**Edelweiss Applied Science and Technology**
*ISSN: 2576-8484*
*Vol. 8, No. 6, 8488-8496*
*2024*
*Publisher: Learning Gate*
*DOI: 10.55214/25768484.v8i6.3822*

# Robust ensemble learning technique for traffic classification in SDN networks

Sura F. Ismail[1]*, Noor Sabah[2]
[1]University of Information Technology and Communications, Baghdad, Iraq; Sura.fawzi89@uoitc.edu.iq (S.F.I.).
[2]Department of Electrical Engineering, University of Wasit, Wasit, Iraq; noors@uowasit.edu.iq (N.S.).

**Abstract:** By constantly changing flow rules, software-defined network (SDN) offers centralized control over a network of programmable switches. This opens the door for the network to be controlled dynamically and independently. SDN requires information from traffic categorization techniques for the appropriate group of rules to be apply to the proper set of traffic flows. Machine learning nowadays uses a range of categorization methods. A framework known as ensemble that mixes independent models to enhance an overall result has grown in popularity in recent studies showing that applying any algorithm does not always result in the best results for a dataset. Therefore, this paper suggests utilizing the ensemble model with two layers of learning methods to categorize incoming network traffic so that SDN may select the best set of possible traffic regulations using Orange platform. We also apply five machine learning methods and analyze their classification performance in terms of accuracy, precision, and recall. The experimental results reveal that ensemble model-based network traffic classifiers outperform other classifiers based on the proposed framework and the real-world network traffic dataset. Notably, the XGBoost model achieves the best classification performance in every type of traffic examined.

***Keywords:*** *Software-defined network (SDN), traffic categorization techniques, XGBoost model.*

## 1. Introduction

Advances in Internet of Things (IoT) technology have increased the number and usage of wireless sensor nodes [1,2]. The rapidly increasing smart devices and IoT modules in volume generate an enormous amount of data traffic, putting an additional burden on existing mobile wireless sensor networks. Therefore, the fifth generation (5G) networks are anticipated to provide a wide range of services for several industrial verticals with various performance and service requirements [3]. The service-based, micro-services-based design of the 5G network enables several virtual networks to operate concurrently, each one configured for a different use case. This is done to enable a variety of use cases. Network slices are the name given to these types of virtual networks. They must support the appropriate policies and, when required, be dynamically configured to fulfill the needs of their user-cases and adapt to changing network circumstances. When networks are adaptable and programmable, network slices may be dynamically established and managed throughout their lifecycle. Software-defined networking (SDN) and network functions virtualization (NFV) can help with this by enabling a wide range of varied end-to-end services across a single, shared physical infrastructure [4]. The core concept of this infrastructure is to give networks the flexibility.

