2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics

# Featuring Real-Time imbalanced network traffic classification

SI SABER Meriem Amina
ETS
Montreal, Canada
meriem-amina.si-saber.1@ens.etsmtl.ca

BAYATI Abdolkhalegh
ETS
Montreal, Canada
abdolkhalegh.bayati.1@ens.etsmtl.ca

NGUYEN Kim Khoa
ETS
Montreal, Canada
Kim-Khoa.Nguyen@etsmtl.ca

CHERIET Mohamed
ETS
Montreal, Canada
Mohamed.Cheriet@etsmtl.ca

***Abstract*—Recently, imbalanced traffic classification has attracted more attention due to the fact that most internet traffic exhibits imbalance behavior. However, few works only have considered real-time imbalanced traffic classification. In this project, we propose a comparative study comprising several machine learning algorithms for nine different scenarios. We vary dataset and flow sizes following an under-sampling approach, in order to establish an objective evaluation of the best parameters for classification. The results showed that: 1) Combined with packet length, inter-arrival time and maximum segment size, features related to TCP session signalization enhance imbalanced traffic classification performances; 2) Ensemble approaches, especially Bagged Random Forest, achieve the best results for real-time imbalanced traffic classification; 3) Increasing flow sizes while reducing (to a certain level) training set sizes, enhances classification performances as we learn more about each individual instance. The best classification scenario includes 500 samples in each class with 8 packets flows.**

## I. Introduction

Traffic classification represents a key step for Quality of Service (QoS) guarantee mechanisms. Indeed, traffic policing strategies managing transmission rates and controlling network congestion, along with scheduling approaches installing priority traffic forwarding rules, and Traffic Engineering (TE) algorithms for QoS routing, mainly depend on the efficiency of traffic classification. While most prior work focused on Machine Learning (ML) algorithms efficiency, few studies only have considered classification in presence of data imbalance problem. Most classification approaches assume uniformity of training samples distributions while real application traffics naturally exhibit imbalanced distributions [1] [2]. An imbalanced dataset involves two types of classes: most of the traffic is contained in majority classes while only few samples are included in minority classes [3]. In this case, classification techniques tend to predict majority classes and ignore minority class instances, consequently biasing the classification results [4]. Hence, the research community has paid more attention to classification of imbalanced traffic [5]. The problems complexity increases for real-time network traffic, since network operators need to rapidly and accurately classify different applications flowing in their network, in support of their various service level agreements (SLA) [6] [7]. In this context, we established a comparative study in order to investigate the efficiency of different ML algorithms in presence of imbalanced traffic and with real-time constraints. Compared to existing comparison works, we contribute with an elaborate study on the correlation between the amount of training data in each class along with the number of packets to consider in each flow for the feature extraction process, and the ML algorithm performances. Our aim is to build an efficient traffic classification framework approaching real-time identification as a first step for efficient flow scheduling in IP networks.

The remaining of this paper is organized as follows. In Section 2, we presented some prior works on the comparison of ML based traffic classification approaches. In Section 3, we went through the different steps of the proposed classification system while browsing through some traffic identification theory. We presented and analyzed the obtained results in Section 4. In Section 5, we discussed our conclusions regarding the realized study. Finally, in Section 6, we concluded the paper and presented our future works.

## II. Related Works

Because of its impact on several IP network management mechanisms especially scheduling, traffic classification has attracted more attention. Some works focused on the sizing problem of the training data sets in order to measure the influence of training data availability on classification accuracy and computational costs. For example, [8] compared the accuracy of three supervised ML algorithms (Decision Tree, Neural and Bayesian networks) with different training dataset sizes: 5000, 20 000 and 80 000 flows per class. Bayes and Decision Tree algorithms proved to be suitable for real-time IP traffic classification. In another study, [9] paid more attention to features structure and introduced different concepts tested for smaller datasets ranging from 100, 500 to 1000 flows per class. Authors compared Hidden Nave Bayes (HNB) and KStar (K*) combined to Entropy Based Minimum Description Length (EBMDL) and Correlation Based Feature Selection

978-1-5386-7975-3/18/$31.00 ©2018 IEEE
DOI 10.1109/Cybermatics_2018.2018.00163

(CSF). They achieved accuracy improvement by combining the two classifiers even when using the smallest training data sets.

