaches emerged, Port-based and Payload approaches were used to classify data packets and traffic in networks. Known ports in the database of registered ports ascribed from the Internet Assigned Numbers Authority are utilized by Port-based method, whereas the Payload approach examines the packets in the flow critically to detect known signatures and classify packets based on similarities found with existing knowledge based signatures [9] [10]. Clustering techniques for traffic classification emerged to address the drawbacks in Port-based and Payload traffic classification, such as poor classification when dynamic ports and IP layer encryption surfaced, challenge of keeping an up to date signature database and handling of proprietary protocols [11].

Machine learning methods fall under supervised learning, unsupervised learning and semi-supervised learning. Supervised learning has the output of clustering known a priori from the labelled examples. Unsupervised learning employs flow characteristics and features to predict the clusters with no labelled examples. Semi-supervised learning works with a mixture of both labelled and unlabeled dataset, supplying a limited form of supervision with the labelled examples present. Clustering techniques can be classified as an unsupervised learning or semi-supervised method of machine learning. Though clustering techniques are efficient in classifying network traffic and flows, in a network scenario where habitual congestion leading to packet loss is present [7], the quality of service can be affected and minimized. To address this problem and enhance the quality of service in networks during the classification process, we propose a hybrid model of clustering to classify packets using restricted traffic features, few number of packets whiles maintaining a high accuracy and low time complexity. The objectives of the research can be summarized as:

- To identify the effect of packet loss in classification procedure
- To propose a hybrid model of classification to classify packets using restricted traffic features, few number of packets whiles maintaining a high accuracy and low time complexity.

II. LITERATURE REVIEW

Over the last two decades, a lot of clustering techniques works have been proposed. McQueen’s [12] non-hierarchical method of partitioning, captioned as the K-means algorithm, was adopted by Lloyd [13] to segregate datasets into clusters on the basis of a predefined number of previously selected centroids. The centroid is a point which

Revised Manuscript Received on May 23, 2019.

Kate Takyi, Department of Computer Applications, Lovely Professional University, Phagwara, India. Email: katetakyi@yahoo.com

Amandeep Bagga, Department of Computer Applications, Lovely Professional University, Phagwara, India. Email: amandeep1@lpu.in Corresponding Author

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

represents the central position within a cluster space. The objective of the algorithm in the clustering process is to diminish the errors in distance computation. Adopting the Euclidean distance, the centroid of *k* number of clusters is calculated iteratively, until a convergence measure is gained. K-means has emerged to be a good blueprint for development of other clustering algorithms due to its low computational costs [14] - [17].

Hirvonen and Laulajainen [14] presented a two phased classifier that detects unknown flows in the network not trained during the classification process. The assignment phase and labelling phases of their approach employs the K-Means algorithm and resulted in classifying 97.8 % of target applications correctly. Though the authors mentioned their method as lightweight, the computational cost is not discussed. Furthermore, they only compared their results to the efficiency of traditional Port-based classification which has already been accepted as less efficient in literature.

Hajikarami *et al.* [18] proposed a high speed link two layered lightweight system using *k* classifiers in the first classification phase instead of only one classifier (where *k* represents number of classifiers). The objective is to limit cost, resources and memory consumed when classifying flows. New signature flows not captured in the existing knowledge base are labelled as unknown. An alert is signaled to the network administrator for examination of these new flows and appropriate changes are made to the classifier. The system is able to classify flows from applications with an accuracy of 99.5% within a time of 41.28 seconds. The results showed that 60% of newly introduced flows from applications were classified as unknown to avoid misclassification. Zander *et al.* [19] suggested an unsupervised automated method of classification using statistical flow characteristics. McGregor *et al.* [20] Expectation Maximization (EM) algorithm and Stutz and Cheeseman [21] AutoClass algorithm are employed to first partition the flows into bi-directional flows to extract the needed flow characteristics. With this, a better separation of the different applications in a flow trace is achieved resulting in an average accuracy of 86.5% in classification and median of approximately 95% for selected individual applications. It is however not known how the performance on larger datasets will result.

Erman *et al.* [22] researched into classifying traffic and proposed a semi-supervised method for distinguishing known and unknown applications with flow characteristics. The problem of class imbalance (high accuracy in flow and low accuracy in byte) present in Erman *et al.* [23] is addressed by giving a good representation of both the mice and elephant flows. High byte accuracy above 90% is achieved with real-time traces collected in a time span of 6 months. Wang *et al.* [24] designed a semi-supervised strategy known as Set Based Constrained K-means (SBCK). Along with some background information of the TCP/IP flows, statistical flows are first extracted. To derive the constraints, Gaussian mixture density is utilized to model the observed data. The introduction of feature discretization in the flow clustering as asserted by the authors increases the level of clustering accuracy. The method of assignment for the proposed work is the distinguishing feature from the traditional K-means. In that, K-means processes each data point independently whiles SBCK regards equivalent samples as a whole, based on the subsidiary set based constraints. SBCK achieves a percentage range of 85% to 91% without the incorporation feature discretization and 94% to 97% is attained with feature discretization.

Wang *et al.* [25] proposed work clusters application traffic based on quality of service (QoS) requirements, with the implementation of Deep Packet Inspection (DPI) in software defined networks (SDN). Detection of incoming long lived flows is assigned to an SDN switch. Parameter values including Hurst packet, port number and average packet inter-arrival time are fed into a mapping function to classify traffic flows into their respective QoS classes. Laplancian Support Vector Machine (LapSVM) semi-supervised algorithm is utilized by the QoS classifier residing in the centralized SDN controller to obtain a coarse grained classification. Accuracy exceeding 90% is achieved giving an indication of a proficient classifier.

