An autonomic traffic analysis proposal using Machine Learning techniques

Fannia Pacheco, Ernesto Expósito, Mathieu Gineste, Cedric Budoin

An autonomic traffic analysis proposal using Machine Learning techniques

Fannia Pacheco
IUNIV PAU & PAYS ADOUR, LIUPPA
ANGLET, France
f.pacheco@univ-pau.fr

Ernesto Exposito
IUNIV PAU & PAYS ADOUR, LIUPPA
ANGLET, France
ernesto.exposito-garcia@univ-pau.f

Mathieu Gineste and Cedric Budoin
Thales Alenia Space
TOULOUSE, France
{mathieu.gineste,cedric.baudoin}@thalesaleniaspace.com

ABSTRACT

Network analysis has recently become in one of the most challenging tasks to handle due to the rapid growth of communication technologies. For network management, accurate identification and classification of network traffic is a key task. For example, identifying traffic from different applications is critical to manage bandwidth resources and to ensure Quality of Service objectives. Machine learning emerges as a suitable tool for traffic classification; however, it requires several steps that must be followed adequately in order to achieve the goals. In this paper, we proposed an architecture to perform traffic analysis based on Machine Learning techniques and autonomic computing. We analyze the procedures to perform Machine Learning over traffic network classification, and at the same time we give guidelines to introduce all these procedures into the architecture proposed. The main contribution of our proposal is the reconfiguration of the traffic classifier that will change according to the knowledge adquired from the traffic analysis process.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; • **Networks** → **Traffic engineering algorithms**; • **Computer systems organization** → *Self-organizing autonomic computing*;

KEYWORDS

Machine Learning, traffic analysis, quality of service, autonomic computing

ACM Reference Format:
Fannia Pacheco, Ernesto Exposito, and Mathieu Gineste and Cedric Budoin. 2017. An autonomic traffic analysis proposal using Machine Learning techniques. In *MEDES '17: MEDES '17: The 9th International Conference on Management of Digital EcoSystems, November 7–10, 2017, Bangkok, Thailand.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3167020.3167061

MEDES '17, November 7–10, 2017, Bangkok, Thailand

ACM ISBN 978-1-4503-4895-9/17/11…$15.00
https://doi.org/10.1145/3167020.3167061

1 INTRODUCTION

Network analysis has recently become in one of the most challenging tasks to handle due to the rapid growth of communication technologies. Network analysis is important due to several reasons: a) Troubleshooting tasks: the main objective is to locate faulty network devices, device/software misconfigurations, measure delays along a path, locate packet loss points, network errors, etc. b) Security: avoid malware, prevent intrusion to private information. c) Quality of Service (QoS): is defined as the overall acceptability of an application or service perceived by end-users. For network management, accurate identification and classification of network traffic is a key task. In particular, identifying traffic from different applications is critical to manage bandwidth resource and to ensure QoS objectives.

Traffic classification uses different tools and methodologies to classify network traffic categorized as [2]: port based, payload inspection, behavioral techniques and statistical approaches. Port based classification techniques are now considered obsolete given the frequent obfuscation techniques, and dynamic range of ports used by applications. On the other hand, packet payload inspection methods remain relevant primarily due to their high classification accuracy, where Deep Package Inspection (DPI) is one of the most popular. DIP tools analyze the content of the packets by searching for specific patterns (i.e., signatures). DPI became one of the fundamental traffic analysis methods to perform traffic classification, network management, intrusion detection, and network forensics. However, DPI fails when privacy policies, and laws prevent access to the packet's content besides encryption, protocol obfuscation or encapsulation [19]. In contrast, behavioral classification techniques analyze total traffic patterns of the endpoints (hosts and servers) such as the number of machines contacted, the protocol used, and the time frame of bidirectional communication to identify the application being used on the host. Behavioral techniques can deal with encrypted packets; however, it requires the number of flows to be collected and analyzed which are not necessarily available [6]. Finally, statistical techniques use statistical features extracted from network flows to classify applications. The classification is commonly performed by using Machine Learning (ML) models, which are considered lightweight and highly scalable from an operational point of view. Additionally, different types of encrypted traffic can be identified by using the statistical based approach [19].