However, traffic scheduling performances also depend on the number and size of packets in each flow, since, in order to guarantee QoS resource provisioning, the classification approach needs to identify each flow quickly. This means exploiting a part of the flow statistics information only for accurate identification. Therefore, other works took a deeper look at the packets involved in feature extraction and thus in classification. In [10], authors compared two existing classification approaches, Classification based Entire Packets (CEP) and Classification based on the First Few Packets (CFFP), to the developed methods: CFFP using the first few packets after a randomly selected lifetime (Classification based Arbitrary Conjoint Few Packet (CACFP)) and Classification based Arbitrary Disjunctive Few Packets (CADFP), that considers the first randomly sampled packets that are not sequential. Results showed that CACFP and CADFP are efficient for on-line traffic classification as their accuracy approaches CFFPs, but the authors concluded that a preselection of features could reduce even more latency.

Despite the important contributions of [8] [9] [10], the data imbalance problem is nowhere to be found in the discussions of all three papers. In this context, researchers built several traffic classification works based on standard data imbalance problem resolution approaches. Some interesting comparisons were proposed. For instance, [11] summarizes several existing related works regarding the three common approaches to deal with imbalanced data: data level solutions (sampling that consist on re-balancing data distributions), algorithm level solutions (cost sensitive learning consisting on the adjustment of classifiers parameters) and hybrid algorithms combining both of the previous ones. In a more detailed study, [12] compared these approaches on large imbalanced datasets for multi class imbalance traffic classification. Results showed that cost sensitive method is the most appropriate solution followed by under sampling.

[13] on the other hand, focused on sampling approaches for a binary classification problem (IP Pear-to-Pear (P2P) traffic identification). Authors compared Synthetic Minority Over-sampling Technique (SMOTE), random under-sampling (removing some majority class samples) and random over-sampling (duplicating minority class samples) combined to two ML algorithms (C4.5 and Neural Networks). Results showed that random over sampling is the most appropriate sampling solution for imbalanced P2P classification. Other works such as [1] did not rely on these methods and proposed novel algorithms to overcome data imbalance problem. Authors in [1] presented a new classification approach based on gravitation measurement (Integrated Data Gravitation based Classification: IDGC). The solution came as an amelioration of Data Gravitation based Classification (DGC) not handling imbalanced data [14]. Thus, the paper compared IDGC integrating class distributions referred to as Amplified Gravitation Coefficient (AGC) to former DGC method [14], along with several other ML algorithms.

However, while [1] [11] [12] [13] introduced the data imbalance problem in binary and multi class traffic classification, no real-time constraint was studied. To our knowledge, very few works only have considered both the data imbalance issue along with traffic classification time constraints. The closest work to our research question is [2], focusing on ensemble algorithms. Authors emphasized the fact that there is no clear ensemble approach for real-time network traffic classification considering data imbalance and built a novel method based on a cascaded hierarchy of Decision Tree (DT) classifiers. Results showed that the proposed approach maintains the same classification performances compared to other ensemble methods in terms of accuracy but outperform all ensemble and DT methods for training and testing times proving its efficiency for on-line network traffic classification.

[2] contributed by building a method enhancing computational times and thus approaching real-time traffic classification while considering imbalanced datasets. However, establishing (n-1) trainings (where n is the number of applications) can overload the systems memory. Therefore, we present in our work another approach that consists on a comparative study involving several scenarios, in which we vary dataset and flow sizes following an under-sampling strategy. Thus, we determined the best classification parameterization in order to create an efficient environment for accurate traffic classification.

## III. Methodology And Proposed Approach

In this section, we present the followed methodology . We describe each block composing the implemented classification system as presented in Figure 1.

### A. Preprocessing

Reading Packet capture traffic trace files (Pcap files) reveals that some packets include incomplete fields, making it difficult if not impossible to compute the feature matrix. To overcome this limitation, we applied Packet Tracer (since we are using an off-line data source) and as a preprocessing step, a filter on the captured traces in order to eliminate the useless samples. This will also be possible in a real-time environment, by introducing a simple filtering strategy in each node for example.