Achunala *et al.* [26] relied on a prototype classifier in OMNET introduced an effortless packet classification. Their research establishes a novel method to classify packets excluding payload data or information. They propose an Inter-arrival Precision (IATP) clustering algorithm which utilizes a clustering algorithm to cluster the training data and further partition them into smaller cluster subsets to be labelled. An accuracy of 85% to 95% is achieved after classification. However, the authors assert the fact that the results obtained does not represent real time classification.

Identification of Traffic proof methodologies that depend on heuristics got from examining the patterns in hosts communication have likewise been proposed [27] - [29]. For instance, Karagiannis *et al.* [28] built up a technique that use the social, practical, and application practices of hosts to recognize traffic classes. Simultaneous with [28], Xu *et al.* [30] built up an approach, in light of data theoretic methods and data mining, to find practical and application standards of conduct of hosts and the resources utilized by the hosts. They accordingly utilize these examples to construct general traffic profiles. For better understanding of our proposed hybrid approach, the K-Medoids algorithm and Support Vector Machine (SVM) algorithm will be discussed under the proposed algorithm section. For further research, renowned works in literature [31] - [35] can be referenced.

III. RESEARCH METHODOLOGY

In the network topology in Fig. 1, we subject the wide area network to a lot of congestion leading to packet loss as a result of fragmented packets with the parameters in Table I. Congestion and packet loss happen due to the use of low speed links and higher arrival rates of packets. The links are not able to handle the rate of incoming packets, leading to packet loss. Random generation of application messages (FTP, VoIP, Sound/Audio, and HTTPs) are initialized and exchanged from one network to another. Messages are exchanged from the clients from one network and transported the other network through the border routers using Boarder Gateway Protocol (BGP) with Transport Control Protocol (TCP) or User Datagram Protocol (UDP) where required.

The simulation is allowed to run for a span of 150 hours with the data logs collected at the server. The gathered data logs are saved in .VEC (vector) file format by OMNET++ to be used as the dataset. The file holds statistical records of the data values as a function of time. Among these are packet types, transmission time, hop counts, latency, inter-arrival time of packets, mean, standard deviation and all other record of time events with respect to the requirements of the network. The dataset is converted to .CSV file format, compactible with MATLAB for clustering and classification with the proposed algorithm. The overall flow of methodology used is illustrated in Fig. 2. The system and software requirements for the simulation tools used are Ubuntu Linux version 16.04, 10GB ram and 2.20 GHz (4CPUs) processor.

IV. PROPOSED ALGORITHM

To improve the quality of service in Wide Area Networks (WANs) and Campus Area Networks (CANs) during traffic classification procedure, we propose a hybrid (semi-supervised) cluster and label model. The aim of our method is to classify application traffic flows using restricted traffic features, small number of packets while maintaining lower error rates with less time and high classification accuracy. The proposed semi-supervised algorithm is a hybrid algorithm consisting of the K-Medoids and Support Vector Machine algorithms (SVM+K- medoids). The advantages of these algorithms are exploited to create a novel algorithm.

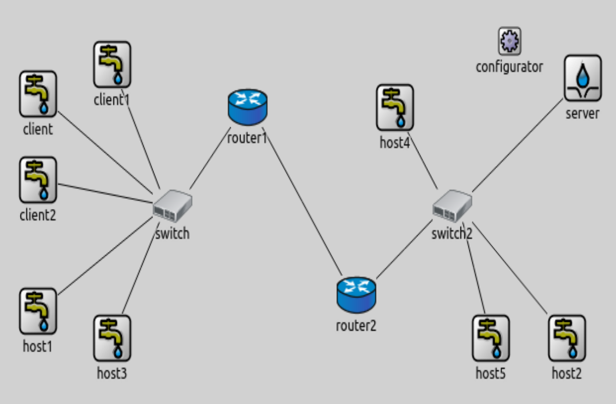


Fig. 1. Network Topology of Proposed Scenario

TABLE I. PARAMETERS USED FOR SIMULATION

Simulation Parameter	**Value / Type**
Simulation Time	150 hours
Channel Type	wired
Channel Delay	10us
Link Speed	Client – Server = 10mbps Client /host – Switch = 10mbps Router – Router = 100mbps
Packet Length (in time)	45ms
Packet Size	5120kbps
Interval	100ms
Sampling Rate	7000Hz
Send Bytes	1000000000 bytes
Protocols	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Queue Type	Drop Tail Queue
Switch Relay Unit Type	Mac Relay Unit
Routing Protocols	BGP (Boarder Gateway Protocol)

A. K-Medoids Clustering Algorithm

K-medoids clustering [36] is an improvement of the traditional K-means. K-means selects the mean distance of the points in the cluster as its best representation of a cluster. The mean does not necessarily represent a point in the original dataset. Mean computation is done using the Euclidean distance. K-medoids on the other hand, chooses one of the data points in a particular cluster as the cluster representative. These selected representatives are also termed as exemplars. The exemplars contribute to the reduction of the total sum of the data objects' dissimilarities. Medoids are the centrally located point in the cluster. K-medoids furthermore has an advantage of using any similarity measure like Euclidean distance and Manhattan distance unlike K-means which is not likely to converge with distances not consistent with the mean.

B. Support Vector Machine Algorithm

Support Vector Machine [37] rely on supervised learning models possessing related learning algorithms, for analyzing large amount data for classification, regression and pattern identification purposes. With SVM, almost the entire attributes are utilized to create parallel partitions giving rise to parallel lines called hyperplanes with margins in high-dimensional space. The given data is separated into classes using the hyperplane margins. Greater margins are directly proportional to lower error rates of the classifier. SVM has an advantage of being flexible and robust which generally gives its exact precision predictions. It is however sensitive to the kernel parameters selected for its implementation leading to a possible high computational complexity. SVM can be linear or non-linear depending on the dataset to be classified. Linear SVM accepts that the examples to be trained in space are parted by visible gap. A straight hyperplane separating the classes is predicted. The essential concentration while drawing the hyperplane is on expanding the margin from the hyperplane to the closest data point of either class. Real-time dataset is for the most part scattered up to some degree. Data division into various classes based on a straight and linear hyperplane cannot be viewed as a preferable decision. To address this, Vapnik *et al.* [38] recommended making non-linear classifiers by incorporating kernel functions to maximize the margins of hyperplanes.

