IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 3, September 2022, pp. 1175~1183
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i3.pp1175-1183

Features analysis of internet traffic classification using interpretable machine learning models

Erick A. Adje[1], Vinasetan Ratheil Houndji[2], Michel Dossou[3]
[1]Ecole Doctorale des Sciences de l'Ingénieur, Université d'Abomey-Calavi, Abomey-Calavi, Bénin
[2]Institut de Formation et de Recherche en Informatique, Université d'Abomey-Calavi, Abomey-Calavi, Bénin
[3]Ecole Polytechnique d'Abomey-Calavi, Université d'Abomey-Calavi, Abomey-Calavi, Bénin

Article Info

Article history:

Received Jul 16, 2021
Revised Mar 14, 2022
Accepted Apr 12, 2022

Keywords:

Classification algorithm
Internet traffic
Machine learning
Traffic classification
Traffic internet discriminators

ABSTRACT

Internet traffic classification is a fundamental task for network services and management. There are good machine learning models to identify the class of traffic. However, finding the most discriminating features to have efficient models remains essential. In this paper, we use interpretable machine learning algorithms such as decision tree, random forest and eXtreme gradient boosting (XGBoost) to find the most discriminating features for internet traffic classification. The dataset used contains 377,526 traffics. Each traffic is described by 248 features. From these features, we propose a 12-feature model with an accuracy of up to 99.76%. We tested it on another dataset with 19626 flows and obtained 98.40% of accuracy. This shows the efficiency and stability of our model. Also, we identify a set of 14 important features for internet traffic classification, including two that are crucial: port number (server) and minimum segment size (client to server).

Corresponding Author:

Vinasetan Ratheil Houndji
Institut de Formation et de Recherche en Informatique, Université d'Abomey-Calavi
01 BP 526 Abomey-Calavi-Bénin
Email: ratheil.houndji@uac.bj

1. INTRODUCTION

Internet traffic has increased significantly over the last decade due to new technologies, industries, and applications. It becomes an interesting challenge for network management. Accurate classification of internet traffic is fundamental for better management of network traffic, from monitoring to security, from the quality of service (QoS) to the provision of the right resource. Automatic traffic classification is an automated process that classifies network traffic according to various parameters (e.g., port number, protocol, and the number of packets exchanged) into various traffic classes (e.g., web, multimedia, database, e-mail, games, and file transfer). It consists of examining internet protocol (IP) packets to extract some specific characteristics to answer some questions related to their origins such as the content or the user's intentions. Typically, it deals with packet flows defined as sequences of packets uniquely identified by the source IP address, source port, destination IP address, destination port and protocol used at the transport layer, and many others.

While research on traffic classification is quite specific, the author's motivations are not always the same [1]. Some approaches classify traffic according to its category i.e., whether the traffic represents file transfer, peer to peer (P2P), games, multimedia, web, or attacks [2]–[8]. Others try to identify the protocol involved at the application level such as file transfer protocol (FTP), hypertext transfer protocol (HTTP), secure shell (SSH), Telnet [9]–[14]. One particular study reviewed current traffic classification methods by classifying them into five categories: statistics-based, correlation-based, behaviour-based, payload-based, and port-based [15]. Some studies [16], [17] have provided classification methods for encrypted traffic, which was challenging to perform in the past. Today port-based analysis is ineffective, being unable to identify 30-70% of today's internet traffic [5], [11]. This leads to the exploration of new features for traffic classification.