### B. Dataset and flow size tuning

Packets are gathered into flows according to their source, destination IP addresses and ports, and protocol type as well. Some flows have a long duration, meaning statistics cannot be available for the ML algorithm until the complete transmission of the flow which contradicts the definition of early identification. Thus, we propose to investigate the impact of using different flow sizes while considering different data set sizes on the classification.

### C. Feature extraction level and directionality

For traffic classification, three feature extraction levels exist. Starting from the lowest level, packet features can be

used to measure different statistical moments such as mean packet size, and inter-arrival time variance. On a higher level, according to flow transmission direction, three flow types exist: (i) Unidirectional flows, (ii) Bidirectional flows and (iii) Full flows: characterizing the next level (connection level), it involves bidirectional flows in a period ranging from the connection establishment until its closure. Several flow level features has been used in the existing works. They involve behavioral statistics such as flow size and duration moments or statistics related to the header information, such as the number of packets with acknowledgments, with synchronization, PUSH, reset retransmission bit statistics or any Transport Control Protocol (TCP) handshake messages of our interest. Connection level variables concern statistics captured during all the transmission session such as the size of TCP advertised window or throughput distribution [15] [16] [17].

In order to fulfill the requirements of our proposed approach, we considered some packet level features along with some bidirectional flow variables. Since we are targeting real-time traffic classification expressing early identification, connection level feature extraction cannot be considered as an efficient strategy. We calculated features manually in order to be able to choose the desired variables. Thus, we provided statistical moments of packet lengths, inter-packet lengths, and packet inter-arrival times. We also measured statistical moments of TCP messages because they constitute an important part of the retrieved information and differentiate rapidly TCP applications from User Datagram Protocol (UDP) ones. We measured the number of acknowledgment messages (ACK), PUSH, Maximum Segment Size (MSS), FIN and number of retransmissions in order to capture sensitive applications. More details can be found in the next section.

### D. Feature selection

Feature selection module filters the significant features mostly to reduce training time [18]. In this work, we studied the impact of feature importance in each scenario using Random Forest. We will show later in the result section that feature ranking changes in each situation. Although it has widely been used in classification, thanks to its tree-based strategy, Random Forest has proved its efficiency in feature selection since it automatically ranks features according to their importance. The algorithm relies on a metric referred to as Gini impurity [1]. We start by deploying a tree where the chosen feature in each node reduces impurity with a certain amount hence trees start from nodes with the most decrease in impurity and end with the ones having the least decrease in impurity. Averaged for all trees, the metric is called Gini impurity and is measured according to equation (1), where, for each feature $\theta$, $\Delta_\theta(\tau, T)$ represents the Gini impurity reduction at each node $\tau$ in each Tree $T$.

$$I_G(\theta) = \sum_T \sum_\tau \Delta_\theta(\tau, T) \quad (1)$$

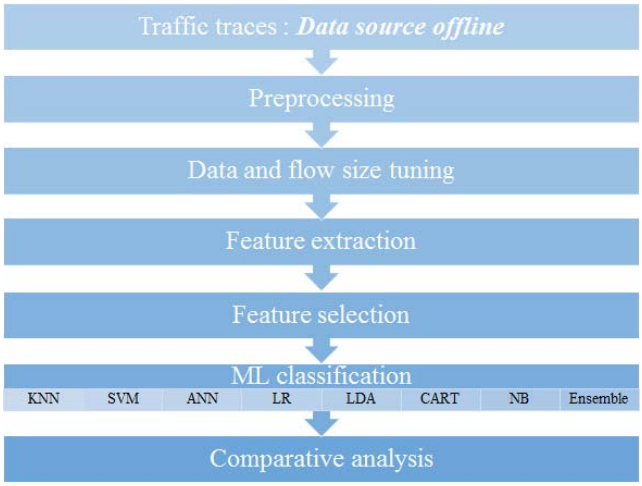


Fig. 1. Classification system

### E. ML classification

We chose to compare, based on literature, the most used ML algorithms for IP traffic classification and study their efficiency for real-time identification in presence of imbalanced data compared to ensemble algorithms. Thus, we compared K Nearest Neighbor (KNN), Support Vector Machine (SVM), Artificial Neural Network (ANN), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART), and Nave Bayes (NB) along with Voting, Bagged Decision Tree (BDT), Bagged Random Forest (BRF), Adaboost and Stochastic Gradient Boosting (SGB) ensemble methods. Because of the limited space, we will not introduce the deployed algorithms. However, detailed explications can be found in these references [2] [19] [20] [21] [22] [23] [24].