ML has been recently applied for traffic network classification, and it aims at improving several aspects such as troubleshooting tasks, security and Quality of Service (QoS) in networks. The survey

in [13] presents an overview of several works that uses machine learning for IP traffic classification. The common point between these works is their methodology, which is characterized by: i) Construction of a historical dataset through a feature extraction process. ii) Training and test the ML model. For this case, a classification model is chosen along with a supervised or unsupervised learning algorithm. The model is tested with a dataset taken from the original historical data for this purpose. In [16], a more recent work shows a detailed survey of the methods used to extract adequate features for HTTP-based botnet traffic. Such features or identifiers are the inputs to many detection mechanisms such as ML techniques, regardless of the methodology used.

For traffic classification, normally, supervised learning is used to classify traces where the most common algorithms deployed are: decision trees [1, 2, 10] , Support vector machines (SVM) [17], Random forest [1], Naïve Bayes [1, 10], and among others. However, k-means (an unsupervised technique) has been widely used to cluster the traffic into categories or application protocols [6, 9, 21].

On the other hand, autonomic computing aims at developing technologies that are able to change, adapt and manage themselves automatically [5]. Such activities can be performed based on techniques such ML.

Some of the ML solutions in traffic network work standalone to solve specific problems, and suffer drawbacks caused by big data management and integration such as [4, 20]. Particularity, these solutions do not integrate autonomic processes that may change the configuration of the ML solutions for improving the results. In order to cope with all these requirements and constrains, this work presents the traffic network classification using ML, along with autonomic computing practices for its implementation. Our contribution are:

- A proposal of an architecture for traffic analysis based on ML
- An autonomic system that will be able to train and test ML models for traffic classification. The autonomic system will be in charge of selecting the best ML model and classify new patterns, as well as scheduling reconfiguration of the ML solution.

A benchmark for traffic classification was used to perform a complete study with ML in order to offer results that may change and improve the QoS. Four different machine learning models were trained and tested. Finally, an example of how the ML process can be implemented is presented. The mapping between the experiments and the autonomic system proposed is remarked along the study.

2 BACKGROUND

This section presents a brief introduction of ML. Following, the basis of traffic analysis and how it influences the QoS. Finally the autonomic computing principles are described.

2.1 Machine Learning

Nowadays, Machine Learning (ML) techniques are a very popular approach to identify and classify patterns in different fields of science. Its main objective is to give the computer automatic learning capabilities, where the machines are able to extract knowledge from a process under certain conditions. The knowledge extraction is handled by a ML model, which in turn is built with historical experiences recorded from the case study. Generally speaking, the ML model should give the current state of the process/case study given a number of incoming inputs. However, such models are not always accurate, and work only under certain defined conditions. In brief, the construction of ML models follows the steps in Figure 1. Normally, an historical dataset from the process is stored and treated to extract features that better characterize the case study. Following a feature selection process, that discriminate between the most important features, may be used. Finally the model is trained and tested with a partition of the historical data for each task.

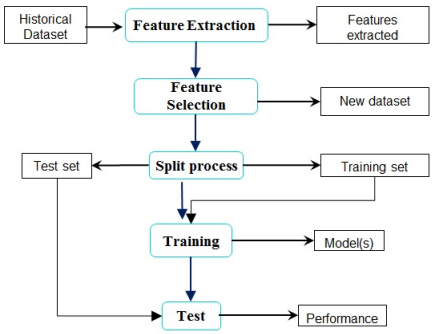

Figure 1: Steps to achieve the ML process.

The feature extraction process is one of the most important steps due to it allows measuring or computing the features that might give information about the state of the process. Normally, this phase has to be carefully studied by experts in the case study field, and might determinate part of the model accuracy. On the other hand, the ML model and the type of learning is associated with the type of problem to solve and with the expertise given in the field.

One branch of the ML models are divided into into parametric and nonparametric. As an example for parametric models, the learning process aims at defining the parameters of the model, this is achieved using a learning algorithm and the historical data. There are two types of learning, i.e., supervised and unsupervised learning. Basically, the supervised learning algorithms adjust the model parameters minimizing the error between the model output and the real expected output for an input. This means that the historical data has to contain the inputs and the outputs of the process, in this context each sample or raw in the dataset is called labeled data. On the other hand, unsupervised algorithms try to find relationships between the inputs without beforehand knowledge of the outputs, and the raw samples are defined as unlabeled samples. These relationships can be similarities, proximities, and statistical relationships, among others [18].

As a derived consequence of the learning process, the supervised algorithms are commonly used to perform classification of patterns, while the unsupervised ones are rather used to cluster inputs in order to find anomalous or similar behaviors between themselves.

