Network Traffic Classification using Machine Learning for Software Defined Networks

Perera Menuka, Kandaraj Piamrat, Salima Hamma

Network Traffic Classification using Machine Learning for Software Defined Networks

J.K.Menuka Perera*, Kandaraj Piamrat, and Salima Hamma

LS2N/University of Nantes
2 Chemin de la Houssiniere,
BP 92208, 44322 Nantes Cedex 3, France
*jkmenukaperera@gmail.com
{firstname.lastname}@univ-nantes.fr

Abstract. The recent development in industry automation and connected devices made a huge demand for network resources. Traditional networks are becoming less effective to handle this large number of traffic generated by these technologies. At the same time, Software defined networking (SDN) introduced a programmable and scalable networking solution that enables Machine Learning (ML) applications to automate networks. Issues with traditional methods to classify network traffic and allocate resources can be solved by this SDN solution. Network data gathered by the SDN controller will allow data analytics methods to analyze and apply machine learning models to customize the network management. This paper has focused on analyzing network data and implement a network traffic classification solution using machine learning and integrate the model in software-defined networking platform.

Keywords: Machine Learning · Classification · Network Traffic · Software Defined Networking

1 Introduction

Recent advances in software defined networking and machine learning techniques have created a new era of network management. This new concept has combined network intelligence and network programmability to create autonomous high performing networking, which will expand 5G (5^{th} Generation) capabilities. With the recent improvements in Internet of Things (IoT), cloud computing self-driving vehicles, etc., the demand for bandwidth consumption has increased exponentially and pushed network operators the ability to search for new concepts of network management.

Software defined networks provide a programmable, scalable and highly available network solution. This solution separates the control plane and the data plane from the network devices and logically centralized the controlling component. The centralized controller has a global view of the network and enables the network operator to program their policies rather than depending on network equipment vendors.

For the past decades, Artificial Intelligence (AI) and Machine Learning (ML) concepts were developed for different use cases with different approaches. The latest concept of AI/ML technologies are developed based on statistics. Integrating these tools into the networking industry will enable network operators to implement self-configuring, self-healing, and self-optimizing networks. We can name this type of network as Knowledge Defined Networks (KDN) as mentioned in [1].

This new concept of intelligent and programmable network is an end-to-end network management solution. It is important to manage existing network resources efficiently. Even the number of users connected to the network is increasing, not all users required the same amount of network resources. Identifying each user's demand and behavior on the network will enable the operator to manage network resources much more efficiently.

In a network, there are two basic types of traffic flows: elephant flows and mice flows. Elephant flows are referred to as heavy traffic flows and mice flows are referred to as light traffic flows. And typically the resource allocation process for these flows are standard. This approach of resource allocation is a waste of network resources and allocating the same amount of resources for both flows is not an optimum solution. There are currently few methods to identify network traffic but the recent technological advancements made these concepts inefficient. Port-based classification is one of the methods that classifies network traffic based on port numbers extracted from packet header, which allow to understanding the traffic behavior and the type of applications having been used. But nowadays, modern applications use dynamic ports or tunneling, which makes this method ineffective. In Payload-based classification method, network traffic is classifying by inspecting packet payload. But this method requires a high level of computing power and storage, which will increase the cost. Another issue with this method is the privacy laws and data encryption.

When it comes to network traffic classification, ML algorithms depend on a large number of network features. And software defined networking will enable ML algorithms to control the network and can become automatic resource allocation process. Therefore, in this study, ML-based traffic classification solution was introduced for SDN. The proposed architecture uses existing network statistics and an offline process for understanding network traffic patterns with a clustering algorithm. For the online process, a classification model is used to classify incoming network traffic in real-time.

The rest of the paper will be presented as follows: Section 2 discusses on related work of similar researches on network traffic classification. Section 3 describes the proposed system architecture. Section 4 presents the experimental result of the system and Section 5 concludes this paper.