## IV. Results Analysis

Before analyzing the obtained results, we describe the validation metrics used to verify the efficiency of each ML algorithm as well as the data set used in the definition of the several scenarios tested.

### A. Evaluation metrics

When dealing with imbalanced data, accuracy does not measure the correct classification performances as it mostly concerns majority classes. Instead, alternative metrics related to the confusion matrix and thus depending on four simple measures: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), can be found. In a traffic classification context, TP, TN, FP and FN are defined as follows:

- True Positive: The amount of samples predicted as class A when they truly belong to class A.
- True Negative: The percentage of samples not belonging to A and not classified as A.
- False Positive: The percentage of samples not belonging to A while classified as A.

- False Negative: The amount of samples that are not classified as A while they belong to class A.

In presence of imbalanced classes, two performance metrics are commonly used: Geometric-mean (G-mean) and F-measure. G-mean parameter maximizes individual classs accuracies and is measured by equation (2), where $Recall = \frac{TP}{FN+TP}$ and $Specifity = \frac{TN}{TN+FP}$. F-measure on the other hand, is a weighted mean of Precision and Recall following equation (3) where $Precision = \frac{TP}{FP+TP}$ and $\beta$ is the parameter fixing the trade-off between Precision and Recall. When $\beta < 1$, precision is more significant than Recall and the reverse situation applies for $\beta > 1$. For $\beta = 1$, F-measure referred to as $F1$ measure equally weights Precision and Recall [24].

$$G - mean = \sqrt{Recall * Specificity} \quad (2)$$

$$F - measure = \frac{(1+\beta)^2 * Precision * Recall)}{\beta^2 * Precision * Recall} \quad (3)$$

Our work focuses on multi class traffic identification, in order to be able, in the future works, to schedule flows entering network nodes. Thus, since G-mean focuses on both positive and negative class accuracies, we chose to measure F1 metric focusing on TPs. We also included training and testing times, in order to measure the efficiency of the tested scenarios for real-time traffic classification. The reason for measuring training time where testing time is a more representative metric for classification computational times is that some classifiers exhibit important training times, others even present retraining periods decreasing the classifiers ability to tackle real-time traffic classification.

### B. Data sets

For our experiments, we chose UNIBS data set [25], which is one of the few labeled databases available for public. It includes traffic traces captured during 3 working days on the edge router of Brescia University campus network. We selected UNIBS dataset traffic because it was captured in consecutive days, which makes it the appropriate candidate for the performance validation of real-time traffic classification. In addition, as shown in table I during the three days of capture, traffic traces exhibit imbalance distributions where majority classes include HTTP, SNMP, Skype, Bit Torrent and SSL protocols. However, in order to distinguish our study from existing ML algorithm comparisons using UNIBS, we also proposed to perform the comparison study using only the first two days traces compared to most works including the three days and others only one day (as for [3], using the first day for training and the rest for testing) emphasizing the relationship between classification performances and available traffic traces. UNIBS traces include 6 traffic classes: Web services involving HTTP, HTTPS requests (class 0), Mail packets with POP3, SNMP, IMAP3 protocol information (class 1), class 2 represents Skype packets, peer to peer class for Bit Torrent (class 3) and for Edonkey packets (class 4) and the last class concerns the rest of the packets including MSN messages, FTP and SSH, referred to as others.