* Correspondence: Sura.fawzi89@uoitc.edu.iq

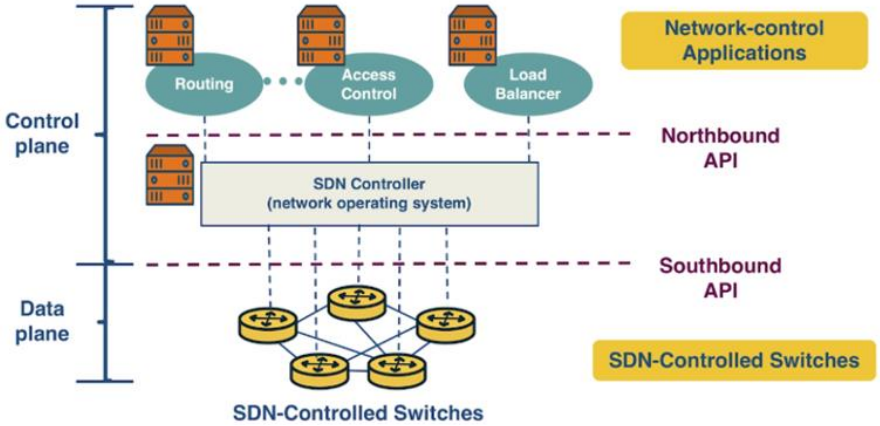

**Figure 1.**
SDN controller architecture.

An SDN controller in an SDN network chooses the rules for forwarding packets and then configures the network components in accordance [5]. The expense of maintaining a highly flexible and programmable network might increase noticeably with the extensive usage of virtual network functions and the dynamic deployment of virtual network slices [6]. The diversity, complexity and volume of 5G traffic flows will be increasing. Because some of these communication types have strict criteria for delay, managing this diversity and amount of traffic calls for a minimum processing delay. Therefore, it is crucial to correctly characterize 5G traffic flows first before implementing these rules in networking hardware. This is possible with SDN architecture with automated features [7] that can utilize the physical resources of the network to their fullest potential while adapting to changing patterns of traffic and the agreements of service level for a variety of traffic kinds. Despite the fact that there is a sizable body of research on SDN solutions that can already perform standard networking activities, the SDN automation networks is still in its infancy. Algorithms of Machine learning (ML) may learn and anticipate traffic patterns by gathering telemetry data, which then enables SDN to more effectively provide the actual network. In this paper, we concentrate on traffic classification for SDN network. In order to assess the efficacy of the classification, several ML models have initially been applied to various categories of application traffic datasets. We contend that a classifier must have two layers when choosing the best ML technique for a particular traffic type in order to increase classification accuracy. We suggest using the ensemble ML framework [8] to be applied to the unique issue of network traffic classification as an input to SDN. The advantages of this strategy in terms of performance accuracy and robustness have been demonstrated by promising results, as seen in Figure 2.