2 Related Work

In the paper [2], the authors have used the ML algorithm for classifying network traffic by application. They have trained few ML models using labeled data by

applications such as Post Office Protocol 3 (POP3), Skype, Torrent, Domain Name System (DNS), Telnet were recognized by the classifier. For this experiment, they have tested six different classification models and compared accuracy. AdaBoost, C4.5, Random Forest, Multi-layer Perceptron (MLP), Radial Biased Function (RBF), Support Vector Machine (SVM) are the classifiers used for this research. They have concluded that Random Forest and C4.5 classifiers give better accuracy than the other models.

Authors of [3] have experimented with mobile network traffic classification ML models. In their project, there are three main objectives. Comparing the accuracy of three classification models [SVM, Multi-Layer Perceptron with Weight Decay (MLPWD), MLP]. Analyzing the effect on accuracy by varying the size of the sliding window. Comparing the accuracy of predictions of the models for unidimensional /multi-dimensional datasets. In their project, they have selected 24 features and selected one of the feature as the target to predict. In terms of accuracy, the paper has concluded that in multi-dimensional data sets SVM performs better and in unidimensional data sets, the MLPWD model performs better.

In the paper [4] they have experimented with the data collection and traffic classification process in software defined networks. In their work, they have developed a network application to collect OpenFlow data in a controlled environment. Only Transmission Control Protocol (TCP) traffic was considered for this project. Several packets of information were gathered using different methods. For example, Packet_IN messages were used to extract source/destination IP addresses and port addresses. First five packet sizes and timestamps were collected from the controller since in this experiment the next five packets after the initial handshake between server and client flow through the controller. Flow duration was collected by subtracting the timestamp of the initial packet and the time stamp of the message received by the controller regarding the removal of the flow entry. To avoid the high variance of the data set, they have used a scaling process named standard score. They have also mentioned that highly correlated features are not contributing much to the algorithm but increase the complexity in computation. They have used the Principle Component Analysis (PCA) algorithm to remove these high correlated factors. Random Forest, Stochastic Gradient Boost, Extreme Gradient Boost are the classifiers used in their research. The results were compared by evaluating the accuracy of each label.

In the paper [5] discussed ML-based network traffic classification. Their motivation for this project is to optimize resource allocation and network management using ML based solution. According to the paper, there are four levels of resource allocation, which are spectrum level, network level, infrastructure level, and flow level. In their paper, they have tested classifying network traffic by applications and they have used support vector machine and Kmeans clustering algorithm. The data set contains 248 features and manually labeled. The traffic labels were www, mail, bulk, service, p2p, database, multimedia, attack, interactive and games. In the SVM model, they have used four kernels namely

linear, polynomial, RBF and sigmoid. And evaluated its performance using the following parameters: accuracy, recall, precision. Considering overall accuracy, the RBF kernel of SVM outperforms other kernels. They have also tested the classification accuracy by varying the number of features. And accuracy is higher with a maximum 13 selected features. In the Kmeans clustering algorithm, they have used the unlabeled data with a predetermined number of clusters. They have compared results with supervised and unsupervised models and according to the paper, SVM has the highest precision and overall accuracy.

Authors of [6] have discussed and concepts of SDN, Network Function Virtualization(NFV), Machine learning, and big data driven network slicing for 5G. In their work, they have proposed an architecture to classify network traffic and used those decisions for network slicing. According to the paper, with the exponentially increasing number of applications entering the network is impossible to classify traffic by a single classification model. So they have used the Kmean clustering algorithm to cop this issue. By using this unsupervised algorithm, they have grouped the data set and labeled them. They have set the number of clusters k=3 associating three bandwidths. With this grouping and labeling, they have trained five classification models: Navie Bayes, SVM, Neural networks, Tree ensemble, Random Forest. And compared its accuracies. The results show that Tree ensemble and Random forest perform with the same accuracy. Depend on the ML output, bandwidth was assigned in the SDN network applications. They have ed this system by streaming YouTube a video before the classification process and check the quality of the video. And compared it with the quality of the video after the classification and bandwidth allocation.