2.2 Quality of Service and Traffic classification

Quality of Service (QoS) handling is the most important part of the Satellite radio resource management. Satellite networks have

limited radio and transmission resources and need to strictly schedule the utilization of radio and transmit resources using different granularity of class of service to provide and ensure QoS for the IP/Ethernet traffic. QoS involves a combination of functions/ mechanisms such as: Packet scheduling (queuing), traffic classification, traffic policing and shaping, active queue management, resource reservation/provisioning and admission control.

Quality of Service (QoS) represents a group of requirements designed to provide communication functions between peers in the network [15]. QoS can be also viewed as a network performance by evaluating everything that is occurring across the network. This is achieved through the evaluation of parameters related to delay, data loss, available bandwidth, etc. Although, the user is not aware of what is internally happening in the network; packet loss or delay might affect his/her Quality of Experience (QoE). Different applications have different requirements regarding the QoS parameters, and they are categorized according to it. For instance, video applications might tolerate certain degree of package loss but not delay. In [7] a group of applications is presented as follows:

- Conversational voice and video
- Command/control (e.g. Telnet, interactive games)
- Voice/video messaging
- Transactions (e.g. E-commerce, WWW browsing, Email access)
- Streaming audio and video
- Messaging, Downloads (e.g. FTP, still image)

Although, this list still keeps up to the date, it can be found different categorizations that integrate newly incoming applications. One of the main goals of service providers is to improve the QoE of theirs clients, this is achieved through the improvement of the QoS by using network analysis. Traffic classification accomplishes prioritization by dividing similar types of traffic. Figure 2 summaries the most common approaches found in the literature focusing on the ML branch. Unencrypted traffic can be treated by payload inspection, behavioral based, statistical based and ML techniques; while for encrypted only by the three last ones. The feature extraction process performed is mainly statistical based, however, several approaches propose studying the time-series behavior and graph representation of the flows as well as the construction of bag of flows. The ML algorithm can be supervised, unsupervised or semi-supervised and it will depend on the ML task to perform. The ML task is directly linked with the objective of the study.

In this sense, one objective could be to differentiate traffic such as e-mail, streaming video, voice, large document file transfer, into classes. For QoS objectives identify this traffic in a flow can help to define different levels of priority, such as those for throughput and packet loss, to each group, and thereby control traffic behavior.

2.3 Autonomic Computing

Autonomic computing principles are based on the human nervous system, which basically monitor, control and regulate automatically different parts of the human body. The nervous system main goal is to assure homeostasis, which in turn aims at maintaining a balance between internal and external interactions through different regulation mechanisms. Autonomic computing tries to map the nervous system autonomic behavior to the computing domain [14].

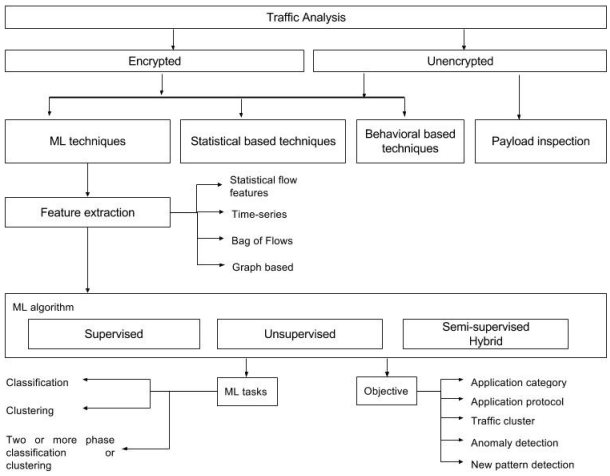

Figure 2: Topology with some of the traffic analysis approaches.

From the computing point of view, autonomic computing aims at developing technologies that are able to change, adapt and manage themselves automatically. Such activities can be performed based on previous observation or learning from the context where the autonomic system is deployed [5].

Briefly speaking, an autonomic computing architecture is composed of touchpoints for managed resources, knowledge sources, autonomic managers and manual managers. The touchpoints allow the communication between the components using sensor/effector modules. The autonomic managers are dedicated to specific tasks, a control loop is proposed to give this component autonomic features as it is shown in Figure 3. Manual managers provide an interface to communicate with the users, and to also change the parameters of the autonomic managers.