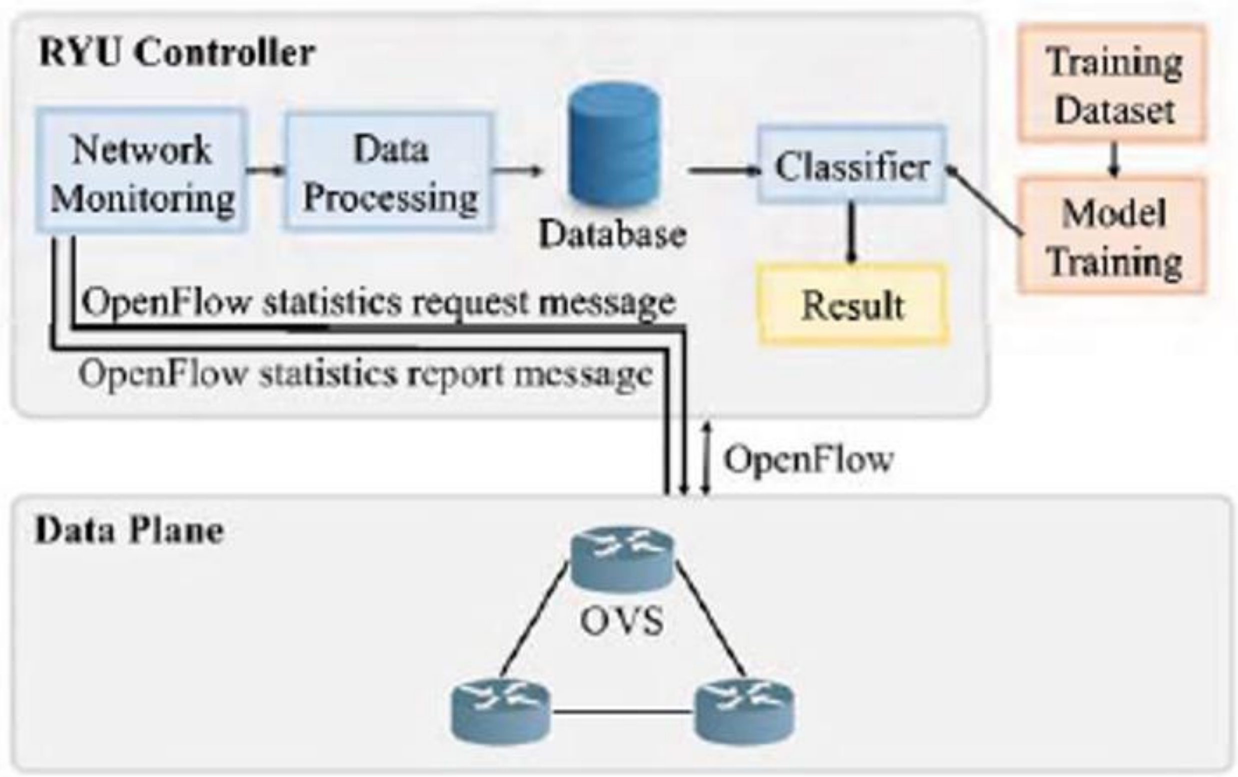

**Figure 2.**
Ensemble model-based traffic classification framework.

## 2. Literature Review

With the development of computer structure, ML has recently become more widely applicable and actually practicable, making it a popular method for solving the many unresolved technological issues of the present. One of the most raising areas is the classification of traffic flows in networks; today's networks have extremely complicated traffic engineering flows due to the wide variety of application scenarios and networking and connection options. Due to SDN's advantages in network programmability, which are currently making it more and more popular, there has recently been interest in integrating machine learning solutions with SDN to accomplish a variety of objectives, including traffic classification. For networks to function effectively, it is vital to identify the different types of traffic flows. For instance, if elephant flows are not adequately handled, they might quickly cause network congestion. However, it might be difficult to identify a particular traffic type among a mix of other traffic flow kinds in an operating network. In [9], a ML approach called C4:5 is suggested particularly for recognizing flows of elephant for real-time in data centers to address this issue. The authors note that while adding additional data characteristics to C4:5 might enhance classification performance; it also slows down real-time traffic response. Machine learning techniques may be used to detect numerous traffic flows by classifies a finer degree of granularity simultaneously instead of just one type of traffic flow. The authors of [10] provide a framework for categorizing various application traffic streams. The top forty downloaded applications from the Google Play store are identified using a model called c5:0 which is a machine learning model with tree decision. Forty-eight application kinds can be correctly recognized, but only eight of them can be done so with any degree of accuracy. For traffic categorization in [11], authors employ deep packet inspection and the Laplacian support vector machine semi-supervised machine learning technique. Nine of the sixty characteristics that the algorithm retrieves from the tested flows of traffic are used by utilizing the classification model's complexity. Using a single and simple classification model when several traffic types coexist for training, may only be effective for certain types of flows. In order to improve