In this study, the number of features was selected based on keeping the compatibility with the implementation (SDN controller) and avoid complexity and heavy computations in the network application. An unsupervised learning algorithm was used to identify the optimum number of network traffic classes rather than selecting a predefined number of network traffic classes, which makes this method a more customized network traffic classification solution for network operators.

3 Proposed Solution

This proposed solution was divided into two sections. One of the sections was to train the machine learning algorithm and the other section was to create a network experiment to run the trained ML model on an SDN platform as a proof of concept. In the first section, a related dataset was selected, cleaned and prepared for ML models. An unsupervised ML algorithm is applied to cluster and label the dataset then we used that dataset to trained multiple classification models. In the second section, the SDN bed was implemented, a network application containing the trained ML model was created and deployed to the network for real-time classification.

3.1 ML Model training

For this paper, "IP Network Traffic Flows, Labeled with 75 Apps" dataset from Kaggle [7] database was used. This dataset was a perfect match for our objectives and satisfy all the three main components of a good dataset, which are real-world, substantial and diverse. This dataset was created by collecting network data from Universidad Del Cauca, Popayn, Colombia using multiple packet capturing tools and data extracting tools. This dataset is consisting of 3,577,296 instances and 87 features and originally designed for application classification. But for this work, only a fraction of this dataset is needed. Each row represents a traffic flow from a source to a destination and each column represents features of the traffic data.

1) Data Preparation - As mentioned above only a few features were used for this research. The most important factors that concerned when selecting features were relatability to the research objective and easily accessed by the controller without using tools or other network applications to reduce high computations. Selected features as follows: Source and destination MAC addresses and port addresses, flow duration, flow byte count, flow packet count, and average packet size. In the data cleaning process, several operations need to be done before it is ready for machine learning model training. If there are duplicate instances in the dataset, it will cause bias in the machine learning algorithm. So to avoid the biasing, those duplicates need to be identified and remove from the dataset. Moreover, some ML models cannot handle missing data entries. In that case, rows with missing data have to remove from the dataset or fill them with the values close to the mean of that feature. In this dataset, there are several features contains different data types. But some ML models can only work with numeric values. To use those data types for the ML model training, it is necessary to convert or reassign numeric values to represent its correlations with other features. Next, Min/Max normalization was used to normalize features with high variance.

2) Data Clustering - Even though the data was clean enough to train ML models, data was not labeled. Classification process is a supervised learning algorithm that need labeled data for the training process. Understanding the traffic patterns in the dataset is a complicated and time-consuming task. Since the dataset is very large, it is very hard to label traffic flows manually. To avoid manual labeling, an unsupervised learning model can be used. By using an unsupervised learning algorithm, network traffic data will be clustered based on all the possible correlations of network traffic data. For this process, Kmeans unsupervised learning model was used as shown in Figure 1. It is a high accuracy, fast learning model ideal for large datasets. The number of clusters will be selected using the Davies-Bouldin algorithm [8]. This method is calculating distances of clusters by using Euclidean distances and lower the score better the cluster in terms of similarity ratio of within-cluster and between cluster distances. By selecting k value with the lowest Davies-Bouldin score, Dataset was clustered and labeled.

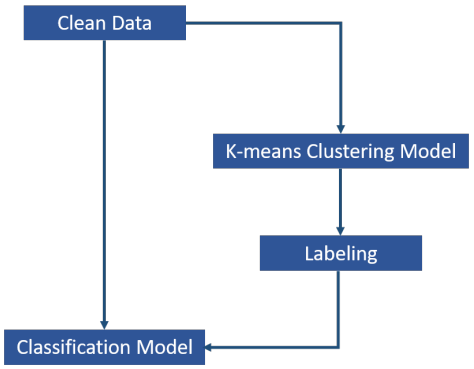


Fig. 1: Labeling dataset using Kmeans clustering Algorithm

3) Classification - Next, labeled data was used to train classification models. There are multiple classification models available and each and every model classify data with different mathematical models. Therefore, results of each model could be different from each other. Some models could perform better and some models perform poorly. In other word, it is better to train and test multiple classification models to find out which model fit better for the project. The tested models are briefly described below.

