A novel approach for performance-based clustering and management of network traffic flows

Muna Al-Saadi[1], Bogdan V Ghita[1], Stavros Shiaeles[1], Panagiotis Sarigiannidis[2]

[1]School of Computing, Electronics and Mathematics, University of Plymouth, UK

[2]Department of Informatics and Telecommunication Engineering, University of Western Macedonia,
Kozani, Greece

{muna, bogdan, stavros }@plymouth.ac.uk, psarigiannidis@uowm.gr

***Abstract*—Management of network performance comprises numerous functions such as measuring, modelling, planning and optimising networks to ensure that they transmit traffic with the speed, capacity and reliability expected by the applications, each with different requirements for bandwidth and delay. Overall, the objective of this paper is to propose a novel mechanism to optimise the network resource allocation through supporting the routing of individual flows, by clustering them based on performance and integrating the respective clusters with an SDN scheme. In this paper we have employed a particular set of traffic features then applied data reduction and unsupervised machine learning techniques, to derive an Internet traffic performance-based clustering model. Finally, the resulting data clusters are integrated within a unified SDN architectural solution, which improves network management by finding nearly optimal flow routing, to be evaluated against a number of traffic data sources.**

Index Terms—Network performance, Clustering, Unsupervised algorithm, SDN

I. INTRODUCTION

Network performance managing includes improvement of the network function in order to maximize capacity, minimize latency and provide high reliability regardless of availability of bandwidth and failures occurrence. Management of network performance comprises many functions such as measuring, modelling, planning and optimizing networks to guarantee that they transmit traffic with the speed, capacity and re- liability, which is expected by the applications. The main factors that influence the resulting performance of traffic as it traverses a path are the packet loss, latency, and bottleneck bandwidth associated with that respective path.

In the context of application performance, Quality of Service (QoS) is the term that refers to control a process that provides network performance either at an ensured or at a differentiated level for data flows according to the application/user requirements. Due to the complexity of the network infrastructure and mix of traffic, designing a network to support QoS is, therefore, not an easy task. The fundamental step is to understand the characteristics of different types of application/network traffic. Therefore, the modelling of data traffic becomes a decisive and indispensable step. This project will propose a comprehensive characterization of traffic with respect to network performance in order to manage network efficiently by integrating machine learning and SDN. SDN is indeed the optimal vehicle to control and prioritise the respective types of traffic according to their needs, as it decouples the network devices in data plane from the traffic and its associated needs in the control plane [1], [2]. Specifically, the data plane devices such as router and switches have a packet- forwarding responsibility while the control plane includes rules that are used by the devices of data plane to forward packets. Depending on the above, SDN is characterized by decoupled control and data planes and control plane programmability [3].

The increased employment of Software Defined Networking lead to significant development of flexible network architectures, allowing the efficient movement of the data flow through the network to provide a more efficient usage of network resources. Many studies have recently tried to integrate the Machine Learning Techniques (MLT) in SDN to enhance security system of network, to improve network flow management and to develop of network design, but the previous approaches have always focused on the characteristics of the network rather than identifying the characteristics of traffic and guiding it accordingly.

The remainder of the paper is organized as follows. Section II provides an overview of the relevant work, then section III introduces the proposed research work, followed by the experimental results in section IV. Section VI outlines the future work and section VII concludes the paper.

II. RELATED WORK

In recent years, SDN became the de-facto mechanism when designing adaptable, flexible networks. As part of the ongoing efforts, a number of studies tried to integrate the Machine Learning (ML) in SDN in order to enhance security provision and improve flow management. This section will

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

provide a brief overview of these studies.