Since the first studies on the statistical classification of internet traffic, the classification of network traffic using supervised and unsupervised machine learning techniques based on flow features such as average packet size, packet arrival times, and flow transmission times, have generated a lot of interest. These features are calculated over several packets grouped into a flow, and these sets of features are associated to the relevant flow class. Khandait *et al.* [14] inspected the few initial bytes of payload to determine the potential application. This study achieved an accuracy of 98%. Moore and Zuev [6] proposed a statistical approach to classify traffic into different classes of internet applications based on a combination of flow features such as the length of the flow, the time between consecutive flows, and the time between arrivals. The classification process uses a Bayesian classifier combined with a kernel density estimation which gives accuracy up to 95%. The models obtained are generally not very effective for certain types of traffic, such as attacks and P2P. However, they are particularly effective for web and mail traffic, which alone represent more than 94% of the data used for the study. Auld *et al.* [18] used a classification approach based on Bayesian neural networks to classify traffic into eight classes and present a traffic classifier that can achieve a high accuracy across various application types without any source or destination host-address or port information. They achieved up to 99% accuracy for data trained and tested on the same day and 95% accuracy for data trained and tested eight months apart. Fan and Liu [19] used support vector machine (SVM) for internet traffic classification. Several SVM cores have been tested, and the most interesting one was the radial core. Several feature combinations were made from 30 features to create models. The most interesting one was a combination of 13 features which resulted in an overall accuracy of 98%. To ensure the stability of their model, an evaluation phase was carried out on a new dataset obtained later. Later, we compare our results against this study. Erman *et al.* [20] proposed a semisupervised traffic classification approach that combines unsupervised and supervised methods. This method achieved an accuracy of 94%. Li *et al.* [21] used the SVM in the classification of multi-class network traffic. Thus, from nine features, they built a model capable of predicting six classes of traffic with an accuracy of 99.4%. Este *et al.* [22] proposed a two-step approach to multi-class traffic classification based on the SVM: a single class classification step followed by a multi-class classification step that achieved an accuracy of around 90% for each class category (http, smtp, pop3, ftp, bittor, msn).

However, it is well known that the characterization of the phenomenon to understand or the object to learn in a learning system is a critical step toward having a good classifier. Unfortunately, most existing works in the state-of-the-art miss a formal study to ensure that the features used are informative and discriminating. Some are just based only on a limited number of features without explanation. It is sometimes difficult to identify the features and the properties that influence the results obtained. Moreover, it is difficult to know which features to combine to obtain a simple but efficient model. Note that some state-of-the-art works, despite achieving good performance, fail to perform well on specific classes especially classes traffic-related to bulk and attack.

In this paper, our contribution is twofold. Firstly, we studied the internet traffic features to select only relevant and informative enough by most machine learning algorithms. These selected features could be described as essential in the classification of internet traffic. Secondly, we made sure that we can detect any class of traffic, i.e., to be very efficient on all types of traffic thanks to our models. Our models will also be adaptable to data deficits. This means exploiting the available features for a given flow to predict its class without having all the features. The remainder of this paper is organized: section 2 presents the datasets used, the performance metrics used, and the methodology. Then section 3 shows results obtained, some comparisons with the existing works and conclusions to be drawn.

2. MATERIALS AND METHODS

In this paper, we considered the following machine learning algorithms: decision tree, random forest, and eXtreme gradient boosting (XGBoost) because of their respective capacities to highlight the most discriminative features. We used the python programming language through libraries such as scikit-learn and XGBoost to implement these different machine learning models. This section describes the dataset used, the performance metrics, and the methodology.

2.1. The dataset

In this paper, the dataset used to develop and evaluate our models was collected by high-performance monitors [23] at Queen Mary in University of London. The experimental site for collecting data is a large research facility host to approximately 1,200 administrators, technical staff, and researchers. Full-duplex gigabit ethernet is used on this site to connect to the internet. The traffic dataset is obtained based on full-duplex traffic traces of the research facility over 24 hours. To build the sets of flows, the trace of each day was split into ten blocks of approximately 1,680 seconds (28 minutes). To provide a wider sample of mixing across the day, the beginning of each sample was selected randomly (uniformly distributed over the whole trace). The dataset consists of 377,526 flows, and each flow is characterized by 248 features described in [24]. Thus, we should have 377,526*248=93,626,448 values, but there are 1,105,574 missing values. These features include traffic statistics about inter-packet time or packet size and information obtained from the transmission control protocol (TCP) headers, such as acknowledgment counts. Note that the features are provided in both directions. Flows in datasets are manually classified into ten broad traffic categories by applying a content-based mechanism. The available traffic classes in this dataset are provided in Table 1. Each flow is mapped to one traffic class. Table 1 shows the number of flows by class/application in the dataset. Note that since game and interactive flows are not sufficient, our work was done on the eight other classes. To evaluate models efficiently, an eleventh block of data was obtained in the same way as the first ten blocks one year later at the same place. This 11th block of 19,626 flows is used for the evaluation phase in our study, as shown in Table 1. All these datasets are public, free to use for academic purposes, and available through a web link [25].