- **Support Vector Machine (SVM)** algorithm is a supervised learning algorithm that uses labeled data to train the model. SVM model will calculate decision boundaries between labeled data also known as hyper planes. And points near these hyper-planes are called extreme points. The algorithm will optimize these decision boundaries by setting up margins that separate hyper-planes. Several kernels that uses to optimize these decision boundaries. Linear, RBF, Polynomial and Sigmoid are the most commonly used kernels. Real-world data can be one dimensional or multidimensional. And these data sets are not always linear separable. The linear kernel can handle datasets that can linear separable and for nonlinear datasets, can use other kernels that can transform nonlinear datasets into linear datasets and classify. SVM is effective in multi-dimensional datasets and it is a memory-efficient model.

- **Decision Tree** is another supervised learning model that classifies data based on information gains by calculating the entropy of the dataset. It is a graphical representation of all the conditions and decisions of the dataset. The root node will be calculated using entropy with the highest information gain among the dataset. This process will continue to split branches and complete the tree. Each internal node is a test on attribute and branches represent the outcome. Leaf represents a class label. The decision tree can use numeric and categorical data for the classification problems. It also supports nonlinear relationships between features.

– **Random Forest** is one of the powerful supervised learning algorithm, which can perform both regression and classification problems. This is a combination of multiple decision tree algorithms and higher the number of trees, higher the accuracy. It works as same as the decision tree, which based on information gain. In classification, each decision tree will classify the same problem and the overall decision will be calculated by considering the majority vote of the results. The most important advantage of this model is that it can handle missing values and able to handle large datasets.

– *Kth* **Nearest Neighbor** or KNN is an instance based supervised learning algorithm. In the KNN model, the value k represents the number of neighbors needs to consider for the classification. The model will check the labels of those neighbors and select the label of the majority. The value k should be an odd number to avoid drawing the decision. It is a robust model that can work with noisy data and perform better if the training data set is large. However, it is not performing well in multidimensional datasets and could reduce efficiency, accuracy, etc.

3.2 Network application development

For the simulation testbed, a simple virtual network was created on Mininet [12] network emulator with five hosts, one OpenFlow [13] enabled open vSwitch and one SDN controller (RYU) [14]. For the simplicity of this research, tree topology was used as shown in Figure 2. There are two other network applications that need to be installed, which are simple_switch and ofctlrest. These applications will allow the controller to switch packets within the network and enable REST API calls. This switching application manages to install flow rules on the flow tables based on source, destination and flow information. These flow tables are the source of information for the classification application.

Table 1: System Configurations

System OS	Ubuntu(18.10)
SDN Controller	RYU(4.30)
Switches	Open vSwitch(2.11)
Network Emulator	Mininet(2.2.2)

This network traffic classification application is the program that contains the trained machine learning model. It is a python based program and communicates with the SDN controller via REST API calls. It is also responsible for extracting data from the controller, cleans, normalize and feed the ML model. The model will classify traffic flows each time when the program runs.

In this paper, traffic has to be generated artificially. To generate traffic, the tool D-ITG [15] as been used. In this tool, various parameters can be modified to