SDN is currently utilized in numerous fields, from IOT and wireless networking, to cloud computing and datacenters, focusing on both QoS and security. In [4] the authors proposed an SDN-based design for the IOT environment, allowing to achieve distinctive quality levels in heterogeneous wireless networking for IoT-related tasks; as part of the study, the authors presented the SDN controller design in IOT multi networks to provide for flexible, effective and efficient management of flows and available network resources. Furthermore, SDN remains the optimal solution to facilitate efficient management and deployment of network services. In [5] authors have proposed a novel service deployment solution for SDN environments through fog computing. The proposed solution can be used for both safety and non-safety services. Moreover, SDN centralized control also works for optimizing the resource utility and can reduce the latency by integrating Fog computing. According to the framework proposed in [6], which is an application that gathers OpenFlow traffic statistics from the controlled switches, SDN was deployed in an enterprise network, then the authors applied several machine learning methods for traffic classification. The aim of this work was to comparatively evaluate the performance of supervised classification algorithms. Similarly, a management architecture, called ATLANTIC, which performs anomaly detection, classification, or mitigation, was presented in [7]. ATLANTIC uses information theory to calculate deviations in the entropy of flow tables combined with machine learning algorithms to classify traffic flows. Consequently, the framework has the ability to categorize traffic anomalies and to block malicious flows by using the collected information. TCP parameters were also used in [8] to provide an input for detecting network attacks.

An application-aware multi path flow routing architecture, which combines Machine Learning Techniques (MLT) in Software Defined Networks (SDN) was proposed in [1]. In this framework, the controller prioritizes each flow using MLT and specifies a path depending on its classified priority. In the same context, the authors in [2] proposed a QoS-aware traffic classification architecture for SDN. The framework uses the QoS requirements to classify traffic into various classes by exploiting deep packet inspection (DPI) and semi-supervised machine learning. Network management is becoming increasingly challenging given the growth of network size, traffic volume, and the diversity of requirements in QoS; to account for this level of complexity, SDN provides flexible and scalable network management. In [9], the authors introduced policycop, a framework that provides QoS-based Service Level Agreement (SLA) using SDN environment.

Clustering traffic to provide improved network management also proved successful to a certain degree by previous studies, where classification of traffic based on user interests [10] , then applied on an SDN environment [11].

Moving further, configuring a large complex network is also a challenging job and it is becoming difficult and increasingly worrisome with the passage of time, as network administrator needs to perform sophisticated actions in order to manage network tasks, thereby, to cope up with this problem, authors in [12] proposed an event-driven network control solution based on the SDN, named Procera, to simplify various aspects related to network operation and management. The authors proposed that network operator could utilize four control domains, such as, traffic flow, data usage, time, traffic flow, and authentication.

Based on the listed studies, it is increasingly apparent that clustering of traffic and SDN are indeed likely to lead to a more effective handling of traffic; however, studies tend to look at optimizing the quality of individual flows, applications, or mixes of applications as employed by end users, without considering the characteristics shared by the flows in the first instance from a performance perspective. Indeed, flows may encounter similar levels of packet loss or delay, as well as share characteristics such as file length or application requirements.

III. PROPOSED TRAFFIC CLUSTERING AND CONTROL METHOD

In this research we have done analysis and profiling of network traffic with respect to network performance. The main objective of this analysis is to find a similarity between the traffic in few points based on network performance features. The concept behind the proposed architecture for traffic clustering and control is that flows sharing certain performance characteristics may benefit from being routed through similar paths in order to be provided with similar network characteristics, ensure they receive a fair allocation of network resources, and accommodate other flows with different requirements. In order to attain this goal, Unsupervised Machine Learning technique is applied to provide some clusters for connections, which would match a variety of other connections as shown in the Figure 1.

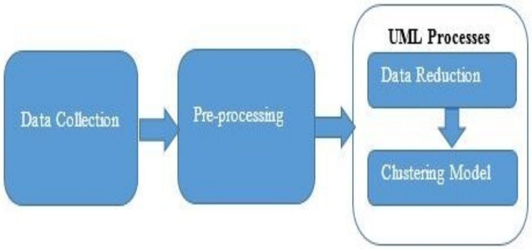

Figure 1 Proposed Clustering Scheme

Additionally, they would also possess the similar features or differences. Moreover, in clustering analysis, connections will be summarized and central cluster connections may be selected as representative for each cluster. Furthermore, in future work clustering property will be integrated to SDN controller for multi path assignment.