Table 1. Dataset statistics with traffic classes application

Traffic class	Applications	Nb of flows (10 blocks)	Nb of flows (11th block)
WWW	Web	328,092	15,597
Mail	SMTP, POP3, IMAP	28,567	1,799
Bulk	FTP	11,539	1,513
Service	DNS, X11, NTP	2,099	121
P2P	BitTorrent, eDonkey	2,094	297
Database	Mysql, Oracle	2,648	295
Multimedia	Windows Media Player	576	0
Attack	Virus, Worm	1,793	0
Interactive	TELNET, SSH	110	4
Games	World of Warcraft	8	0

2.2. Performance metrics

Let true positive (TP) be the number of correct positive classifications, true negative (TN) the number of correct negative classifications, false positive (FP) be the number of incorrect positive classifications, and false negative (FN) be the incorrect negative classification. We consider four main classical machine learning performance metrics: precision, recall, accuracy, and F1-score. Precision (1) is the percentage of correct positive classifications (TP) from samples that are predicted as positive. Recall (2) is the percentage of correct positive classifications (TP) from samples that are actually positive. Accuracy (3) is the percentage of correct classifications from the overall number of samples. F1-score (4) is a combined metric that evaluates the trade-off between precision and recall.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$f1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

2.3. Methodology

We followed a traditional methodology for the machine learning task. The main steps are:

- Pre-processing: cleaning, homogenization, and adaptation of the dataset to the learning algorithm. We replaced each missing quantitative value by the mean of the corresponding column and each qualitative missing value by the most frequent value of the corresponding column for the decision tree and random forest algorithms. It should be noted that, relying on the adaptability of the XGBoost algorithm to missing data, we did not perform any imputation for XGBoost.

- Learning: for each algorithm, we varied several hyper-parameters and trained the different models obtained on the same training set. 70% of total traffic is used for training and 30% for testing. Then we selected the best model based on the metrics used. Moreover, to ensure the efficiency of the model each time (evaluation phase), we test it again on a new dataset of 19,626 flows which is the eleventh block described before.
- Features selection: for each model, we identified the importance of each of the 248 features and removed less important ones.
- Repeating: we repeated steps 2 and 3 by considering the new reduced features until we got the fewest possible features with performance better or close to the performance of the model selected in the previous iteration.

3. EXPERIMENTAL RESULTS

As explained in section 2 about materials and methods, we performed the experiments with three algorithms: decision tree, random forest, and XGBoost. For each algorithm, we used the classical grid search approach to find the best hyperparameters. In this section, we present the experimental results obtained during the test and evaluation phase. These results consist of the accuracy of each traffic class and the overall accuracy.

3.1. Decision tree

Table 2 presents the results using the decision tree. We can see that, even without too many parameters, the decision tree is an efficient technique to solve traffic classification problems. Despite the considerable amount of missing data, which were filled, the decision tree can adapt and gives a good performance. The first model built by the decision tree with 248 features has revealed that: i) more than half (129) of the features provided zero information to the model; ii) 236 features had total importance of 1.38%, while the other 12 features provided a total of 98.62% information needed. Then, only the 12 most important features were considered. Note that the 12-feature model is better than the 248-feature model in terms of overall accuracy and specifically on traffic classes such as web, mail, bulk, P2P. In addition, this model consumes less memory space and is faster during the training than the 248-feature model.

Table 2. Summary of performance obtained with decision tree

Traffic class		Test phase			Evaluation phase	
	248 features	12 features	6 features	248 features	12 features	6 features
WWW	99.85%	99.86%	99.85%	98.72%	99.76%	99.90%
Mail	99.93%	99.97%	99.94%	99.77%	99.94%	95.83%
Bulk	99.47%	99.83%	99.38%	83.21%	83.28%	83.61%
Service	99.41%	99.17%	99.12%	99.17%	99.17%	99.17%
Database	99.88%	99.88%	99.76%	98.98%	98.98%	98.98%
P2P	98.53%	97.39%	96.08%	90.90%	93.93%	52.52%
Attack	82.92%	82.54%	81.02%	-	-	-
Multimedia	93.64%	94.21%	95.95%	-	-	-
Overall accuracy	**99.74%**	**99.76%**	**99.72%**	**97.52%**	**98.40%**	**97.53%**

Considering the six most essential features, we have noticed a lot of loss performance in some traffic classes, in particular on P2P, which decreases from 93.93% to 52.52%. We have then considered this model less interesting for the next step. With the 12-feature model, we have noticed that two features have total importance of 92.77% against 7.23% for the other ten features. These two features are the port used at the server for the traffic with importance of 39.96% and the total number of bytes sent in the initial window, i.e., the number of bytes seen in the initial flight of data before receiving the first ack packet from the other endpoint (client to server) with importance of 52.81%. Table 3 shows the importance of the features in the 12-feature model building process.