C. Algorithm Implementation

The algorithm and its implementation are conducted in MATLAB simulation tool. The raw dataset is filtered to obtain the relevant data contents. K-medoids algorithm in conjunction with Silhouette tool is used to identify the clusters and outliers. Since a real-time and non-linear dataset is being used, there is a tendency of overlapping classes. A kernel trick for SVM must be utilized to transform the data into a higher dimensional space for accurate classification. A hybrid or multiclass kernel is incorporated by the application of both Cubic Polynomial and Gaussian kernels. The data set is transformed into a higher dimensional data space with the cubic polynomial function given in Equation (1).

$$K(x, y) = (\sum_{i=1}^{n} x_i\, y_i + C) \qquad (1)$$

where: x , y = vectors in input space

C = free parameter

n = size of training data

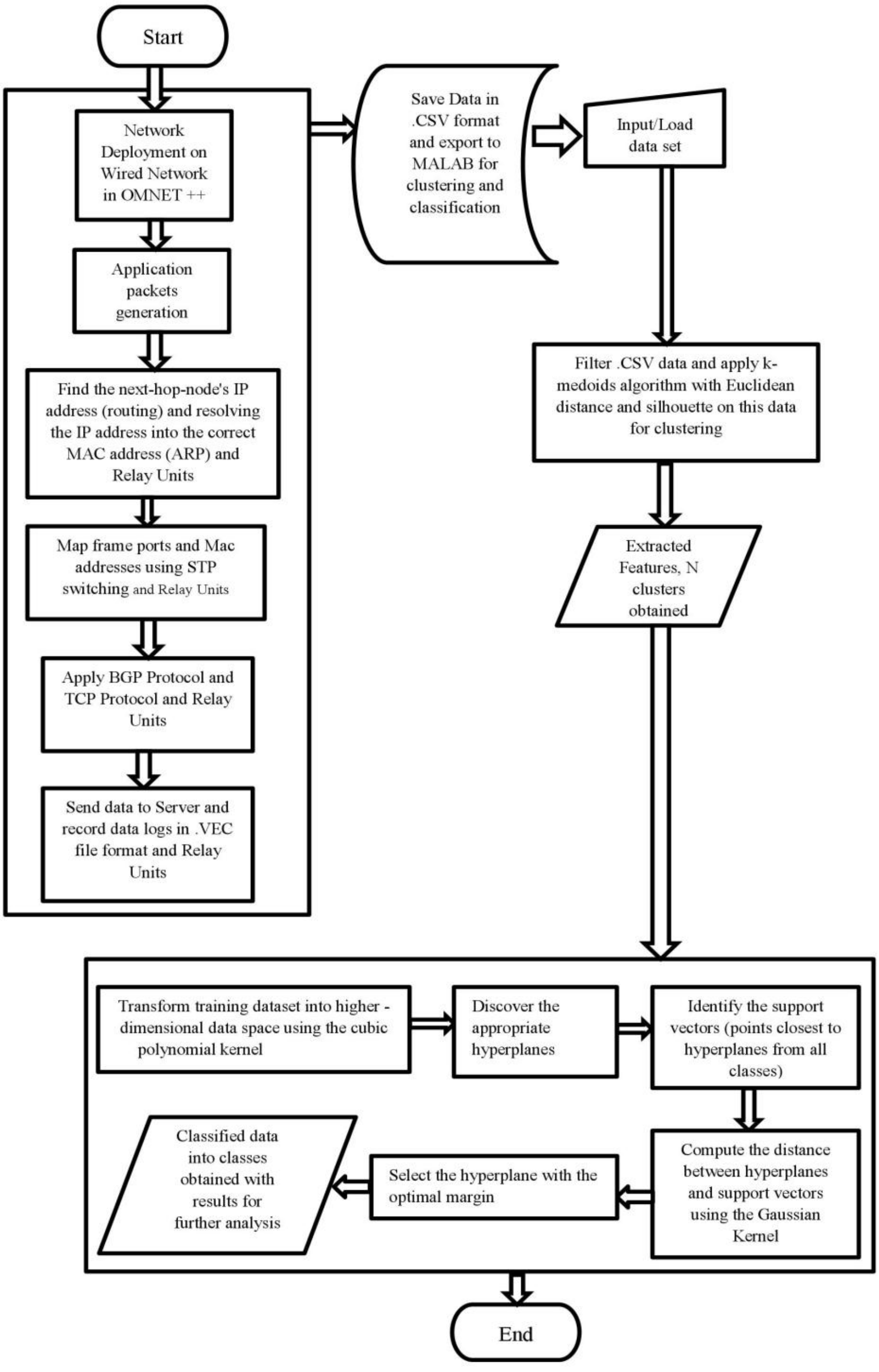

Start
Network Deployment on Wired Network in OMNET ++
Application packets generation
Find the next-hop-node's IP address (routing) and resolving the IP address into the correct MAC address (ARP) and Relay Units
Map frame ports and Mac addresses using STP switching and Relay Units
Apply BGP Protocol and TCP Protocol and Relay Units
Send data to Server and record data logs in .VEC file format and Relay Units
Save Data in .CSV format and export to MALAB for clustering and classification
Input/Load data set
Filter .CSV data and apply k-medoids algorithm with Euclidean distance and silhouette on this data for clustering
Extracted Features, N clusters obtained
Transform training dataset into higher - dimensional data space using the cubic polynomial kernel
Discover the appropriate hyperplanes
Identify the support vectors (points closest to hyperplanes from all classes)
Compute the distance between hyperplanes and support vectors using the Gaussian Kernel
Select the hyperplane with the optimal margin
Classified data into classes obtained with results for further analysis
End

Fig. 2 Overall Proposed Methodology

$C \geq 0$ designates a free parameter that trades off the impact of a single training example. The features extracted and number of clusters obtained after clustering with K-Medoids provides a form of supervision for obtaining hyperplanes and number of classes during classification. The points closest to the hyperplanes also known as support vectors from all classes are fetched. Distance between the hyperplanes and support vectors is computed using the Gaussian kernel function given in Equation (2).