TABLE I
NUMBER OF FLOWS FOR EACH PROTOCOL IN UNIBS TRACES

| protocols | unibs 20090930 | unibs 20091001 | unibs 20091002 |
|---|---|---|---|
| HTTP | 12984 | 15685 | 17390 |
| SNMP | 120 | 132 | 42 |
| POP3 | 185 | 253 | 337 |
| IMAP | 31 | 42 | 5 |
| MSN | 0 | 6 | 0 |
| RTP | 12 | 0 | 0 |
| Skype | 459 | 448 | 1402 |
| SSH | 17 | 6 | 4 |
| Bit Torrent | 4245 | 1499 | 608 |
| Edonkey | 205 | 6 | 397 |
| SSL | 1329 | 1915 | 1639 |

### C. Performance evaluation

Since the data imbalance problem is defined as inequitable class distributions, we investigated how varying data set sizes and thus reducing the gap between classes number of samples, influences classification performances. Thus, we tested the 12 algorithms when limiting, following a down-sampling approach, the data set to N=1000, N=500 and N=100 samples for each class, after observing the number of traffic instances in each class. For each scenario we built, we constructed bidirectional flows of k packets, where k varies between 2, 5 and 8 per flow (according to [26] [27], in order to build a timely classifier. We will expose in this section, the obtained results for different scenarios. As mentioned earlier, we measure F1 metric as well as training and testing time for each classification strategy. Due to space limitation, we will only present the results related to what we considered the best scenario. However, we will detail the results for each scenario in the paragraphs below.

Scenario 1: In the first scenario (N=1000, k=2), data is still quite imbalanced and the information we use to characterize each flow is small. As a first observation, for standard algorithms, we note a better precision when using SVM, KNN and CART, especially KNN, compared to LDA, ANN and LR having trouble identifying class 5 (most minority class). However, almost all the ensemble algorithms (except for Adaboost) show good results, especially for voting, predicting class 5 with more than 85% of F1. For the training time, we observe as expected that SVM and ANN present high values compared to the other standard algorithms and that ensemble approaches exhibit higher training times. The worst results (in terms of training time), were obtained for SGB with almost 6 seconds, since we had to go to 500 trees in order to obtain a good F1 value.

Scenario2: Compared to the previous scenario, we observe with (N=1000, k=5), better F1 values for almost all classifiers and for all classes with training time decrease. Although computational speeds are not satisfying yet, results show that by increasing the flow sizes to 5, we obtain good classification performances while decreasing training times. In this case,

the best compromise is obtained with BDT, BRF, voting and CART followed by SVM.

Scenario 3: With (N=1000, k=8), we also see that almost all classifiers succeed in class 5 prediction and that the optimal results are obtained by ensemble algorithms while the others approach these optimal observations, especially CART. Although class 5 identification is performed with better F1 using ensemble algorithms at the previous scenario, we chose to designate the present scenario as the best one (so far) since F1 parameter exhibits better values for most classes using all classification algorithms.

Scenario 4: For (N=500, k=2), the first observation we make, is the overall F1 enhancement (for almost the twelve ML algorithms) compared to the first scenario (N=1000, k=2). Also, we observe that scenario 3 fits standard algorithms while scenario 4 is more adequate for ensemble approaches. Based on training and testing times, which as expected reduced with data set size reduction, the current scenario is the best one for now. Also, SGB remains the worst classification algorithm in terms of training time.

Scenario 5: For the fifth scenario (N=500, k=5), we observe the best class 5 identification results as it is identified by almost all classifiers even Adaboost, actually exhibiting its best F1 value in the five scenarios. However, the classifiers are not yet stable, since we observe variations of the F1 measure over several classes. As for classification speeds, results show more efficient classification times but less accurate training times since they present some variations for most algorithms because of flow size increase.

Scenario 6: For (N=500, k=8), classifiers are more stable as they mostly succeed in the minority class identification as well as other classes F1 measure enhancements. Also, while we observe an increase of a few milliseconds in training times (10 in the case of SGB), classification time has decreased drastically especially for the ensemble algorithms, representing at this time the best compromise between F1 performances and training and testing times.

Scenario 7: In (N=100, k=2), we observe that F1 measure maintains a good level for some algorithms mostly ensemble, while it decreases for the standard algorithms, proving the robustness of ensemble approaches in the presence of imbalanced data even with a small size training sets. As for training times, we observe a clear enhancement compared to the past scenarios, but a tens milliseconds increase in classification times.

Scenarios 8, 9: Almost the same observations are noticed for the next scenarios ((N=100, k=5), (N=100, k=8)). We find that for minority classes, F1 values has increased. However, majority class classification performances has decreased, meaning the chosen dataset and flow size does not highlight the packets characteristics.