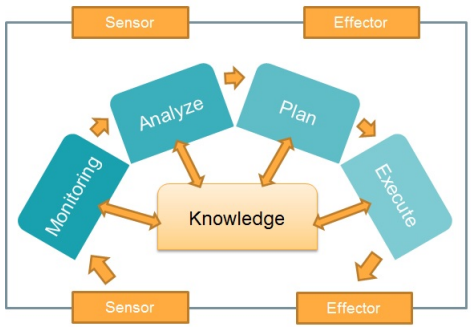

Figure 3: Autonomic blocks in an autonomic computing architecture [5].

In this paper, we present an initial approach of how autonomic components can be integrated into a specialized architecture for traffic network analysis. Additionally, these components use some of the techniques exposed in Figure 2.

3 PROPOSAL

Particularly, this work tries to propose an automatic and logic process to analyse traffic for improving the QoS. The architecture proposed is shown in Figure 4. The architecture is based on the autonomic computing principles, and it is conformed of the following elements:

- Self-optimizing manager: this component is in charge of providing a complete loop that will allow achieving the QoS objectives in an autonomic manner. As it is shown in Figure 4, it integrates elements from the traffic analyzer manager and elements from the QoS analyzer manager. The knowledge used to close the loop is considered as the features and models that must be used to perform the prediction, as well as, the QoS requirements. The knowledgebase is stated by Orchestrating system manager which evaluates the performance of this manager.
- Traffic analyzer manager: In this component, the state of the traffic at a monitoring point is studied. This module is composed of the Monitoring and Analizer steps, which in turn are internally divided into different steps based on the ML process to discover knowledge. The general process consists on taking traffic (flow traces) at a monitoring point in the network in order to forward this information to the autonomic engine. The flow traces are processed to extract features that characterize their behavior, these features can be reduced through a feature selection process, or other features can be generated with them. Following, the flow traces are analyzed by a ML model that will allow predicting the type of traffic monitored.
- QoS manager: it is in charge to manage and control access to network resources. This module evaluate the results provided by the traffic analyzer module in order to plan and to execute actions, that will reconfigure the network at the monitoring point.
- Orchestrating system: This manager evaluates the results of the self-optimizing system in order to improve its performance. Different traffic analysis techniques can be used to classify or discover patterns. The proposal will be mainly based on ML, but it can also be supported by other techniques such as payload inspection. This module acknowledges the advantages and disadvantages of the techniques selected in order to modify the knowledge of the Self-optimizing manager according to the conditions given at the monitoring point and the results perceived. At an initial stage, this manager does not have sufficient information to set the knowledge of the lower manager; for such as case, historical datasets are processed and studied to define the parameters needed.

The present work is only focused on the traffic analyzer manager and the orchestrating system at its initial stage. The QoS manager is a component considered as already defined and, in consequence, out of our scope. We present as follows the most important characteristics of such modules.

3.1 Orchestrating System

Different traffic analysis techniques can be used to classify or discover patterns. As an initial setting, we proposed to study historical data from the process in order to build several features and classifiers. The best configuration feature-classifier is selected and set in the knowledge of the self-optimizing manager.

In brief, to build classifiers, the process presented in Figure 1 is replicated as many times as required. We will describe above the most important elements that allow this component to work for traffic classification.

- Feature extraction
  It will transform the historical dataset through a feature extraction process. The features extracted from the flows are mainly statistical based features, which are defined under the assumption that traffic at the network layer has statistical properties (such as the distribution of flow duration, flow idle time, packet inter-arrival time and packet lengths). These properties are unique for certain classes of applications, and enable different source applications to be distinguished from each other. Several approaches try to define a methodology for the feature extraction process in traffic network; the most popular statistical based are described by [12]. 249 features are detailed and they are widely used to traffic analysis.
- Feature selection
  The aim of the feature selection process is to determinate which of the statistical features are significant to perform the traffic classification. For instance, we will use two approaches and the combination of both, that is:
  - Correlation analysis (corr): It was already established that the feature extraction is performed by computing statistical parameters, which come from the flow behavior. Therefore, it is likely to find correlated attributes that contribute with the same information to the study problem. The correlation analysis can be executed over the original dataset in such a way that one of the attributes pair with a correlation higher than a threshold can be removed.
  - Random forest (RF): it is an algorithm based on decision trees that presents a huge improvement in terms of precision. RF is suitable for obtaining the most representative attributes, since it computes the information degree contributed by each attribute to the classes.
- Training and test ML models
  In the present study, we will focus only in supervised methods for classification of the traces. Particularly, the classification will be performed in two stages: 1) category classification of the flows and 2) application classification. The category classification is the most important one due to with this information the QoS manager can evaluate and modify the state of the network. As an example, real-time applications belonging to the *Streaming audio and video* category should be prioritized to improve the QoS.
  The application classification can be considered as additional information for the experts. Moreover, we propose to construct an application classifier for each category; for instance, when the category *Streaming audio and video* is predicted,