3.2. Random forest

From a random forest model of 248 features with an overall accuracy of 99.71%, the model of 23 features has an accuracy of 99.77%, which is the highest accuracy obtained during the test phase. However, the performance on the attack class is only about 71%. Let us note that the model with 23 features performs very well on the traffic class database during the evaluation phase (98.30% accuracy), knowing that at the beginning with 248 features, we have an accuracy of 38.30%. Subsequently, we have reduced the number of features to 17. The performance is very close to the 23-feature model and particularly a tiny improvement once again on the database traffic, which has increased to 98.64% accuracy during the evaluation phase. Table 4 presents the results using random forest, and Table 5 shows the importance of the features in the 17-feature model building process.

Table 3. 12-feature decision tree model features with their importance on the model set up

Discriminators: importance (%)	Discriminators: importance (%)	Discriminators: importance (%)
Port number (server): 39.96%	Port number (client): 1.32%	Maximum of bytes in Ethernet packet: 0.5%
Variance of control bytes packet: 0.44%	Number of pushed data packets (Client to Server): 1.51%	Minimum segment size (Client to Server): 0.20%
Average segment size (Client to Server): 0.20%	Average segment size (Server to Client): 0.23%	Initial window bytes (Client to Server): 52.81%
The variance of total bytes in IP packet: 0.43%	The theoretical stream length (Client to Server): 1.10%	The theoretical stream length (Server to Client): 1.30%

Table 4. Summary of performance obtained with random forest

Traffic class	Test phase				Evaluation phase			
	248 ftrs	78 ftrs	23 ftrs	17 ftrs	248 ftrs	78 ftrs	23 ftrs	17 ftrs
WWW	99.89%	99.91%	99.93%	99.91%	99.93%	99.97%	99.97%	99.97%
Mail	99.94%	99.93%	99.93%	99.94%	90.88%	92.77%	99.94%	99.94%
Bulk	99.66%	99.83%	99.72%	99.86%	99.34%	99.67%	99.73%	98.67%
Service	99.27%	99.41%	99.41%	99.41%	99.17%	99.17%	99.73%	99.73%
Database	99.16%	99.64%	99.76%	99.76%	38.30%	55.93%	98.30%	98.64%
P2P	95.92%	96.90%	98.04%	97.23%	95.62%	98.65%	94.61%	96.30%
Attack	71.53%	72.48%	72.86%	71.92%	-	-	-	-
Multimedia	88.44%	94.21%	95.37%	94.22%	-	-	-	-
Overall accuracy	**99.71%**	**99.75%**	**99.77%**	**99.75%**	**97.52%**	**98.61%**	**99.85%**	**99.80%**

Table 5. 17-feature random forest model features with their importance on the model set up

Discriminators: importance (%)	Discriminators: importance (%)	Discriminators: importance (%)
Port number (server): 18.78%	The number of unique bytes sent (Server to Client): 3.47%	The count of all the packets with at least a byte of TCP data payload (Server to Client): 1.56%
The minimum segment size (Client to Server): 13.61%	The average segment size (Server to Client): 7.16%	Initial window bytes (Client to Server): 20.32%
Initial window bytes (Server to Client): 5.32%	The theoretical stream length (Server to Client): 2.13%	Total data transmit time (Client to Server): 1.23%
The total number of Round-Trip Time (RTT) samples found (Client to Server): 1.66%	Maximum of Ethernet data bytes (Client to Server): 2.40%	Maximum of total bytes in IP packet (Client to server): 2.13%
Maximum of Ethernet data bytes (Server to Client): 4.07%	Variance of Ethernet data bytes (Server to Client): 5.53%	Maximum of total bytes in IP packet (Server to Client): 3.07%
Variance of total bytes in IP packet (Server to Client): 4.90%	The maximum segment size (Server to Client): 2.65%	