TABLE II. PROPOSED ALGORITHM

SVM+K-Medoids Algorithm
Step 1: Load dataset
Step 2: For each column in the dataset: • Transform all contents of numeric strings to numbers • Substitute all non-numeric strings with NaN (Not a Number)
Step 3: Select *Km* as the Medoids for n data points at random.
Step 4: Compute the distance between the data points, n and selected Medoids *k* for the closest Medoids.
Step 5: Conduct a mapping from the Medoids to the data objects within the cluster
Step 6: Perform a swapping for each medoids *m* and the data object connected to it in order calculate the total cost using the Euclidean Distance
Step 7: Choose the medoids with the smallest cost
Step 8: Repeat steps 4-7 until a convergence is reached if and there is no change in the assignment process.
Step 9: Transform training dataset into higher-dimensional data space using the cubic polynomial kernel.
Step 10: Discover the appropriate hyperplanes
Step 11: Identify the support vectors (points closest to hyperplanes from all classes)
Step 12: For all possible hyperplanes, do this • Compute the distance between hyperplane and support vectors using the Gaussian Kernel
Step 13: Select optimal hyperplane for which the margin is maximized to classify and assign data points to appropriate classes.

$$K\,(x, y) = \exp\left(-\ \frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (2)$$

where: x, y = vectors in input space
$\|x-y\|^2$ = squared Euclidean distance

The line for which the margin is maximized is selected as the optimal hyperplane. The optimal hyperplane classifies the data points into their appropriate classes. The proposed algorithm is as shown in Table II.

V. TEST AND ANALYSIS

The incorporation of the parameter values in Table I will result in a high packet drop count with the number of sent bytes exceeding the amount of bytes the link can handle at a time. A graph of packet drop count against inter-arrival time is plotted and shown in Fig. 3. From Fig. 3, as packet drop count increases, the inter-arrival time of the packets increases. This shows that the latency to reach the destination is increased with rapid packet loss in the network. Hence, only limited or small amount of flows can be classified at a time fulfilling the first objective. Five set of clusters as shown in Fig. 4 are revealed after applying K-medoids with silhouette clustering. The figure depicts the extent of the inliers and outliers with respect to each cluster. The longer bars (inliers) in a cluster represent the data points possessing similar features confirming its belongingness to the cluster. The shorter bars are the outliers to that particular cluster. Precision gives prediction of accuracy measure. A graph of precision against inter-arrival time is plotted. A higher inter-arrival time is as result of increasing packet drop count from the scenario in the topology. Hence, in a worst scenario of congestion or high packet loss, precision values ranging from an average of 85% to 94% can be obtained with the proposed hybrid approach. Using a 5-fold cross validation, an average of 92.4% accuracy is achieved when classifying packets, falling within the percentage range of the precision values. The measurements used for evaluation including Accuracy, Precision and Error rates are computed using Equation (3), Equation (4) and Equation (5) respectively.

$$Accuracy\ =\ \frac{TP+TN}{TP+TN+FN+\ FP+FN} \qquad (3)$$

$$=\ \frac{Number\ of\ Correctly\ Predicted\ Classes}{Total\ Number\ of\ Predicted\ Classes}$$

$$Precision\ =\ \frac{TP}{TP+FP} \qquad (4)$$

$$=\ \frac{Correctly\ predited\ classes}{Correctly\ predicted\ +\ Incorrectly\ predicted}$$

$$Error\ =\ \frac{FP+FN}{TP+TN+FN+\ FP+FN} \qquad (5)$$

$$=\ \frac{Number\ of\ Incorrectly\ Predicted\ Classes}{Total\ Number\ of\ Predicted\ Classes}$$