classification performance, current research offer ensemble methods that integrate several classifiers. The identification of abnormal network traffic is another use for ensemble algorithms. While the ensemble classifier is further investigated in an SDN context for recognizing DDoS traffic in [12], authors of [13] assess a number of ensemble frameworks for detecting anomalies in network traffic. The research described in [8] utilizes a classification model with multi-class traffic wherein incoming traffic flows are subjected to a pipeline of classification models. Each model in a pipeline-based multi-classifier technique anticipates a particular type of traffic. The first classification model that corresponds to a certain traffic type receives a traffic flow as it arrives. The traffic is subsequently sent to the next model if it is not a match. This procedure is repeated until a model in the pipeline is able to identify the proper traffic type and conduct detection with adequate accuracy.

Multi-model with pipeline-based classifiers is useful for categorizing a small number of traffic flow types, but they are ineffective when the machine learning system needs to handle many different traffic kinds. This is due to the possibility that alternative models in the pipeline, albeit having varying classification ability, may produce appropriate results for a particular kind of incoming traffic. The best model may not always be utilized for classification when using a pipeline-based multi-model classifier since when a model selects the kind of traffic, the traffic leaves the pipeline. Therefore, the pipeline's model ordering has an effect on the final classification outcomes. For a certain collection of traffic types, the ideal pipeline architecture can be chosen, however this configuration can change for another set. Determining the best arrangement for a variety of traffic kinds can be quite expensive. In the worst case, a flow of traffic must also transit through every model before it can reach the right traffic category. To improve classification performance, the pipeline must expand as more models are introduced as more traffic types need to be categorized. In conclusion, there is more work to be done in order to maintain the ML system's scalability and efficiency while also getting the highest performance outcomes for classification of wide range of traffic. For this reason, we suggest an ensemble model that aggregates the output of many categorization models into a two-tiered structure.