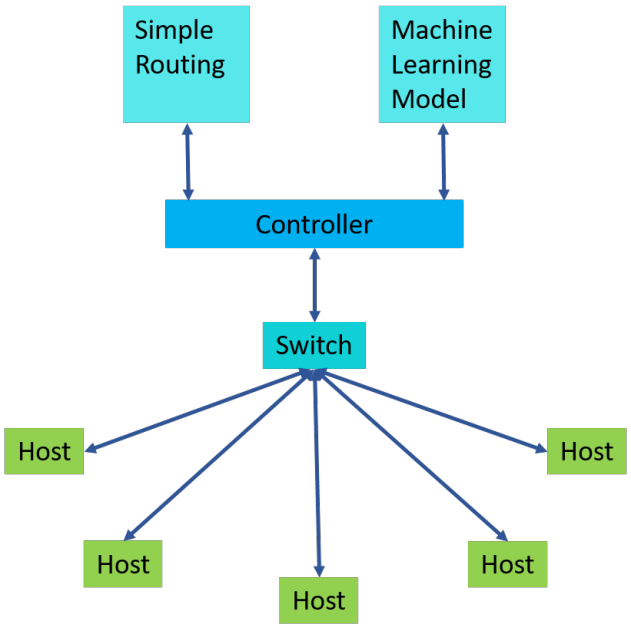


Fig. 2: SDN Testing platform

mimic real-world network traffic. Bandwidth, window size, packet size are some of them. There are also CBR (Constant Bit Rate) and VBR (Variable Bit Rate) options available within this tool. For this experiment, multiple traffic flows were generated between hosts to evaluate the machine learning output and compare it with its traffic flow characteristics.

4 Performance Evaluation

4.1 Kmeans Clustering

In the Kmeans clustering results, the number of clusters (k value) will be varied from 2 to 15 and calculate the Davies-Bouldin score for each k value. From Figure 3, k=4 has the lowest Davies-Bouldin score, which reflects that there are four types of traffic behaviors that can be identified from this dataset.

The four types of network traffic behaviors recognized by the Kmeans algorithm were analyzed for understanding their characteristics. However, they are not clearly specific to typical traffic classes that we encounter on the internet. Therefore, in order to better define each cluster, more features need to be added to refine the clusters. This needs to be done in the future work. Nevertheless, for this research, ranges from features of each cluster are sufficient to continue with the classification process.

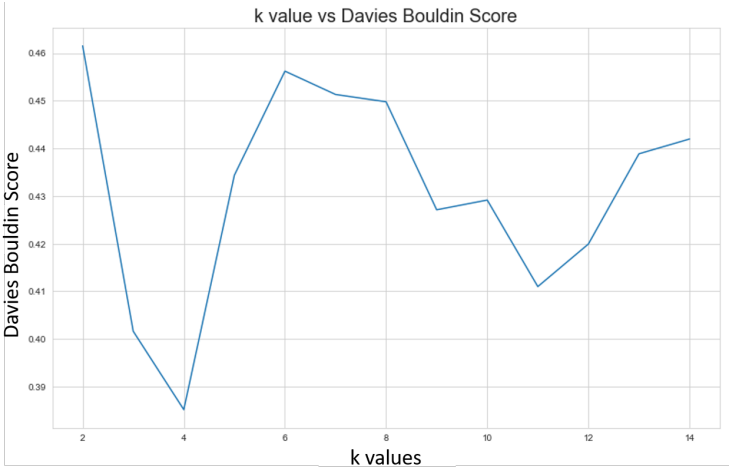


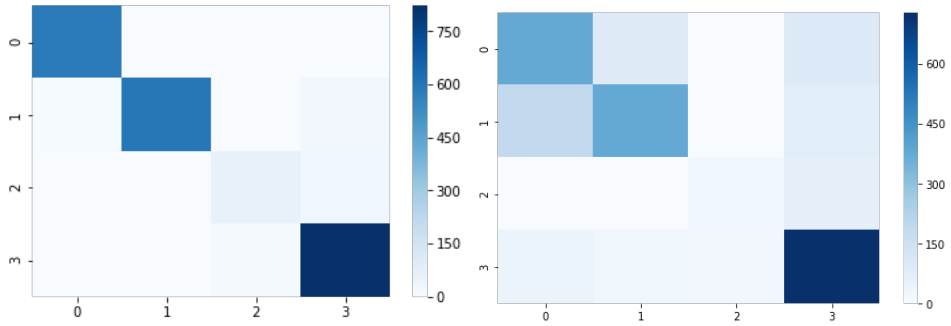
Fig. 3: k value vs Davies Bouldin Score

4.2 Network traffic classification

Using the labeled dataset from the above clustering algorithm, five supervised learning models were trained and evaluated. The labeled dataset was divided into two parts as training dataset and testing dataset with 70% to 30% ratio. All the models were trained using the training dataset separately and as shown in Table 2, model accuracies were calculated using the testing dataset. All the classification models were further analyzed using confusion matrices to checking the cluster accuracies and Figure 4 shows the results for each model. From the confusion matrices, it is clear that SVM linear model has the most accurate clusters. Decision Tree and Random Forest models have failed to classify cluster No.2 correctly even though those have classified other clusters correctly. With the highest overall accuracies and high cluster accuracies, SVM linear model was selected for the network application.