where: TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

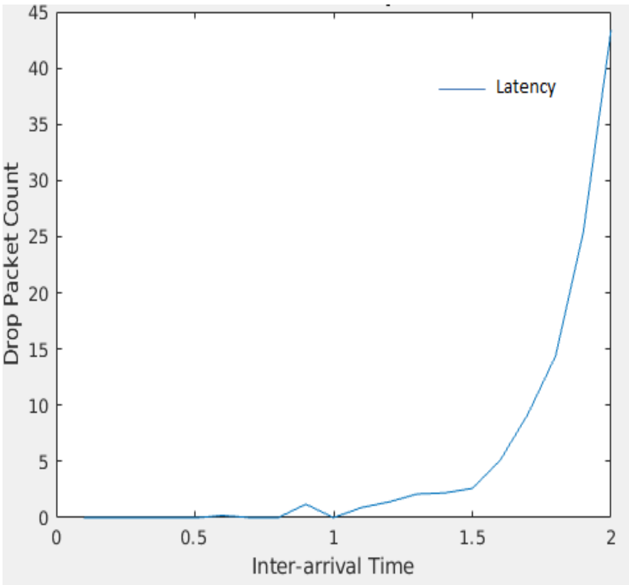


Fig. 3. Graph showing the effects of Packet Loss with Inter-arrival Time

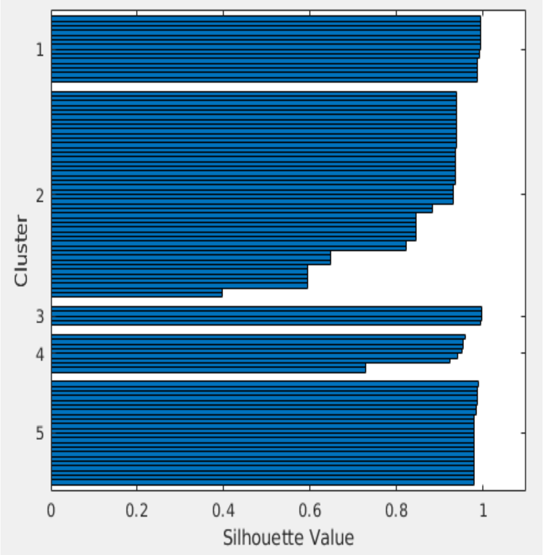
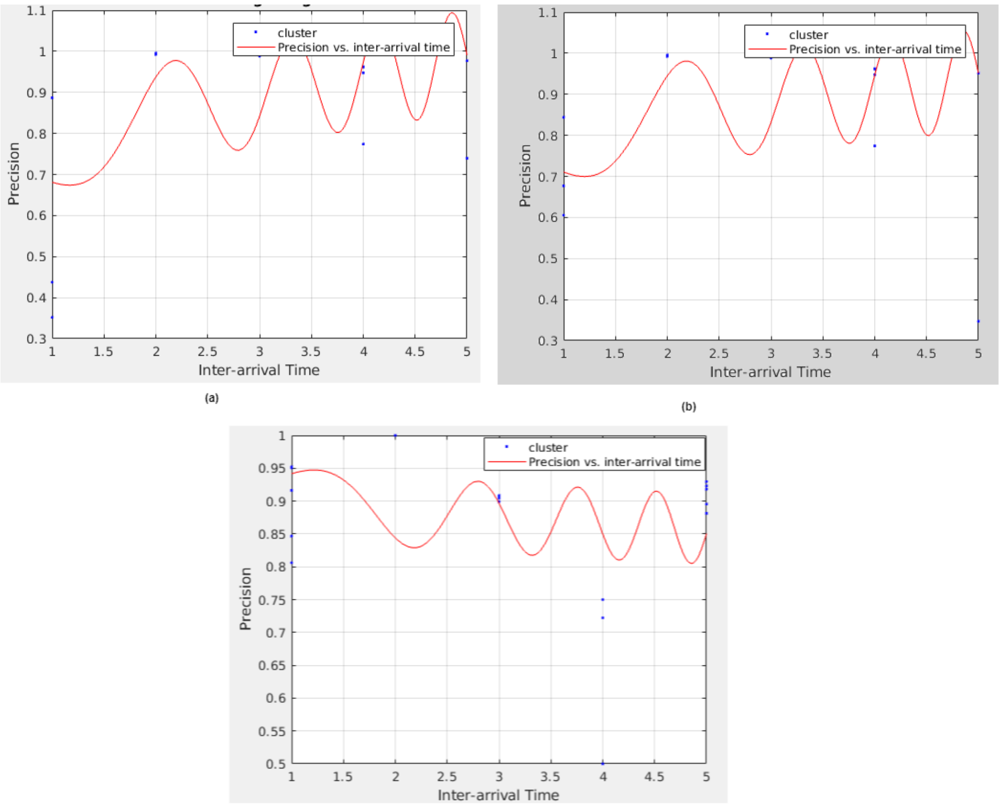


Fig. 4. Inliers and Outliers revealed after clustering

V. RESULTS AND DISCUSSIONS

To validate our model, we compare the proposed to KNN+K-means hybrid classifier [39] and K-Nearest Neighbour (KNN) classifier [40] using the same dataset with 5-fold cross validation. A graph of precision against inter-arrival time is plotted to derive the average range of precision values. From Fig. 5, 5(a) and 5(b) shows that the precision range of KNN and KNN+K-means are 62% to 94% and 70% to 93% respectively, whiles the proposed (SVM+K-Medoids) ranges from 75% to 94% in 5(c). We can conclude that the proposed model is likely to perform better and give a high accuracy prediction value.

From Fig. 6, the proposed in (6c) and KNN+K-means in (6b) both resulted in identifying 20 classes whiles KNN in 6(a) resulted in 14 classes. Thus, finer classes are revealed for the former two approaches. Out of the 14 classes, KNN in 6(a) correctly classified 3 classes (True positive Rate/TPR), 6 classes were misclassified (False Positive Rate/ FPR) and 5 classes were partially classified (falls under 3 classes due to overlapping features). For KNN+K-Means in 6(b), 1 class was classified correctly, 1 class is predicted for two classes and the rest misclassified. From 6(c) 14 classes were classified perfectly by the proposed approach out of the 20 classes whiles 6 classes were not classified correctly. Receiver Operating Characteristics (ROC) curve plots Sensitivity (true positive rate) against 100-Specificity (false positive rate) for varying cut-off points of a parameter. Fig. 7 shows the ROC for each classifier comparatively. The area under the ROC curve (AUC) depicts the measure of a



(a)

(b)

(c)

Fig. 5. (a, b & c) A Comparative graph of Precision for each classifier with clusters revealed