## 3. The Proposed Ensemble Classifier

The general technique used in previous research when developing a classifier model for traffic flow in a network is either to apply a variety of machine learning models and compare their results to choose the best one, or to improve a particular model. A system, however, that uses a single machine learning algorithm is not reliable. As a result, we take a different structure and use the ensemble model to concurrently train numerous classifiers in order to combine their predictions to get a final prediction. The suggested two-tier ensemble approach's process is depicted in Figure 3.

On the top tier, the first tier, several classifier models are independently trained using the same dataset for training. There is no difference between this training procedure and the absence of a second layer (i.e. no ensemble). The second layer is where the distinction resides; the ensemble algorithm concentrates on taking the prediction of the first tier to enhance the final prediction results.

The support vector machine (SVM), KNN, random forest, AdaBoost, and gradient boosting machine learning models are the five most popular learning models in the first tier. These five models have been trained, and the outcomes they predict are kept in an array called predictions. The projected traffic categories of the five different machine learning models, marked by predictions, are the output of the top-tier classifiers. The second-tier classifier is then trained using this as training data. In the second level, we select the well-known extreme gradient boosted learning algorithm XGBoost [14] to forecast the outcomes. This approach to classification model construction tries to merge many models to create a strong and adaptable classifier.

*DOI: 10.55214/25768484.v8i6.3822*

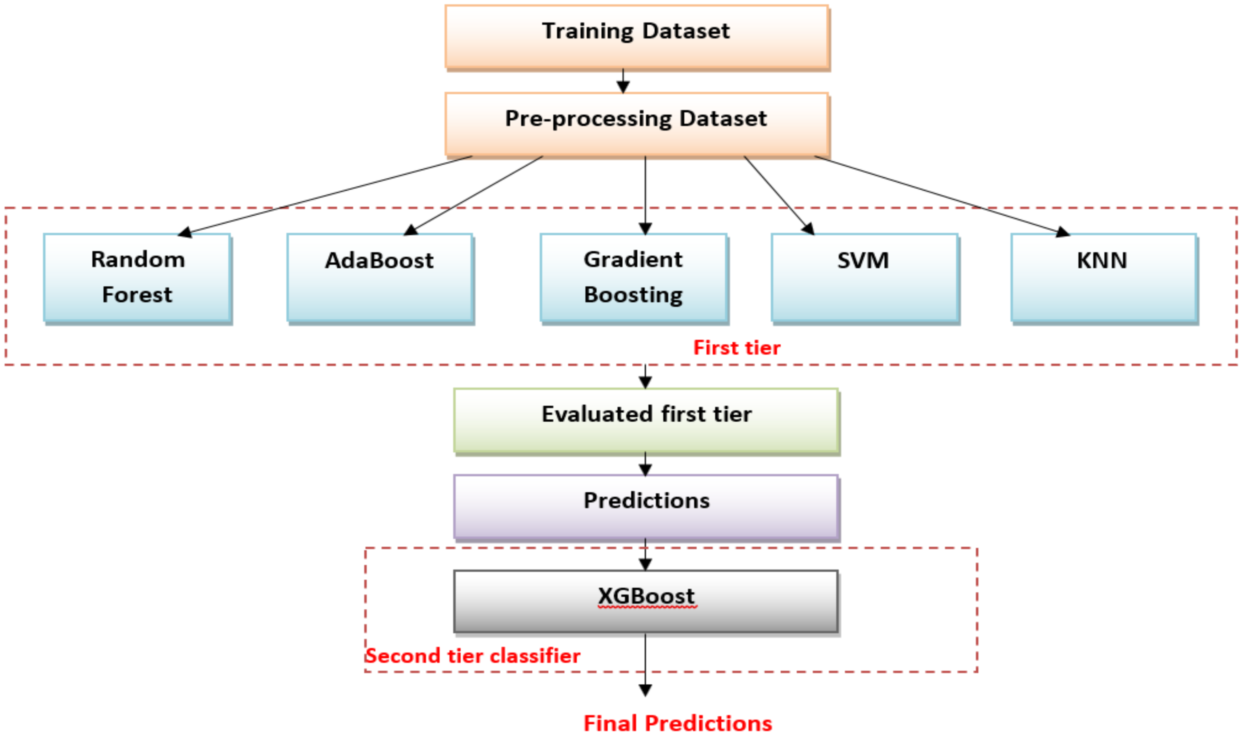

**Figure 3.**
General block diagram of the ensemble classifier.

## 4. Performance Results

The performance results of the proposed model with ensemble framework for traffic classification in Software- Defined Network are shown in this section.

### *4.1. Performance Setup*

In this section we will briefly illustrate the dataset, testbet and evaluation metrics. Firstly, Tor-nonTor dataset [15] which contains only time-related features is the dataset used for testing in this paper. All network flows were categorized into seven classes, such as web browsing, email, chat, streaming, file transfer, VoIP and P2P. The traffic classes and corresponding network applications are summarized in Table 1. Instead of identifying specific applications, we classified the network traffic into different classes according to the QoS requirements. Also, we used the Orange platform [16] to build the proposed ensemble algorithm as shown below.

**Table 1.**
Traffic classes and corresponding applications.

| Traffic class | Application |
|---|---|
| Web Browsing | Firefox, Chrome |
| Email | SMTPS, POP3, IMAPS |
| Chat | AIM, Skype, Facebook, Hangouts |
| Streaming | Vimeo, YouTube |
| File Transfer | Skype, FTPS, SFTP |
| VOIP | Facebook, Skype, Hangout voice calls |
| P2P | uTorrent, BitTorrent |