The proposed system comprises the following stages as a block diagram is illustrated in Figure 2.

A. Data Collection:

Raw data was collected using tcpdump, then packet traces were passed on for off-line processing.

B. Analysis and preprocessing:

The traffic traces were analysed using tcptrace [13], followed by a brief statistical processing to extract the relevant flow-based parameters

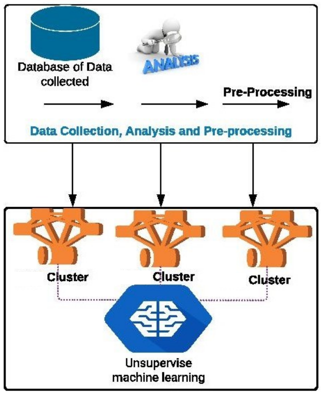
Database of Data collected
ANALYSIS
Pre-Processing
Data Collection, Analysis and Pre-processing
Cluster
Cluster
Cluster
Unsupervise machine learning

Figure 2 Proposed architecture for Clustering Network traffic

C. *Unsupervised Machine Learning*

In this study PCA method and K-means algorithm have been used, which are the most common algorithms in the research community [6], [13]. They are used for the feature selection and clustering complete connections to derive and define a number of clusters that share similar characteristics with respect to network performance.

IV. EXPERIMENTAL RESULTS

Most of network monitoring tools store the data measurements in a database for later analysis. This work encompasses network traffic collection from an offline pcap file by a linux machine by utilising tcpdump capturing tool. The network traffic was captured through one hour in laboratory of Plymouth University. Afterwards, in-depth analysis was performed on the captured data to extract features using the tcptrace tool, which has been installed on Linux Ubuntu 14.0.4. Finally, PCA and K-means Machine Learning techniques were applied for data reduction and to find similar connections.

The focus of this work is to group traffic based on the performance and network characteristics experienced. For the initial experiments, a pcap file that contains a set of 19004 connections, captured using tcpdump from a group of users the Center for Security, Communications and Network Research (CSCAN) in University of Plymouth. The captured traffic was anonymised, analysed, and filtered using tcptrace[14], which outputs information and parameters related to network performance, such as elapsed time, number of retransmissions, round trip time, window advertisements, and throughput.

A total of 146 features, representing the output of tcptrace and associated with network performance, were extracted from the header field of each TCP connection to create a dataset. Generally, raw data is incomplete, noisy and also inconsistent. While the employment of k-means needs a complete data matrix, data preprocessing can deal with these issues while performing multiple tasks such as data cleaning, normalization, and reduction. In this study, PCA was used to handle outliers and to reduce dimensions of data, while z-Score normalization technique was utilized to scale features' values that fall within a specified range to ensure that unit of feature doesn't distort the closeness of cases. In this stage, all the pre-processing steps are executed using R-based script. Principal Component Analysis (PCA) is a technique to extract key variables (in form of components) from a larger set of variables that are available in a dataset. In this study, PCA has been used as a first phase to analyze the correlation between the features for dimensional reduction in order to make the clustering algorithm more effective and efficient. The dataset of this work has over one hundred variables to describe each connection. Furthermore, there is a significant correlation between these variables. Figure.3. depicts the correlation of the first ten variables in a dataset. In this figure correlation coefficients has been colored according to the value.

Table I, depicts the first 13 components and their associated eigenvalues, proportion of variance and cumulative variance. In the original space, each of the 129 features contained approximately 0.77% (1/129) of the total variance. The principal components that are selected should explain at least 0.77% of the total variance. Since PCA is data reduction method, therefore, it is significant to retain a suitable number of components based on trade-off between the completeness and simplicity. In this work, different numbers of components have been chosen.