3.3. XGBoost

The results obtained with the XGBoost models are presented in Table 6. Note that XGBoost can build models with missing data and still give good models. However, it is an algorithm that requires many features to perform well when dealing with missing data. The best model with XGBoost was obtained by considering 67 features. With 23 features, we have observed some more or less significant drops with an overall accuracy that went from 99.87% to 99.82% during the test phase and from 99.90% to 99.60% for the evaluation phase. The most remarkable drop is in the attack class, decreasing from 80.45% accuracy to 73.43% during the test phase. However, the 23-feature model still performs well on the other traffic classes. As we can see, for a better adaptation of the XGBoost algorithm on traffic classification by exploiting a database with missing data, the more the algorithm has features of the flow to exploit, and the more the model obtained is better. Thus, we can hypothesize that if our models are used for real-time classification, their efficiencies will depend on the number of the features captured at a specific time of the traffic processing and will improve as the traffic progresses. Future works could look at these aspects to confirm. However, it is worth remembering that even with few features, the models based on XGBoost are still very efficient. The experiments with 23 features highlight the importance of two features which are: the port number (server) with importance of 24.79% and the total number of bytes sent in the initial window (client to server) with importance of 47%. These two features contribute the most to the model construction, and further confirming again their importance in internet traffic classification. Table 7 shows the importance of the features in the 23-feature model building process.

Table 6. Summary of performance obtained with decision tree

Traffic class	Test phase			Evaluation phase		
	248 features	67 features	23 features	248 features	67 features	23 features
WWW	99.97%	99.96%	99.97%	99.99%	99.99%	99.99%
Mail	99.97%	99.98%	99.98%	99.94%	99.77%	99.97%
Bulk	99.97%	99.97%	99.86%	99.93%	99.87%	96.16%
Service	99.56%	99.56%	99.41%	95.04%	97.52%	96.69%
Database	99.88%	99.88%	99.88%	98.64%	98.64%	98.64%
P2P	98.86%	98.86%	98.20%	96.96%	97.31%	96.30%
Attack	80.45%	80.45%	73.43%	-	-	-
Multimedia	97.69%	98.26%	96.53%	-	-	-
Overall accuracy	**99.86%**	**99.87%**	**99.82%**	**99.89%**	**99.90%**	**99.60%**

Table 7. 23-feature XGBoost model features with their importance on the model set up

Discriminators: importance (%)	Discriminators: importance (%)	Discriminators: importance (%)
Port number (server): 18.05%	Minimum of bytes in Ethernet packet: 1.10%	Maximum of bytes in Ethernet packet: 0.47%
Mean of total bytes in IP packet: 0.64%	The number of unique bytes sent (Client to Server): 2.51%	The count of all the packets with at least a byte of TCP data payload (Client to Server): 1.96%
If the endpoint requested Window Scaling/Timestamp options as specified (Server to Client): 1.32%	Minimum segment size (Client to Server): 2.57%	Average segment size (Client to Server): 2.27%
Initial window bytes (Client to Server): 55.81%	Initial window bytes (Server to Client): 1.31%	The theoretical stream length (Client to Server): 2.27%
The theoretical stream length (Server to Client): 1.85%	The missed data, calculated as the difference between the ttl stream length and unique bytes sent (Server to Client): 2.09%	Total data transmit time (Server to Client): 0.60%
The total number of Round-Trip Time (RTT) samples found (Server to Client): 0.48%	Maximum of Ethernet data bytes (Client to Server): 0.65%	Mean of total bytes in IP packet (Client to Server): 1.19%
Maximum of total bytes in IP packet (Client to server): 0.40%	Variance of total bytes in IP packet (Client to server): 0.44%	Median of Ethernet data bytes (Server to Client): 0.47%
Maximum of Ethernet data bytes (Server to Client): 1.13%	FFT of packet IAT, Frequency #2 (Client to Server): 0.38%	

3.4. Overall discussions

The results show that each algorithm used has its particularities, weaknesses, and strengths. For example, the decision tree was more efficient on attack traffic than the other algorithms. But it gave less interesting results for bulk traffic during the test phase and generally performed worse than other algorithms. The random forest is the algorithm that achieved good results in both the test and evaluation phases with the least number of features but remained less efficient on attack traffic. XGBoost gives a lower performance as we reduce the features. This can be reflected in the data gaps. It then looks at other features to improve performance. In general, XGBoost performed better in our study when considering 67 features. We obtained an overall accuracy of 99.87% for the test phase and 99.90% for the evaluation phase, representing our best results during the study. Each algorithm works based on the features it considers most discriminating. Nevertheless, several features are very often found in the top 20 features of either two or all three algorithms considered. These features summarized in Table 8 can be considered as essential features for the classification of internet traffic.