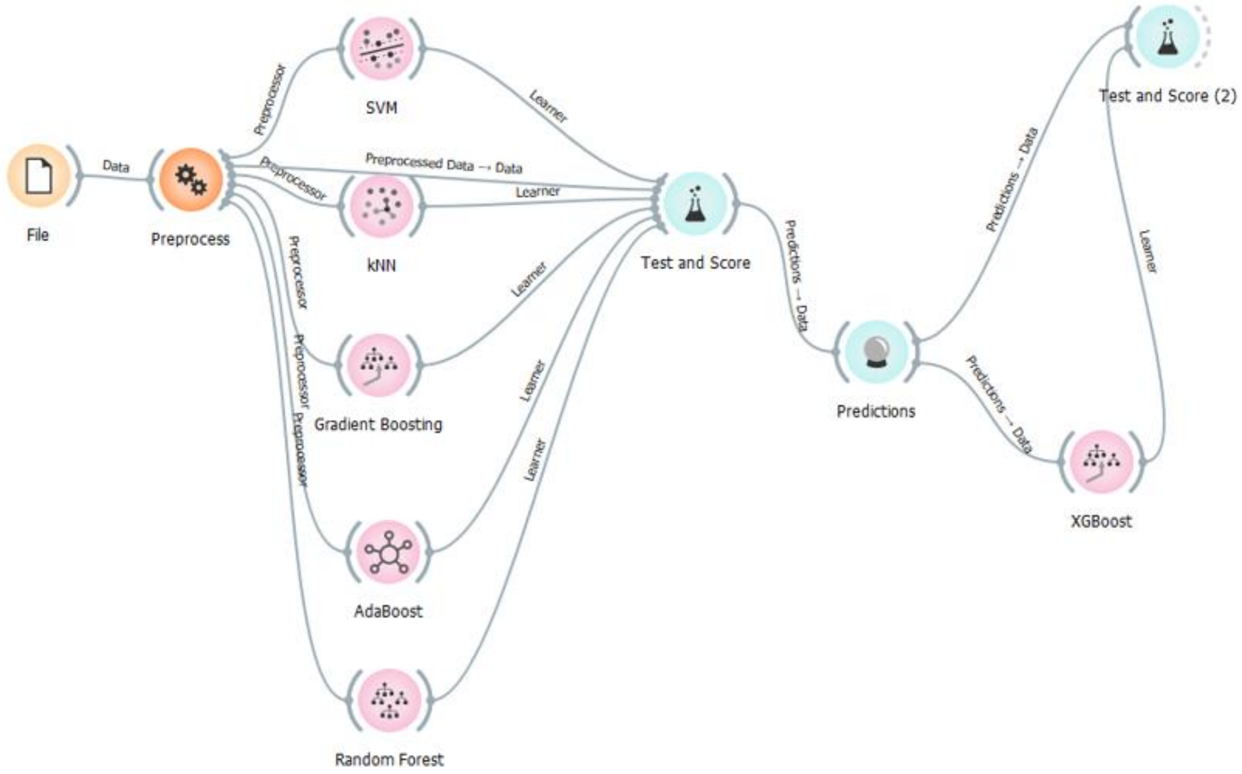

**Figure 4.**
A screen shoot for the proposed two-tier classifier in Orange platform.

The essential metrics used to calculate the performance results are described briefly in Table 2.

**Table 2.**
General forms and interpretation of the three metrics used in this paper.

| Metric | Interpretation |
|---|---|
| Accuracy | Accuracy is expressed as a percentage of all traffic flows that were accurately anticipated. Accuracy can be expressed using the true positive (TP), true negative (TN), false positive (FP), and false negative (FN):<br>$$Accuracy = \frac{\mathrm{TP + TN}}{\mathrm{TP + TN + FP + FN}}$$ |
| Recall | The percentage of properly anticipated traffic flows in a given traffic class, out of all the traffic flows in that class, is known as recall. Recall can be expressed using the true positive (TP) and false negative (FN):<br>$$Recall = \frac{\mathrm{TP}}{\mathrm{TP + FN}}$$ |
| Precision | A traffic class's precision is calculated as the proportion of all anticipated traffic flows that really belong to that class that were accurately predicted. Precision can be described in terms of the true positive (TP) and false positive (FP):<br>$$Precision = \frac{\mathrm{TP}}{\mathrm{TP + FP}}$$ |

### *4.2. Performance Results*

As mentioned in the previous section, we utilize the most prevalent and well-liked ML techniques for traffic categorization in the top layer of the proposed ensemble models: SVM, KNN, AdaBoost, random forest and gradient boost. The XGBoost classifier is used in the second layer to provide the final strong prediction outcomes. We compare the performance of the proposed ensemble model with results obtained by utilizing single classifiers in isolation (e.g. use gradient boost only). Figure 5 displays the overall accuracy, precision, and recall of several machine learning techniques for multi-class categorization. Different machine learning algorithms can provide outcomes that are noticeably quite different. It is important to note that the SVM classifier performs rather poorly. This is because SVM typically performs well with a small dataset size and a big number of features, but only a small number of features may be employed for classification in SDN networks. The suggested ensemble classifier, in contrast, minimizes the negative effects of the few characteristics to produce the best results. Additionally, the ensemble classifier displays acceptable generalization performance, which means that the over-fitting issue is reduced, as seen in Figure 5. This is crucial for real-world networks because highly dynamic networks can display a wide range of traffic patterns. Overall, the performance outcomes in Figure 5 indicate that employing the suggested ensemble classifier rather than depending solely on a single classifier produces superior results for various classification applications. The "accuracy, recall, and precision results" are shown in Table 3 for various forms of traffic; the closer these numbers are to 1, the better. Overall, it can be seen that each classifier has both strengths and weaknesses when it comes to categorizing particular kinds of traffic flows. Examples include the exceedingly poor performance of SVM and KNN classifiers on "VIDEO" type traffic flows. In contrast, we consistently get great performance for all distinct traffic types when utilizing the suggested ensemble classifier.

In conclusion, the performance results of Figure 5 and Table 3 demonstrate that the suggested ensemble classifier not only works effectively for the entire dataset but also for different types of traffic.