Table I Components and Their Associated Eigenvalues

Component	Eigenvalues	Variance%	Cumulative of variance %
Comp1	18.502	14.34	14.34
Comp2	13.443	10.42	24.77
Comp3	8.175	0.06	31.10
Comp4	7.551	0.05	36.96
Comp5	5.059	0.03	40.88
Comp6	4.253	3.29	44.18
Comp7	4.130	3.20	47.38
Comp8	3.626	2.81	50.19
Comp9	3.126	2.42	52.62
Comp10	2.972	2.30	54.92
Comp11	2.874	2.22	57.15
Comp12	2.587	2.00	59.15
Comp13	2.479	1.92	61.08

A. *Clustering Model*

Cluster analysis gathers data, which is based only on the information found in the data that depicts the subjects and their relationships. The aim is that the objects within a group should be the same or at least appear closer to one another and should be different or distinguishable from the objects placed in other groups. The best clustering is gained when the similarity within a group is great and the difference between groups is great. This step is the integral part of our proposed model where we have quite distinct performance clusters, which are constructed from the flows. SDN will depend on these performance clusters in order to flow routing decision.

K-means Clustering: High dimensional data contains hundreds or thousands of features, but not all of these features are relevant or somehow vital for the goal function. Therefore, if these features are extracted from the data, it will not affect the outcome of the goal function. Furthermore, high-dimensional data may contain redundant features, which means that numerous features may have same effectiveness on the objective function results. To minimize the workload computation of high-dimensional data; the obsolete features can be represented by one feature. In this work, two phases have been implemented as the following:

PHASE 1: K-means, which is used with unlabeled data (i.e. categories or groups of data are not defined), has been utilized for clustering and because of using more than one hundred of features, principal component analysis (PCA) has been used for dimensional reduction. In this part of work, finding an optimum 'k' value was carried out using the Elbow method, which limits the number of clusters to a value beyond that appending another cluster does not improve the modeling of data.

PHASE 2: K-means has been used as feature selection technique, aiming at output minimized dataset, which only contains the essential features and diminishes the size of high dimensional data. Therefore, the output of this stage is a vector of 52 features, which will accelerate the clustering process, which implemented by K-means also as a clustering technique. Moreover, PCA technique has been used for data reduction.

Two feature sets were used in the above phases: feature set1 included 129 features generated from *tcptrace* and statistical analysis, and feature set2 had 77 features removed from feature set1, which represents the control (driven) features for each cluster, to minimize the total number of features; therefore, the new set of features, which is a subset of set1, contains 52 features as shown in Table II.

Table II Feature Set 2

Seq	Feature	Seq	Feature
1.	first packet	2.	avg owin a2b
3.	total packets a2b	4.	wavg owin a2b
5.	total packets b2a	6.	wavg owin b2a
7.	ack pkts sent a2b	8.	initial window bytes b2a
9.	ack pkts sent b2a	10.	ttl stream length a2b
11.	pure acks sent a2b	12.	ttl stream length b2a
13.	sack pkts sent a2b	14.	throughput b2a
15.	sack pkts sent b2a	16.	RTT samples a2b
17.	dsack pkts sent a2b	18.	RTT samples b2a
19.	unique bytes sent a2b	20.	RTT min a2b
21.	unique bytes sent b2a	22.	RTT min b2a
23.	actual data pkts a2b	24.	RTT max a2b
25.	actual data pkts b2a	26.	RTT max b2a
27.	actual data bytes a2b	28.	RTT avg a2b
29.	actual data bytes b2a	30.	RTT avg b2a
31.	rexmt data pkts a2b	32.	RTT stdev a2b
33.	rexmt data pkts b2a	34.	RTT stdev b2a
35.	rexmt data bytes a2b	36.	post.loss acks a2b
37.	rexmt data bytes b2a	38.	post.loss acks b2a
39.	outoforder pkts a2b	40.	ambiguous acks a2b
41.	outoforder pkts b2a	42.	ambiguous acks b2a
43.	sacks sent a2b	44.	segs cum acked a2b
45.	sacks sent b2a	46.	segs cum acked b2a
47.	avg win adv b2a	48.	duplicate acks a2b
49.	max owin a2b	50.	duplicate acks b2a
51.	max owin b2a	52.	triple dupacks b2a.