3.5. Comparison of our results with the state-of-the-art

We compare our results to the [19] ones as the context of the studie is the same and they used the same datasets as in our study with good performance. We use our minimal models for each algorithm: the 12-feature decision tree model, the 23-feature XGBoost model, and the 17-feature random forest model. In [19] use SVM for this task. Several SVM cores have been tested and the most interesting one was the radial core. The most interesting model obtained was a combination of 13 features which resulted in an overall accuracy of 98%. However, with our 12-feature decision tree model, we obtained an overall accuracy of 99.76%, which represents fewer features for more performance. We also obtained higher overall accuracies from the other models despite requiring more features (99.75% for the 17-feature random forest model and 99.82% for the 23-feature XGBoost model). Tables 9-11 summarise [19]. Results and ours when using the eleventh block. In general, we notice that whatever the considered model our results are better. Moreover, for traffic such as P2P, we have better precision with our models. It is important to mention that 5 of the decision tree model features, as shown in Table 3 are commons to some most discriminating features in the [19]. Study, which once again confirms the importance of some specific features in the classification of internet traffic. However, the use of the other 7 features made the difference and allowed us to achieve better results compared to the study of [19]. Therefore, our analysis of the features was very interesting.

Table 8. Top 14 of most important features common to the considered algorithms

Discriminators	Discriminators
Port number (server)	Mean of total bytes in IP packet
Number of unique data bytes (Client to Server)	If the endpoint requested Window Scaling/Timestamp options as specified (Server to Client)
Minimum segment size (Client to Server)	The average segment size observed during the lifetime of the connection (Client to Server)
Initial window bytes (Client to Server)	Initial window bytes (Server to Client)
The average segment size observed during the lifetime of the connection (Server to Client)	Maximum of Ethernet data bytes (Client to Server)
Maximum of total bytes in IP packet (Client to Server)	Mean of total bytes in IP packet (Client to Server)
The variance of total bytes in IP packet (Client to Server)	Maximum of Ethernet data bytes (Server to Client)

Table 9. Comparison of precision with the work of [19] during the evaluation phase

Traffic class	Fan and Liu [19]	Decision tree	Random forest	XGBoost
WWW	96.03%	99.75%	**99.97%**	99.65%
Mail	75.93%	**99.94%**	99.78%	99.83%
Bulk	69.58%	**100%**	99.40%	99.52%
Services	93.05%	99.97%	99.97%	**100%**
P2P	52.82%	92.08%	**97.91%**	94.35%
Database	83.97%	97.33%	**100%**	**100%**

Table 10. Comparison of recall with the work of Fan *et al.* during the evaluation phase

Traffic class	Fan and Liu [19]	Decision tree	Random forest	XGBoost
WWW	98.72%	99.76%	99.97%	**99.99%**
Mail	97.10%	**100%**	99.94%	**100%**
Bulk	71.91%	82.94%	**99.73%**	96.03%
Services	55.37%	**99.97%**	**99.97%**	96.69%
P2P	54.55%	93.34%	94.61%	**95.62%**
Database	51.52%	**98.98%**	98.30%	98.30%

Table 11. Comparison of f1-score with the work of [19] during the evaluation phase

Traffic class	Fan and Liu [19]	Decision tree	Random forest	XGBoost
WWW	97.36%	99.76%	**99.97%**	99.82%
Mail	85.22%	**99.97%**	99.86%	99.92%
Bulk	69.53%	90.68%	**99.57%**	97.75%
Services	69.43%	**99.97%**	**99.97%**	98.32%
P2P	53.67%	93.00%	**96.23%**	94.98%
Database	63.86%	98.15%	97.97%	**99.14%**