Finally, ensemble algorithms exhibit the best classification performances, especially Bagged Random Forest as it represent a good compromise between F1 measure enhancement and training and testing time reduction. Moreover, data training sizes affects training time while flow sizes affect both

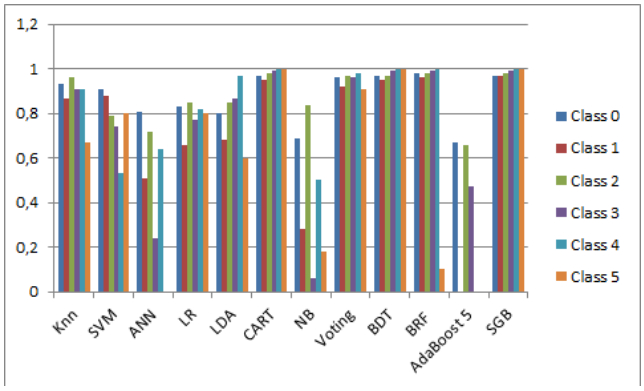


Fig. 2. Precision comparison for scenario 6: N= 500, k=8: Best scenario

training and testing time. Thus, according to results, increasing flow sizes while reducing to a certain level training set sizes, enhances classification performances as we learn more about each sample. The best classification scenario is then the sixth, including 500 samples for each class with 8 packets in each flow, as it represents a good trade off between efficient and rapid traffic classification. Results are presented in fig.2 and table II.

As for feature selection, we present in fig.3, fig.4 and fig.5 the results obtained in three scenarios (chosen as examples) including the best one (N=500, k=8). For N=100, especially k=2, we notice the importance of SACK (Selective acknowledgment) feature representing the number of TCP lost segments. This value explains the important training times and the modest F1 values. Consequently, we can confirm that the first three scenario configurations are not efficient. However, while SACK values are also important for N=500 and scenario 6 (N=500, k=8) particularly, we notice classification performance enhancement, which is explained by the introduction of features related to TCP signalization characteristics such as retransmission count, PUSH: pushing packets to transmission or to receiving applications and FIN closing the TCP connection. In addition, it is worth noting that the metrics associated to TCP sessions are less present when choosing a small number of packets in the flow because of the short possibility of finding all signaling messages in a small flow. For more information about the different features we chose to include in our study, please refer to [28].

## V. Discussion

Results can be summarized in the following important conclusions. First, although we clearly observed that packet length, inter-arrival time and maximum segment size are sufficient features for good classification performances, better F1, training and testing time values are obtained in presence of TCP signalization related features especially because TCP packets represent an important part of Internet traffic. KNN and CART show interesting classification results while ensemble methods outperform all classification approaches. However, algorithm parameter tuning represents a critical step

TABLE II
COMPUTATIONAL TIME COMPARISON BETWEEN ML ALGORITHMS FOR N=500 SAMPLES FOR EACH CLASS AND K=8 (SCENARIO 6)

| Algorithms | Training time | Testing time |
|---|---|---|
| KNN | 0 | 0,0099 |
| SVM | 0,212 | 0,0199 |
| ANN | 0,393 | 0 |
| LR | 0,06 | 0,0099 |
| LDA | 0,0099 | 0 |
| CART | 0,0099 | 0 |
| NB | 0,002 | 0,003 |
| Voting | 0,23 | 0,0099 |
| BDT | 0,28 | 0,0099 |
| BRF | 0,0599 | 0 |
| AdaBoost 5 | 0,044 | 0,003 |
| SGB | 2,62 | 0,0099 |

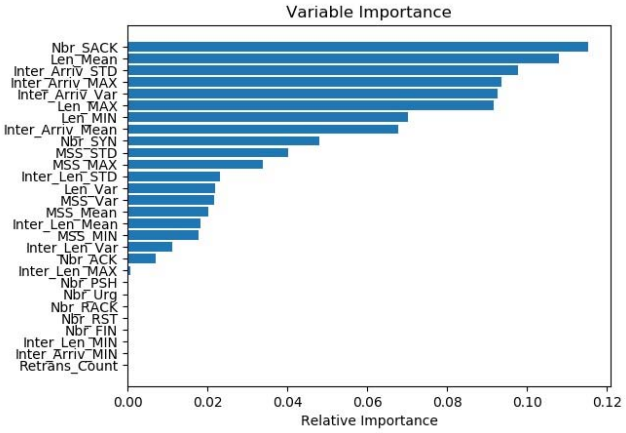


Fig. 3. Feature importance for scenario: (N=100, k=2)