Convergence of Clusters: A good cluster analysis is achieved when all clusters have population between 5% and 30% of the overall dataset. Since, the dataset that is used in this work contained 11593 connections, the minimum and maximum number of connections in any cluster should be between 500 and 4000. If the size of a cluster is beyond these limits, it will be considered as outlier and it needs to be handled. A few outliers can be easily incorporated by cluster structure.

Profiling of the Clusters: After validating the convergence of clusters, it is essential to identify behavior of each cluster. Cluster identification is achieved by mapping combination of variables with respect to network performance, such as packet loss, delay, connection size, throughput and congestion window; the output of this stage will be the representative connection for each cluster.

B. K-means Clustering Analysis

1) Analysis of using K-means Clustering Only

Only K-means algorithm has been employed to cluster traffic information with K equal 3 and 100, as depicted in Table III. Two different sets of features are used to cluster traffic in order to yield an optimal clustering.

Accuracy: Each object is included to the closest cluster and then Euclidean distance is used to calculate the distance between the object and the cluster center. Each cluster center will be updated as the mean for objects in each cluster.

The within-sum of squares is:

$$\sum_{k=1}^{K}\sum_{i\in S_k}\sum_{j=1}^{p}\left(x_{ij}-\bar{x}_{kj}\right)^2 \qquad (2)$$

The process is iteratively repeated until either it reaches the maximum number of iterations or the change of within-cluster sum of squares in two successive iterations is less than the threshold value.

As in Table III, the accuracy of the clustering, as determined for the examined dataset, is 92.9%. The accuracy of each features' sets is shown in Table III. The recorded accuracy for feature set1 is considerably low as compared to feature set2 for both numbers of clusters, although, the number of features are more in set1. As shown in Table III the accuracy in this particular case is low. For feature set 2, comprising of reduced number of features the results illustrate a significant improvement in accuracy, yielding 43.2% and 92.9%.

Table III Accuracy of K-Means with Features Sets

Feature set	Accuracy of K=3	Accuracy of K=100
Features set1	15.2%	80.7%
Features set2	43.2%	92.9%

For K=3, the clustering lead to a poor spread of connections among clusters. One of the three clusters (K=3) had just 12 connections out of 11593 and 12 clusters of one hundred clusters (K=100) have one connection. This goes against the safe range for each cluster, which should contain between 5% and 30% of the overall dataset. The reason for having this problem is the existence of outliers, which will be discussed in next section.

2) Analysis of using K-means Clustering with PCA

Principal component analysis is data reduction method, it is significant to retain suitable number of factors on the basis of keeping a balance between retaining as few as possible factors (simplicity) and explaining most of the variation in the data (completeness). The Kaiser's rule recommends only factors with eigenvalues exceeding unity should be retained. Intuitively, this means that any retained factor should compute at least as much variation as any of the original variables [15].

In this step of work, PCA is used for dimensional reduction before applying clustering algorithm with five components. The findings as in Table V show that when applying K-means on features set 1, with three clusters and five components of PCA, the accuracy has increased significantly, which is 36.7% comparing with the findings of using K-means. However, when the number of clusters is 100, the accuracy has increased to 97.6% and the clusters which have only one connection have been eliminated. This keeps all resultant clusters in the safe range, which should contain between 5% and 30% of the overall dataset. While, with features in set 2, the result shows that, K-means with three clusters and five components of PCA

(5 PCs) provide the better accuracy, which is 60.6% when compared with the accuracy of using K-means only. This result has changed to the best level with 100 clusters, where the accuracy becomes 99%.

Table IV Accuracy of K-Means with PCA

Feature set	Accuracy of K=3 & 5 PCs.	Accuracy of K=100 & 5 PCs.
Features set1	36.7%	97.6%
Features set2	60.6%	99%

V. LIMITATIONS

The size of dataset is an important aspect of machine learning for training, testing and validation. So, dataset with larger size will be used in the future work. Detection and treatment of outliers is of major significance for Machine Learning where the quality of data is as important as the quality of a prediction or classification model. In this work, it was not possible to completely avoid outliers existence that cause presence one of the clusters with size beyond the limits of clusters convergence, which was between 500-4000 connections for each cluster. Moreover, with PCA, it is significant to retain suitable number of factors on the basis of keeping a balance between retaining as few as possible factors simplicity and completeness, however, in this study, the choice of accuracy as an aim for clustering leads to be restricted to choose just five components of PCA in spite of they cover 40% of data.