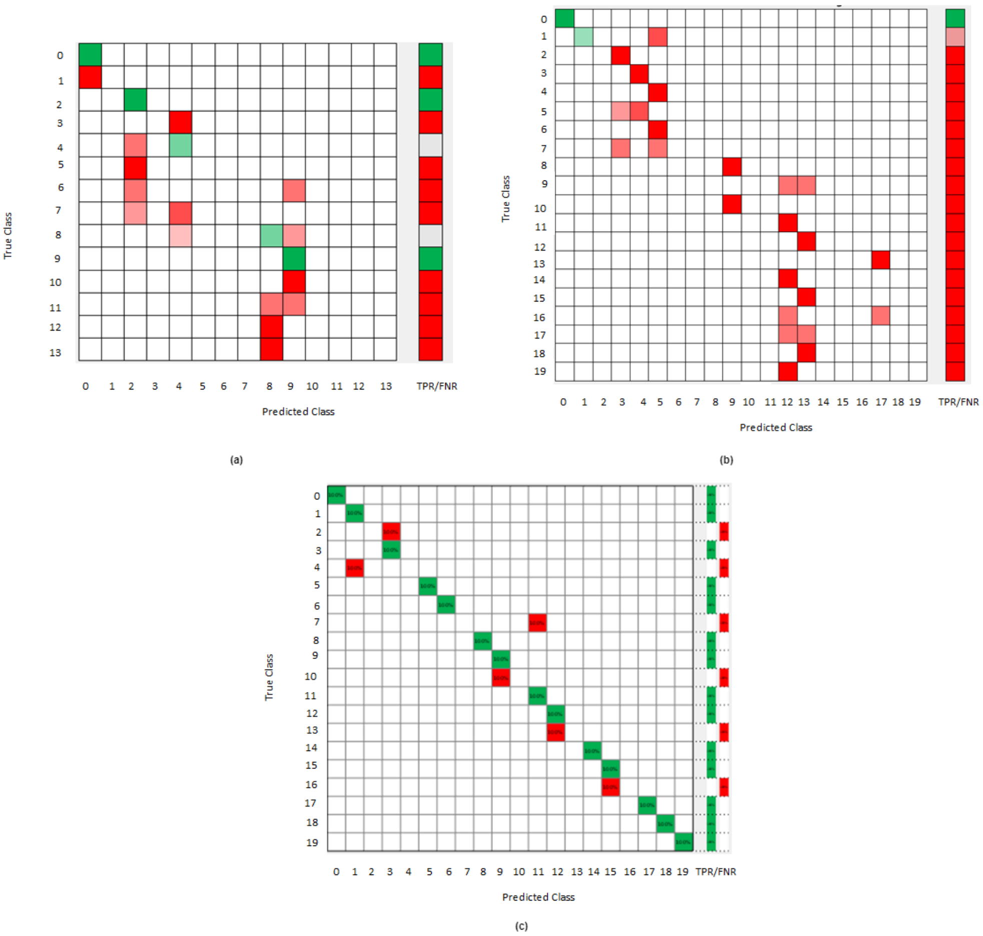


Fig. 6. (a, b & c) Confusion matrix showing classified classes for each classifier

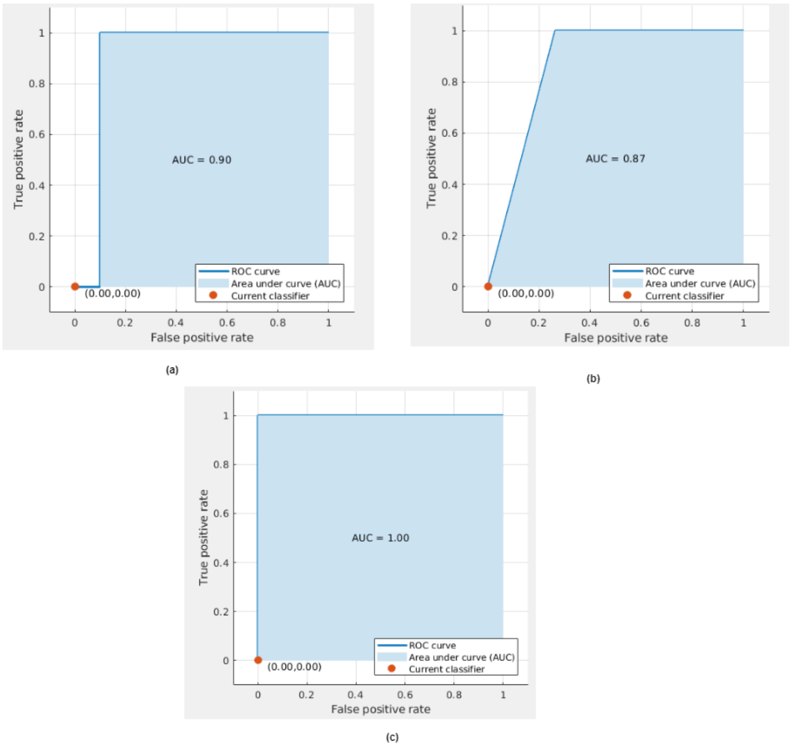


Fig. 7. (a, b & c) Area under ROC achieved for each classifier

parameter's performance to differentiate the various classes. The closeness of the ROC curve to the upper left corner denotes how high the overall accuracy of the model on a scale of 0 to 1, where 1 is highest and 0 is lowest. KNN in 7(a) achieved a value of 0.9 in 9(a), KNN+K-Means with 0.87 in 7(b) and SVM+K-Medoids with 1.0 in 7(c). This shows that the proposed model has the highest overall accuracy. In terms of accuracy and error rates, it can be concluded that the proposed hybrid model performs better than the two existing models.