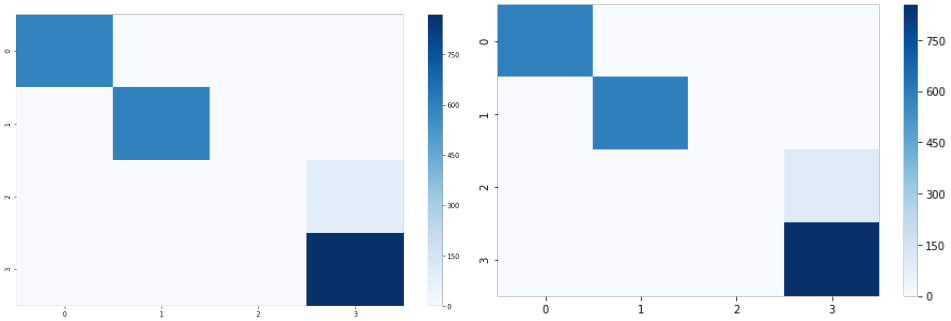
Table 2: Classification model accuracies

Model	**Accuracy**
SVM (Linear)	96.37%
SVM (RBF)	70.40%
Decition Tree	95.76%
Random Forest	94.92%
KNN	71.47%



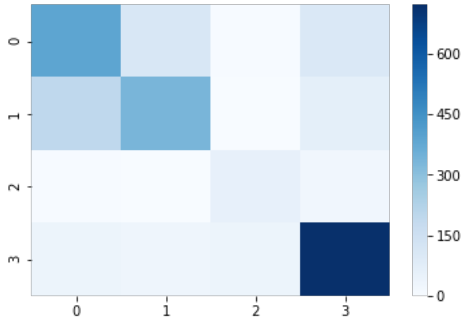
(a) Support vector machine (Linear)

(b) Support vector machine (RBF)



(c) Decision tree

(d) Random forest



(e) K^{th} nearest neighbor

Fig. 4: Confusion matrices of classification models

4.3 Network performance

The trained classification model was integrated with the network application and evaluated the real-time network traffic classification by generating network traffic in the testbed using D-ITG tool. For this evaluation, 50 traffic flows were generated considering cluster characteristics identified by the clustering algorithm. Generated traffic were compared with its characteristics and classi-

fication outputs. Figure 5 shows the percentages of accurate classifications by cluster number. These results shows that even though the network application can classify three clusters with high accuracy (100%), it has some confusions to classify cluster No.2 (96.50%) as recognized before by the confusion matrix.

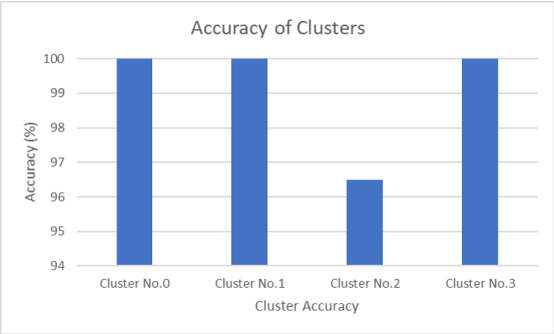


Fig. 5: Accuracy of clusters

5 Conclusion

This work has been carried out as a proof of concept while combining machine learning with software defined networking, in particular, for network traffic classification. It can be seen that traffic classification using machine learning algorithms provides good results within SDN environment. This is possible thanks to the ability of collecting information in this type of architecture. It is clear that this is a promising solution. In the near future, these high performing, intelligence-based networking concepts will enhance or even replace conventional networking management.