VI. FUTURE WORK

This session discusses our future work and framework consisting of SDN and focus will be on the SDN sub-controller module.

Sub-Controller Module

Sub controller module will be built in SDN controller. It includes three modules:

- Machine learning algorithm: This stage includes two phrases:
  - Feature Selection & Data Reduction: This has been executed by implementing PCA and K-means.
  - Classification, which has been implemented by using K- means or SOM clustering method to get the clusters as an output.
- Calculates cost of link module: in this module, available bandwidth and latency of each link will be calculated for path updating.
- Discovery & Selection of path module: based on the representative connection characteristics, available paths will be computed and then appropriate path is chosen for each cluster.

The last two modules will be implemented by SDN controller according to the output of clustering.

Implementation of Module

In this study, PCA method and K-means algorithm, were used for the feature selection and clustering complete connections. The two data sets of features, which are explained in clustering model section, will be used. Moreover, cluster characterization will be the output of this step. Characterization process describes each cluster based on a combination of features with respect to network performance. Furthermore, in future we aim to explore more unsupervised machine learning algorithms along with the python script language for clustering.

The second step includes Cost of Link Calculation that has three modules, Latency computation module that compute latency uses all links between switches and second is, bandwidth assessment module, which is responsible to find available bandwidth for all the links. The last module links cost computation module that computes the cost of all links once the latency and available bandwidth are computed.

In Selection of Path part, the SDN controller interacts with UML module. For each connection in data set, SDN controller captures the vector of features and forwards it to UML module to get the corresponding cluster. Based on the cluster reply from UML module, SDN controller computes best N available paths from source to destination and chooses one suitable path for the connection. A connection will be selected according to the cost of the path and feature vector of connection, which is sent through that path. Once the path is chosen, proposed controller will send the corresponding rules of connection in all the switches of that path. SDN controller is responsible for the routing of the packets to relevant SDN switches. It is assumed in this research that network are assumed to have SDN aware switches. The steps involved in flow of a packet in the network are as follows:

- A packet is sent by the client/host to one of the switches, which we assume, in this research, that they are SDN aware switches.
- When SDN switch will receive a packet from the host it will check it first in flow table in order to find the matching flow rule. If switch will be successful in finding flow rule in flow table, it will forward the packet according to flow rule.
- If switch will not be able to find any flow rule for respective packet in the flow table of SDN switch, the switch will send a PACKET_FOR_SC message to sub controller.
- The sub controller will receive the PACKET_FOR_SC message and will compare the characteristics of flow against the performance clusters. Moreover MOD_FLOW messages will be send to the SDN switches with hard timeout of 't'.
- The sub controller will explore the characteristics from the PACKET_FOR_SC and will send it to unsupervised machine learning clustering algorithm (LC) in sub controller.
- Unsupervised machine learning clustering algorithm LC on exploring the feature vector from sub controller will compare it with one of the predefined clusters.
- The Sub-Controller on receiving the relevant cluster label from LC will send this information to SDN controller.
- As in our proposed method, SDN controller will possess capability to calculate the cost of link and optimal path selection, so it will perform both the operations.
- As SDN controller contains the information about the all possible network paths so it is important to compute the best optimal path which will be calculated by computing cost of link and path

selection and it will route the packet to relevant SDN switch.