The results after classification with KNN gives an accuracy of 73.79% whiles KNN+K-means gives an accuracy of 65.2% compared to the proposed giving 92.4% as revealed in Fig. 8. Also, the proposed classifier achieved lowest error rate of 7.6% in misclassification compared to the existing models. From the results above the proposed classifier has best accuracy in classification and also overcomes the issue of overlapping features by predicting distinct classes. In terms of training time, KNN outperforms the other two methods with 1.0767 seconds, followed by the proposed with 2.9839 seconds and KNN+K-Means with 4.1804 seconds. The best performance on KNN in terms of time complexity can be attributed to the fact that out of the three classifiers being compared, KNN requires least amount of resources and fewer instructions to be executed. Nevertheless, the efficacy of the proposed hybrid approach in terms of classification accuracy over compensates the amount of time and resources required. Therefore we can conclude that the objective to classify packets using restricted traffic features, few packets whiles maintaining a high accuracy and low time complexity is achieved, fulfilling the second objective. Table III illustrates a summary of results comparatively.

VI. CONCLUSION

In relation to networks with limited resources or low speed links that encounter extreme packet loss, quality of service is minimized in all the overall transmission as well as in the process of traffic classification. Traffic classification is a requirement in securing the network, hence network administrators make it a necessity to know traffic types that traverse through the network to foster security policies updates. Clustering techniques over the years has proved to be a very efficient approach in traffic classification. To improve upon the quality of service in such scenarios, a network with starved network resource is implemented in OMNET++. The proposed algorithm is used to classify the dataset generated. The proposed model with 5-fold cross validation is able to classify traffic with high precision and accuracy values performing better two existing renowned works. We wish to investigate the effects other quality of service parameters in the classification process in our future work and propose novel algorithms to that effect. Furthermore, although an accuracy of 92.4% is high, for future research we aim to improve upon it in order to minimize the error rate achieved.

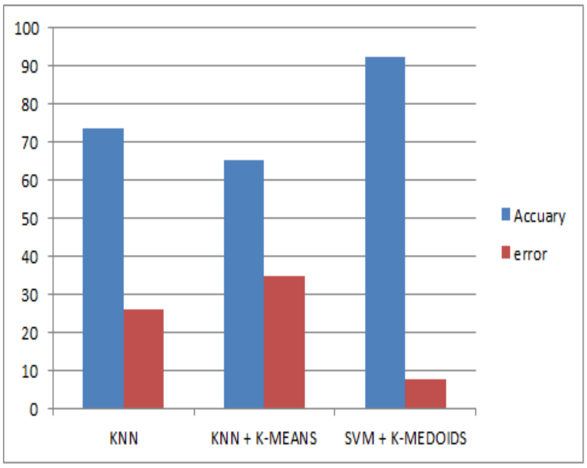


Fig. 8. A comparative graph of Accuracy and Error rates for each classifier

TABLE III. SUMMARY OF EVALUATION RESULTS

Classifier	Metrics (Parameter)				
	Precision Range (%)	Accuracy (%)	Error (%)	AUC	Time (s)
KNN	62 - 94	73.79	26.21	0.9	1.0767
KNN+K-Means	70 -93	65.2	34.8	0.87	4.1804
SVM+K-Medoids	85-94	92.4	7.6	1.0	2.9839

REFERENCES

1. H. Alaidaros and M. Mahmuddin. "Flow-based approach on bro intrusion detection." Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, no. 2-2, pp. 139-145, 2017.
2. A. Garg and P. Maheshwari. "Identifying anomalies in network traffic using hybrid Intrusion Detection System." In Advanced Computing and Communication Systems (ICACCS), 2016 3rd International Conference on, vol. 1, pp. 1-6. IEEE, 2016.
3. R.F.M. Dollah, M.A. Faizal, F. Arif, M.Z. Mas'ud and L.K. Xin. "Machine learning for HTTP botnet detection using classifier algorithms." Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 1-7, pp. 27-30, 2018.
4. T. Y. Christyawan, A. A. Supianto, W. F. Mahmudy. "Anomaly-based intrusion detector system using restricted growing self organizing map." Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no. 3, pp. 919-926, 2019.
5. A.E. Danganan, A.M. Sison and R.P. Medina. "An Improved Overlapping Clustering Algorithm to Detect Outlier." Indonesian Journal of Electrical Engineering and Informatics (IJEEI), vol. 6, no.4, pp. 401-409, 2018.
6. A. Abuarqoub, M. Hammoudeh, B. Adebisi, S. Jabbar, A. Bounceur and H. Al-Bashar. "Dynamic clustering and management of mobile wireless sensor networks." Computer Networks, vol. 117, pp.62-75, 2017.
7. A. Mohanty, S. Mahapatra and U. Bhanja. "Traffic congestion detection in a city using clustering techniques in VANETs." Indonesian Journal of Electrical Engineering and Computer Science, vol. 13, no.3, pp. 884-891, 2019.
8. S. H. Yoon, J. W. Park, J. S. Park, Y. S. Oh, M. S. Kim. "Internet application traffic classification using fixed IP-port." In Asia-Pacific Network Operations and Management Symposium,

pp. 21-30, Springer, Berlin, Heidelberg. 2009.