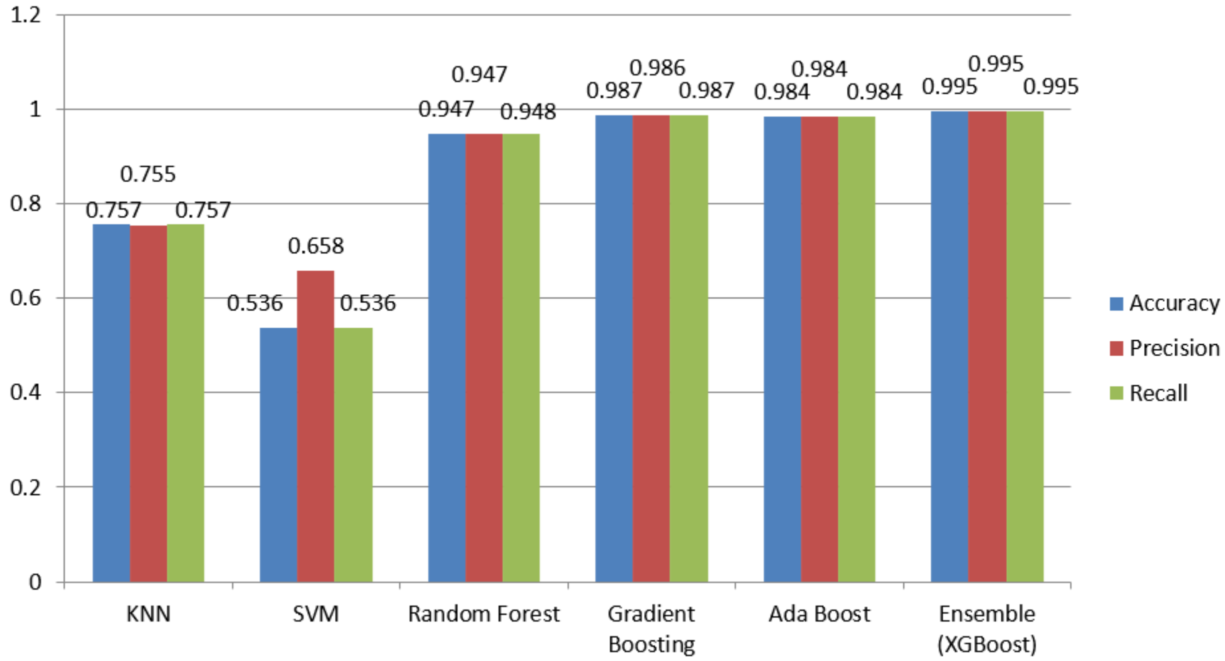

**Figure 5.**
Accuracy, Precision and Recall of different ML algorithms.

*DOI: 10.55214/25768484.v8i6.3822*

**Table 3.**
Accuracy, precision and recall results of different classifiers for each traffic types.

| Traffic types | KPI | KNN | SVM | Random forest | Ada boost | Gradient boosting | XGboot |
|---|---|---|---|---|---|---|---|
| Audio | Accuracy | 0.931 | 0.850 | 0.977 | 0.995 | 0.998 | 0.998 |
| | Precision | 0.596 | 0.327 | 0.887 | 0.974 | 0.992 | 0.996 |
| | Recall | 0.713 | 0.634 | 0.853 | 0.972 | 0.989 | 0.986 |
| Browsing | Accuracy | 0.839 | 0.835 | 0.966 | 0.992 | 0.996 | 0.997 |
| | Precision | 0.577 | 0.627 | 0.893 | 0.981 | 0.989 | 0.985 |
| | Recall | 0.727 | 0.424 | 0.941 | 0.981 | 0.998 | 0.997 |
| Chat | Accuracy | 0.956 | 0.968 | 0.991 | 0.997 | 0.999 | 0.999 |
| | Precision | 0.422 | 0.653 | 0.897 | 0.963 | 0.991 | 0.997 |
| | Recall | 0.241 | 0.455 | 0.867 | 0.963 | 0.981 | 0.978 |
| File-Transfer | Accuracy | 0.943 | 0.893 | 0.993 | 0.998 | 0.999 | 1.000 |
| | Precision | 0.735 | 0.889 | 0.981 | 0.993 | 0.999 | 1.000 |
| | Recall | 0.740 | 0.009 | 0.955 | 0.992 | 0.999 | 0.999 |
| Mail | Accuracy | 0.959 | 0.964 | 0.992 | 0.999 | 1.000 | 1.000 |
| | Precision | 0.346 | 0.477 | 0.917 | 0.986 | 0.989 | 0.989 |
| | Recall | 0.188 | 0.365 | 0.858 | 0.982 | 0.996 | 1.000 |
| P2p | Accuracy | 0.988 | 0.997 | 0.995 | 0.995 | 0.997 | 0.997 |
| | Precision | 0.953 | 0.999 | 0.988 | 0.980 | 0.994 | 0.994 |
| | Recall | 0.959 | 0.987 | 0.975 | 0.982 | 0.987 | 0.984 |
| Video | Accuracy | 0.912 | 0.727 | 0.988 | 0.996 | 0.999 | 1.000 |
| | Precision | 0.629 | 0.176 | 0.930 | 0.980 | 0.999 | 1.000 |
| | Recall | 0.460 | 0.412 | 0.965 | 0.985 | 0.999 | 1.000 |
| Voip | Accuracy | 0.984 | 0.837 | 0.992 | 0.995 | 0.998 | 0.999 |
| | Precision | 0.987 | 0.743 | 0.992 | 0.992 | 0.997 | 0.997 |
| | Recall | 0.958 | 0.653 | 0.982 | 0.990 | 0.996 | 0.998 |

## 5. Conclusion and Future Work

We present a machine learning framework with ensemble classifier and assess its functionality in SDN networks for traffic categorization. The amount of real-time data characteristics accessible to classification algorithms is constrained since SDN can only process and read the header of packet information. We provide a multi-tier classification, two layers, approach that utilizes the benefits of many classifiers to make up for the lack of classification information. The suggested classifier not only resulting in a high accuracy, but also enhances the overall performance for different traffic kinds, according to experimental data. The ensemble classifier will be tested for unbalanced datasets with semi-supervised ML models in upcoming research.