For the future work, several issues have to be addressed. First, the proposition was tested only on a simple topology and mainly focused on ML model accuracy. But in the real world, the networks are much more complicated and accuracy is not enough. There are other factors such as scalability, availability, etc., which directly effect the performance of a real-world network. Furthermore, the four traffic pattern detected by the clustering algorithm needs to be refined while keeping complexity reasonable when increasing number of features. This result is also context-dependent because user behavior patterns are different from network to another. For example, the number of clusters in a data center dataset would be different from the number of clusters in a sensor network dataset. Finally, for the classification, only five models were trained and compared. However, there might be another classification model that can be a better fit for this type of classification problem.

6 Acknowledgement

This work is a part of CloudIoT project, funded by Atlanstic 2020 programme, which is supported by the Pays de la Loire region and the cities of Nantes, Angers and Le Mans.

References

1. A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcn, M. Sol, V. Munts-Mulero, D. Meyer, S. Barkai, M. J. Hibbett, G. Estrada, K. Ma'ruf, F. Coras, V. Ermagan, H. Latapie, C. Cassar, J. Evans, F. Maino, J. Walrand, and A. Cabellos, "Knowledge-Defined Networking," SIGCOMM Comput. Commun. Rev. 47, 3 (September 2017), pp. 2-10. DOI: https://doi.org/https://doi.org/10.1145/3138808.3138810
2. R. C. Jaiswal and S. D. Lokhande, "Machine learning based internet traffic recognition with statistical approach," 2013 Annual IEEE India Conference (INDICON), Mumbai, 2013, pp. 1-6. DOI: https://doi.org/10.1109/INDCON.2013.6726074 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6726074&isnumber=6725842
3. A. Y. Nikravesh, S. A. Ajila, C. Lung and W. Ding, "Mobile Network Traffic Prediction Using MLP, MLPWD, and SVM," 2016 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, 2016, pp. 402-409. DOI: https://doi.org/10.1109/BigDataCongress.2016.63 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7584969&isnumber=758490
4. 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 International Conference on Network Protocols (ICNP), Singapore, 2016, pp. 1-5. DOI: https://doi.org/10.1109/ICNP.2016.7785327 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7785327&isnumber=7784399
5. Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," 2017 International Symposium on Wireless Communication Systems (ISWCS), Bologna, 2017, pp. 1-6. DOI: https://doi.org/10.1109/ISWCS.2017.8108090 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8108090&isnumber=8108089
6. L. Le, B. P. Lin, L. Tung and D. Sinh, "SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G," 2018 IEEE 5G World Forum (5GWF), Silicon Valley, CA, 2018, pp. 20-25. DOI: https://doi.org/10.1109/5GWF.2018.8516953 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8516953&isnumber=8516707
7. Rojas Melndez, Juan and Rendn, Alvaro and Corrale,"Personalized Service Degradation Policies on OTT Applications Based on the Consumption Behavior of Users", 2018, pp. 543-557 DOI: https://doi.org/10.1007/978-3-319-95168-3_37
8. D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2, pp. 224-227, April 1979. DOI: https://doi.org/10.1109/TPAMI.1979.4766909 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4766909&isnumber=4766893

9. R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N Shahriar, F. Estrada-Solano and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Journal of Internet Services and Applications, Elsevier, 2018. DOI: https://doi.org/10.1186/s13174-018-0087-2
10. M. Wang, Y. Cui, X. Wang, S. Xiao and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," in IEEE Network, vol. 32, no. 2, pp. 92-99, March-April 2018. DOI: https://doi.org/10.1109/MNET.2017.1700200 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8121867&isnumber=8329608
11. K. Sideris, R. Nejabati and D. Simeonidou, "Seer: Empowering Software Defined Networking with Data Analytics," 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), Granada, 2016, pp. 181-188. DOI: https://doi.org/10.1109/IUCC-CSS.2016.033 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7828600&isnumber=7828567
12. URL: http://mininet.org/
13. URL: https://www.opennetworking.org/
14. URL: https://osrg.github.io/ryu/
15. URL : http://www.grid.unina.it/software/ITG/