9. A. W. Moore. and K. Papagiannaki. "Toward the Accurate Identification of Network Applications." In PAM, vol. 5, pp. 41-54, 2005.
10. F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian. "Real-time traffic classification based on statistical and payload content features." In Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on, pp. 1-4. IEEE, 2010.
11. T. T. Nguyen and G. Armitage. "A survey of techniques for internet traffic classification using machine learning." IEEE Communications Surveys & Tutorials 10, no. 4 (2008), pp. 56-76, 2008.
12. J. MacQueen. "Some methods for classification and analysis of multivariate observations." In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, pp. 281-297, 1967.
13. S. Lloyd. "Least squares quantization in PCM." IEEE transactions on information theory, vol. 28, no. 2, pp. 129-137, 1982.
14. M. Hirvonen and J. P. Laulajainen. "Two-phased network traffic classification method for quality of service management." In Consumer Electronics, 2009. ISCE'09. IEEE 13th International Symposium on, pp. 962-966. IEEE, 2009.
15. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu. “An efficient k-means clustering algorithm: Analysis and implementation.” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 1, no.7, pp. 881-92, 2002.
16. J. Z. Xiao and L. Xiao. “Analysis and improvement for K-Means Algorithm.” In Applied Mechanics and Materials, vol. 52, pp. 1976-1980, Trans Tech Publications, 2011.
17. P. Luarn, H. W. Lin, Y. P. Chiu, Y. L. Lee and P. C. Shyu. “The Categorising Characteristics of Facebook Pages: Using the K-Means Grouping Method.” International Journal of Business and Management, vol. 11, no.2 pp. 60, 2016
18. F. Hajikarami, M. Berenjkoub and M.H. Manshaei. “A modular two-layer system for accurate and fast traffic classification.” In 2014 11th International ISC Conference on Information Security and Cryptology, pp. 149-154, 2014.
19. 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, pp. 250-257, 2005.
20. A. McGregor, M. Hall, P. Lorier and J. Brunskill. "Flow clustering using machine learning techniques." Passive and Active Network Measurement, pp. 205-214, 2004.
21. P. Cheeseman and J. Stutz. "Bayesian classification (autoclass): Theory and results in advances in knowledge discovery and data mining eds." Articles FALL, pp. 51, 1996.
22. J. Erman, A. Mahanti and M. Arlitt. “Byte me: a case for byte accuracy in traffic classification.” In Proceedings of the 3rd annual ACM workshop on Mining network data, pp 35-38, 2007.
23. J. Erman, A. Mahanti, M. Arlitt, I. Cohen and C. Williamson. "Offline/realtime traffic classification using semi-supervised learning." Performance Evaluation, vol. 64, no. 9, pp. 1194-1213, 2007.
24. 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, vol. 25, no. 11, pp. 2932-2943, 2014.
25. P. Wang, S. C. Lin and M. Luo. "A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs." In Services Computing (SCC), 2016 IEEE International Conference on, pp. 760-765. IEEE, 2016.
26. D. Achunala, M Sathiyanarayanan & B. Abubakar. “Traffic classification analysis using omnet++.” In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 417-422. Springer, Singapore 2018.
27. T. Karagiannis, A. Broido and M. Faloutsos. “Transport layer identification of P2P traffic.” In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, pp. 121-134, ACM, October 2004.
28. T. Karagiannis, K. Papagiannaki and M. Faloutsos. “BLINC: multilevel traffic classification in the dark.” In ACM SIGCOMM computer communication review, vol. 35, no. 4, pp. 229-240, ACM, August 2005.
29. P. Wang, S. C. Lin and M. Luo. "A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs." In Services Computing (SCC), 2016 IEEE International Conference on, pp. 760-765. IEEE, 2016.
30. K. Xu, Z. L. Zhang and S. Bhattacharyya. “Profiling internet backbone traffic: behavior models and applications.” In ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 169-180, ACM, August 2005.
31. W. Zai-jian, Y.N. Dong, H. X. Shi, Y. Lingyun and T. Pingping. “Internet video traffic classification using QoS features.” In 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1-5, 2016.
32. L. Bin and T. Hao. “P2P Traffic Classification Using Semi-Supervised Learning.” In 2010 International Conference on Artificial Intelligence and Computational Intelligence, vol. 1, pp. 408-412, 2010.
33. D. Achunala, M Sathiyanarayanan and B. Abubakar. “Traffic classification analysis using omnet++.” In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 417-422. Springer, Singapore 2018.
34. J. Yan, X. Yun, Z. Wu, H. Luo, S. Zhang, S. Jin and Z. Zhang. “Online traffic classification based on co-training method.” In 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 391-397, 2012.
35. D. B. Shukla and G. S. Chandel. “An approach for classification of network traffic on semi-supervised data using clustering techniques”. In 2013 Nirma University International Conference on Engineering (NUiCONE, pp. 1-6. 2013.
36. H. S. Park and C. H. Jun. “A simple and fast algorithm for K-medoids clustering.” Expert systems with applications, vol. 36, no. 2, pp. 3336-41, 2009.
37. L. Wang. Support vector machines: theory and applications. Springer Science & Business Media; ed. 2005, vol. 177, ch. 1.
38. V. Vapnik, I. Guyon and T. Hastie. “Support vector machines.” Machine. Learning, vol. 20, no. 3 pp. 273-97, 1995.
39. R. Bar-Yanai, M. Langberg, D. Peleg and L. Roditty. “Realtime classification for encrypted traffic.” In International Symposium on Experimental Algorithm, pp. 373-385, Springer, Berlin, Heidelberg, 2010.
40. T. M. Cover and P. E. Hart. “Nearest neighbor pattern classification.” IEEE transactions on information theory, vol. 13 no.1, pp. 21-7, 1967.

AUTHORS PROFILE

Kate Takyi attained her BSc. Degree in Computer Science from Kwame Nkrumah University of Science and Technology, Ghana in 2009 and worked as Network Support Officer at Noble Gold Bibiani Limited from 2011 to 2013. She attained her Master’s Degree in Network Technology and Management (MSc) from Amity University Gurgaon, Haryana - India in 2016. She has worked on a project “Modular Framework for network security” and proposed a model for enhancing security for organizations with several branch offices. She is currently pursuing her PhD. Degree at Lovely Professional University, Punjab – India. Her research areas of interest include Network traffic Classification, Network security, and Network Management

Amandeep Bagga attained her Ph.D. Degree at Lovely Professional University in 2016. She is an Associate Professor and Assistant Director, Computer Applications Department in Lovely Professional University Punjab, India. Her broad area of research is Network Security, including sub area of Cryptanalysis. She has 10 years of research experience and total of 13 years teaching experience. She has 12 publications in the field of networking and security and currently working on research in crypto currency.