## Copyright:

## References

[1] Mainetti, L.; Patrono, L.; Vilei, A. Evolution of wireless sensor networks towards the Internet of Things: A survey. In Proceedings of the SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks, Split, Croatia, 15–17 September 2011; pp. 16–21.

[2] Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. IEEE Trans. Ind. Inform. 2014, Vol.10, pp. 2233–2243.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," IEEE Commun. Mag., 2017.

[4] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," IEEE Commun. Surv., 2019.

[5] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," IEEE Commun. Surv., 2014.

[6] Ericsson, "Scalable network opportunities: An economic study of 5G network slicing for IoT service deployment," 2018.

[7] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," IEEE Network, 2018.

[8] S. E. G´omez, B. C. Mart´ınez, A. J. S´anchez-Esguevillas, and L. H. Callejo, "Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal," Computer Networks, 2017.

[9] P. Xiao, W. Qu, H. Qi, Y. Xu, and Z. Li, "An efficient elephant flow detection with cost-sensitive in SDN," in INISCom, 2015.

[10] Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir, "Application-Awareness in SDN," in ACM SIGCOMM, 2013.

[11] P. Wang, S. Lin, and M. Luo, "A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs," in IEEE SCC, 2016.

[12] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Design of Ensemble Learning Methods for DDoS Detection in SDN Environment," in ViTECoN, 2019.

[13] I. P. Possebon, A. S. Silva, L. Z. Granville, A. Schaeffer-Filho, and A. Marnerides, "Improved Network Traffic Classification Using Ensemble Learning," in IEEE ISCC, 2019.

[14] [Online]. Available: https://xgboost.readthedocs.io/en/latest/

[15] Lashkari, A.H.; Draper-Gil, G.; Mamun, M.S.I.; Ghorbani, A.A., "Characterization of tor traffic using time based features", In Proceedings of the 3rd International Conference on Information System Security and Privacy (ICISSP), Porto, Portugal, 19–21 February 2017; pp. 253–262.

[16] [Online]. Available: https:// Orange Data Mining - Download.