4. CONCLUSION

In this paper, we have considered three interpretable machine learning algorithms: decision tree, random forest, and XGBoost. These algorithms, thanks to their ability to quantify the importance of a feature in the model, have allowed identifying some essential features for a traffic classification problem. This allowed us to identify a set of 14 features that are considered the most important by most of our algorithms. Two of these 14 features were revealed to be crucial because of their high importance rates each time a model was built. These are the port number (server) and the minimum segment size (client to server). This step of reducing the features to the important ones allowed us to achieve high performance with very few features. Our 12-feature model built using the decision tree is a good example because, with only 12 of the 248 features, we obtained an overall accuracy of 99.76% in the test phase and 98.40% on a new dataset in the evaluation phase. With this model, we get better results than some existing works that use more features for less performance. This confirms once again the relevance of the study performed on the features. Let us note that XGBoost can be efficient in the classification of real-time traffic thanks to its ability to give good results even with missing data. Future studies will focus on real-time traffic classification and the effectiveness of XGBoost on this type of problem.

REFERENCES

[1] M. Zhang, W. John, K. Claffy, N. Brownlee, and U. S. Diego, “State of the art in traffic classification: A research review,” *10th International Conference on Passive and Active Network Measurement Student Workshop*, pp. 1–2, 2009.

[2] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, “Towards automated application signature generation for traffic identification,” in *NOMS 2008-2008 IEEE Network Operations and Management Symposium*, 2008, pp. 160–167, doi: 10.1109/NOMS.2008.4575130.

[3] A. Callado *et al.*, “A survey on internet traffic identification,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 3, pp. 37–52, 2009, doi: 10.1109/SURV.2009.090304.

[4] N. B. Azzouna and F. Guillemin, “Analysis of ADSL traffic on an IP backbone link,” in *GLOBECOM ’03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, pp. 3742–3746, doi: 10.1109/GLOCOM.2003.1258932.

[5] A. Madhukar and C. Williamson, “A longitudinal study of P2P traffic classification,” in *14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pp. 179–188, doi: 10.1109/MASCOTS.2006.6.

[6] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems-SIGMETRICS ’05*, 2005, p. 50, doi: 10.1145/1064212.1064220.

[7] A. W. Moore and K. Papagiannaki, “Toward the accurate identification of network applications,” in *Proceedings of the Passive and Active Measurement Workshop*, 2005, pp. 41–54.

[8] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, “Flow clustering using machine learning techniques,” in *Proceedings of the 5th International Conference on Passive and Active Network Measurement*, 2004, pp. 205–214.

[9] J. Erman, M. Arlitt, and A. Mahanti, “Traffic classification using clustering algorithms,” in *Proceedings of the 2006 SIGCOMM workshop on Mining network data-MineNet ’06*, 2006, pp. 281–286, doi: 10.1145/1162678.1162679.

[10] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, “Traffic classification on the fly,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, Apr. 2006, doi: 10.1145/1129582.1129589.

[11] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, “Unexpected means of protocol inference,” in *Proceedings of the 6th ACM SIGCOMM on Internet measurement-MC ’06*, 2006, p. 313, doi: 10.1145/1177080.1177123.

[12] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, “ACAS,” in *Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data-MineNet ’05*, 2005, p. 197, doi: 10.1145/1080173.1080183.

[13] C. Kohnen, C. Uberall, F. Adamsky, V. Rakocevic, M. Rajarajan, and R. Jager, “Enhancements to statistical protocol identification (SPID) for self-organised QoS in LANs,” in *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, Aug. 2010, pp. 1–6, doi: 10.1109/ICCCN.2010.5560139.

[14] P. Khandait, N. Hubballi, and B. Mazumdar, “Efficient keyword matching for deep packet inspection based network traffic classification,” in *2020 International Conference on COMmunication Systems and NETworkS (COMSNETS)*, Jan. 2020, pp. 567–570, doi: 10.1109/COMSNETS48256.2020.9027353.

[15] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, “Network traffic classification for data fusion: A survey,” *Information Fusion*, vol. 72, pp. 22–47, Aug. 2021, doi: 10.1016/j.inffus.2021.02.009.

[16] E. Mahdavi, A. Fanian, and H. Hassannejad, “Encrypted traffic classification using statistical features,” *isecure*, vol. 10, no. 1, pp. 29–43, 2018.

[17] V. A. Muliukha, L. U. Laboshin, A. A. Lukashin, and N. V. Nashivochnikov, “Analysis and classification of encrypted network traffic using machine learning,” in *2020 XXIII International Conference on Soft Computing and Measurements (SCM)*, May 2020, pp. 194–197, doi: 10.1109/SCM50615.2020.9198811.