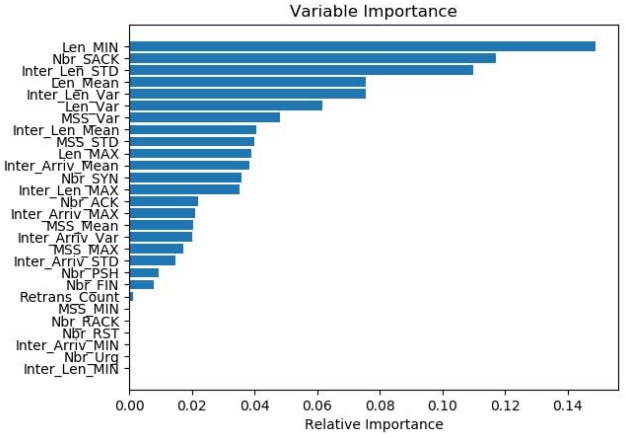


Fig. 4. Feature importance for scenario: (N=500, k=8): Best scenario

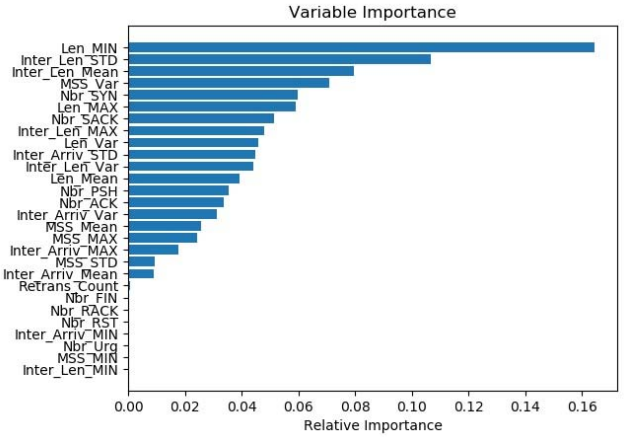


Fig. 5. Feature importance for scenario: (N=1000, k=5)

as it affects training and testing times. Also, Random forest proved to be a good feature selection approach as it produced accurate feature sets (according to F1 results) in a flexible and quick manner especially for small data set sizes. However, the number of trees needs to be efficiently tuned otherwise it can increase the training time. In the same context, even though more TCP packet features were available, since random forest extracted in most scenarios packet length and inter-arrival statistics, TCP and UDP packets were classified in the same conditions. Finally, reducing dataset sizes promotes for a certain flow size early traffic identification and presents a good alternative for the existing methods solving the data imbalance problem.

## VI. CONCLUSION AND FUTURE WORKS

While most classification studies focused on traffic early identification and how to meet the strict needs of existing applications and anticipate future ones, data imbalance constraint is often ignored, resulting in performance bias. In this paper, we merged the necessity of rapid classification with data variability by performing a comparative study including nine scenarios varying dataset sizes and flow sizes. With this work we aim to be able to anticipate classification preprocessing and reduce computation times. For future works, we will include more recent traffic applications, propose a Bagged Random Forest based classification approach and apply the resulting algorithm in an SDN context (data center inter connection). Later, we plan to build a scheduling approach considering the traffic imbalance problem and real-time constraints.

## ACKNOWLEDGMENT

This work is supported in part by NSERC and Ciena for the project CRDPJ 461084. It also received support from the Canada Research Chair, Tier 1, held by Mohamed Cheriet. Finally, we would like to thank Ana Carolina Riekstin for her continuous help and precious advices.

## REFERENCES


[1] L. Peng, H. Zhang, Y. Chen, and B. Yang, "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," *Computer Communications*, vol. 102, pp. 177–189, 2017.

[2] S. E. Gómez, B. C. Martínez, A. J. Sánchez-Esguevillas, and L. H. Callejo, "Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal," *Computer Networks*, vol. 127, pp. 68–80, 2017.