Algorithm 1 Proposed Algorithm for Network Management optimization using SDN

```
INPUT: A packet ’Pf’ is sent by the client/host ’C’ to one
of the SDN aware switches ’Sn’
OUTPUT: SDN controller compute the best optimal path
which will be calculated by computing cost of link ’Cl’ and
path selection ’Ps’ and it will route the packet to relevant SDN
switch ’Sn’. Finding the relevant flow rule in flow table 1
for all The packets ’Pf ’ in ’Tf’ do
    forward ’Pf’ according to ’Tf’
end for
for all other packets ’Pf'' do
if The packets ’Pf ’ NOT in ’Tf’ then
      find ’CL’ such that ’Pf’ in cluster1 , OR ’Pf’ in
      cluster2, OR ’Pf’ in cluster3 AND Computer ’Ps’ and
      ’Cl’
end if
end for
```

Table V Table of Notations and Symbols

Name	Symbols
Flows in packets	’Pf’
Client/hosts	’C’
SDN Switches	’Sn’
Flow table	’Tf’
Packet for sub controller.	PACKET_FOR_SC
Mode flow	MOD_FLOW
Learning Cluster	LC
cost of link	’Cl’
path selection	’Ps’

Furthermore, in future we aim to explore more unsupervised machine learning algorithms along with the python script language for clustering.

VII. CONCLUSION

In this research, we have proposed the scheme which is based on the machine learning. A K-means algorithm was used for feature selection and clustering the traffic based on the network performance feature vector, as processed by tcptrace, followed by PCA analysis to improve the encompassing set of parameters. It is noteworthy that the accuracy of the clustering process varied according to the use of K-means only and K-means with PCA. Moreover, in future SDN based network will be used for optimal assignment of the path.

References

[1] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, “AMPS: Application aware multipath flow routing using machine learning in SDN,” *2017 23rd Natl. Conf. Commun. NCC 2017*, 2017.

[2] P. Wang, S. C. Lin, and M. Luo, “A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs,” *Proc. - 2016 IEEE Int. Conf. Serv. Comput. SCC 2016*, pp. 760–765, 2016.

[3] J. Chen, X. Zheng, and C. Rong, “Survey on software-defined networking,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9106, no. 1, pp. 115–124, 2015.

[4] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “PolicyCop: An autonomic QoS policy enforcement framework for software defined networks,” *SDN4FNS 2013 - 2013 Work. Softw. Defin. Networks Futur. Networks Serv.*, 2013.

[5] H. Singh, “Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification,” *2015 Fifth Int. Conf. Adv. Comput. Commun. Technol.*, pp. 401–404, 2015.

[6] G. LIN, Y. XIN, X. NIU, and H. JIANG, “Network traffic classification based on semi-supervised clustering,” *J. China Univ. Posts Telecommun.*, vol. 17, pp. 84–88, Dec. 2010.

[7] M. Zulfadhilah, Yudi Prayudi, and I. Riadi, “Cyber Profiling using Log Analysis and K-Means Clustering A Case Study Higher Education in Indonesia,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 7, pp. 430–435, 2016.

[8] M. Siracusano, S. Shiaeles, and B. Ghita, “Detection of LDDoS Attacks Based on TCP Connection Parameters,” in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–6.

[9] D. P. Ismi, S. Panchoo, and M. Murinto, “K-means clustering based filter feature selection on high dimensional data,” *Int. J. Adv. Intell. Informatics*, vol. 2, no. 1, p. 38, 2016.

[10] T. Bakhshi and B. Ghita, “User traffic profiling,” in *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 91–97.

[11] T. Bakhshi and B. Ghita, “OpenFlow-enabled user traffic profiling in campus software defined networks,” in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016, pp. 1–8.

[12] V. Kumar, H. Chauhan, and D. Panwar, “K-Means Clustering Approach to Analyze NSL-KDD Intrusion Detection Dataset,” *Int. J. Soft Comput. Eng.*, vol. 3, no. 4, pp. 1–4, 2013.

[13] I. H. Witten, E. Frank, and M. A. Hall, “Embedded Machine Learning,” *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, pp. 531–538, 2011.

[14] Shawn Ostermann, “tcptrace.” [Online]. Available: http://www.tcptrace.org/. [Accessed: 01-Oct-2018].

[15] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, “Machine Learning in Software Defined Networks: Data collection and traffic classification,” *2016 IEEE 24th Int. Conf. Netw. Protoc.*, no. NetworkML, pp. 1–5, 2016.