[18] T. Auld, A. W. Moore, and S. F. Gull, “Bayesian neural networks for internet traffic classification,” *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 223–239, Jan. 2007, doi: 10.1109/TNN.2006.883010.

[19] Z. Fan and R. Liu, “Investigation of machine learning based network traffic classification,” in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2017, pp. 1–6, doi: 10.1109/ISWCS.2017.8108090.

[20] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, “Semi-supervised network traffic classification,” in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems-SIGMETRICS ’07*, 2007, p. 369, doi: 10.1145/1254882.1254934.

[21] Z. Li, R. Yuan, and X. Guan, “Accurate classification of the internet traffic based on the SVM method,” in *2007 IEEE International Conference on Communications*, Jun. 2007, pp. 1373–1378, doi: 10.1109/ICC.2007.231.

[22] A. Este, F. Gringoli, and L. Salgarelli, “Support vector machines for TCP traffic classification,” *Computer Networks*, vol. 53, no. 14, pp. 2476–2490, Sep. 2009, doi: 10.1016/j.comnet.2009.05.003.

[23] A. W. Moore, J. Hall, C. Kreibich, E. Harris, and I. Pratt, “Architecture of a network monitor,” *Passive and Active Measurement Workshop*, 2003, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.8563.

[24] A. Moore, D. Zuev, and M. Crogan, “Discriminators for use in flow-based classification,” *Queen Mary and Westfield College, Department of Computer Science*, no. August, pp. 1–14, 2005, doi: 10.1.1.101.7450.

[25] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” [Dataset], *University of Cambridge*, 2005. (Accessed: July 17, 2020). [Online]. Available: https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/nprobe/data/papers/sigmetrics/.

BIOGRAPHIES OF AUTHORS

Erick A. Adje was born in Cotonou, Benin on January 13, 1996. He received Engineer degree in Computer Science and Telecommunications (from Ecole Polytechnique d'Abomey-Calavi, UAC) in 2019. In December 2021, he obtained a Research Master degree in Computer Science (from Ecole Doctorale des Sciences de l'Ingénieur, UAC). Currently, he is a freelance software developer and member of the Association for the Advancement of Artificial Intelligence (AAAI) Benin chapter. His research interests include IA, machine learning and computer vision. He can be contacted at email: erickadje96@gmail.com.

Vinasetan Ratheil Houndji received a Ph.D. in Computer Science (from Université catholique de Louvain UCL, Belgium and Université d'Abomey-Calavi-UAC, Benin) in 2017 after obtaining a Master of Science degree in Computer Science (from Ecole Polytechnique de Louvain, UCL, Belgium) in 2013 and an Engineer degree in Computer Science and Telecommunications (from Ecole Polytechnique d'Abomey-Calavi, UAC) in 2011. He co-founded the company Machine Intelligence For You (MIFY) in 2017 and spent one year as Chief Executive Officer of this company. Currently, he is senior lecturer at UAC, mainly in Artificial Intelligence and Combinatorial Optimization. He is also chair of the Association for the Advancement of Artificial Intelligence (AAAI) Benin chapter. He can be contacted at email: ratheil.houndji@uac.bj.

Michel Dossou was born in Cotonou, Benin on June 12, 1982. He received the B.S. and M.S. degrees in electrical engineering from the Polytechnic faculty of Royal Military Academy (RMA), Brussels, BELGIUM in 2003 and 2006, and the Ph.D. degree in non-linear optics from the University of Lille, FRANCE in 2011 with a scholarship from the Centre National de la Recherche Scientifique (CNRS). From 2011 to 2012, he was a Research Assistant with the PhLAM (Physique des Lasers, Atomes et Molécules) laboratory of the University of Lille, FRANCE. In 2012, he joined the Polytechnic School of Abomey-Calavi, becoming Assistant Professor in 2015. Since 2019, he has been appointed Associate Professor with the Electrical Engineering Department from the University of Abomey-Calavi, Benin. He is the author of more than 30 articles. His research interests include optical fibre technology, wireless communications and broadcasting. Dr Dossou was awarded the 2006 Best M.S. dissertation Award of the Association of Engineers from RMA. He can be contacted at email: michel.dossou@epac.uac.bj.