[3] H. Wei, B. Sun, and M. Jing, "Balancedboost: A hybrid approach for real-time network traffic classification," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*. IEEE, 2014, pp. 1–6.

[4] L. Peng, H. Zhang, B. Yang, and Y. Chen, "A new approach for imbalanced data classification based on data gravitation," *Information Sciences*, vol. 288, pp. 347–373, 2014.

[5] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, 2018.

[6] W. Li, K. Abdin, R. Dann, and A. Moore, "Approaching real-time network traffic classification," Tech. Rep., 2013.

[7] T. S. Tabatabaei, F. Karray, and M. Kamel, "Early internet traffic recognition based on machine learning methods," in *Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*. IEEE, 2012, pp. 1–5.

[8] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Performance Evaluation*, vol. 67, no. 6, pp. 451–467, 2010.

[9] R. Raveendran and R. R. Menon, "A novel aggregated statistical feature based accurate classification for internet traffic," in *Data Mining and Advanced Computing (SAPIENCE), International Conference on*. IEEE, 2016, pp. 225–232.

[10] S. Zhao, Z. Chen, L. Peng, X. Yu, and B. Yang, "Online traffic classification based on few sampled packets," in *Communication Technology (ICCT), 2012 IEEE 14th International Conference on*. IEEE, 2012, pp. 1182–1187.

[11] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 42–47, 2012.

[12] Q. Liu and Z. Liu, "A comparison of improving multi-class imbalance for internet traffic classification," *Information Systems Frontiers*, vol. 16, no. 3, pp. 509–521, 2014.

[13] W. Zhong, B. Raahemi, and J. Liu, "Learning on class imbalanced data to classify peer-to-peer applications in ip traffic using resampling techniques," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 3548–3554.

[14] L. Peng, B. Yang, Y. Chen, and A. Abraham, "Data gravitation based classification," *Information Sciences*, vol. 179, no. 6, pp. 809–819, 2009.

[15] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[16] H. R. Loo, S. B. Joseph, and M. N. Marsono, "Online incremental learning for high bandwidth network traffic classification," *Applied Computational Intelligence and Soft Computing*, vol. 2016, p. 1, 2016.

[17] T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive ip traffic," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 6, pp. 1880–1894, 2012.

[18] Y. Huang, Y. Li, and B. Qiang, "Internet traffic classification based on min-max ensemble feature selection," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 3485–3492.

[19] A. Pektaş and T. Acarman, "Identification of application in encrypted traffic by using machine learning," in *International Conference on Man–Machine Interactions*. Springer, 2017, pp. 545–554.

[20] S. Vanaja and K. Rameshkumar, "Performance analysis of classification algorithms on medical diagnoses-a survey," *Journal of Computer Science*, vol. 11, no. 1, pp. 30–52, 2015.

[21] L. Li, J. Zhang, Y. Zheng, and B. Ran, "Real-time traffic incident detection with classification methods," in *International Conference on Green Intelligent Transportation System and Safety*. Springer, 2016, pp. 777–788.

[22] D. Wu, X. Chen, C. Chen, J. Zhang, Y. Xiang, and W. Zhou, "On addressing the imbalance problem: a correlated knn approach for network traffic classification," in *International Conference on Network and System Security*. Springer, 2014, pp. 138–151.

[23] Z. Wang, "The applications of deep learning on traffic identification," *BlackHat USA*, 2015.

[24] W.-J. Lin and J. J. Chen, "Class-imbalanced classifiers for high-dimensional data," *Briefings in bioinformatics*, vol. 14, no. 1, pp. 13–26, 2012.

[25] "UNIBS: Data sharing kernel description," http://netweb.ing.unibs.it/ ntw/tools/traces/, accessed: 2002-06-23.

[26] 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, 2007.

[27] M.-h. HONG, R.-t. GU, H.-x. WANG, Y.-m. SUN, and Y.-f. JI, "Identifying online traffic based on property of tcp flow," *The Journal of China Universities of Posts and Telecommunications*, vol. 16, no. 3, pp. 84–88, 2009.

[28] A. Moore, D. Zuev, and M. Crogan, "Discriminators for use in flow-based classification," Tech. Rep., 2013.