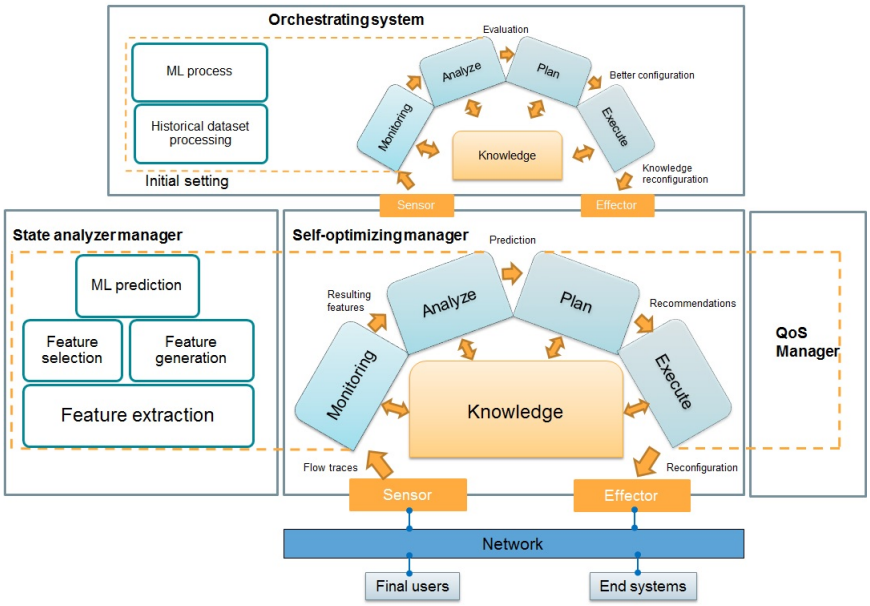

Figure 4: Autonomic traffic analysis architecture.

a model for this type of application is used and it predicts more accurately the name of the application.
Four ML algorithms were selected to obtain traffic network classification models; that is, Random Forest(RF), decision trees(DT), Support Vector Machine(SVM) and K Nearest Neighbors(KNN). Such models are trained with 70% of the data, with the total number of features and the new features resulting from the feature selection process.
The evaluation of the models will be performed by predicting the class of each sample in the test set. A performance metric is selected, and it is defined as the accuracy of the classifier. In general, the accuracy counts the number of flows that were classified correctly divided into the total of flows.

We will denote as the best combination pair $\{F, M\}$ as the knowledge reconfiguration set in the Self-optimizing manager (see Figure 4).

Once the initial knowledge is established, this manager could study new historical dataset for further evaluations and reconfiguration. Moreover, it could add more parameters to fine its results, such as performance metrics from the QoS manager.

3.2 Traffic analyzer manager

This module will collect in an on-line manner the flow traces passing trough a monitoring point. The processes proposed in this module are described below.

- Monitoring: During this procedure the feature extraction, selection or generation processes are applied over the flow trace.
- Analyze: This module is in charge to classify the flow processed using M. The main objective is of classifying the traffic category to forward this information to the QoS manager as it is denoted in Figure 4.

The traffic analyzer manager relies on how well the orchestrating system is defining the features and models to classify the traffic.

4 EXPERIMENTAL TEST

In this section, we present the procedure that should be performed by the orchestrating system to initialize the knowledge of the Self-optimizing manager over a case study. We will show the tests that must be done to perform a feature extraction or selection process, and to assure the construction of ML models. The result will be the best configuration $\{F, M\}$ obtained in a manual manner.

4.1 Historical dataset

The dataset used to the present study was developed by [3]. The dataset was collected using a modified version of the Volunteer-Based System (VBS). The authors collect all the packets passing the network interfaces, where the packets are grouped into flows, and the process name (taken from the system sockets) is assigned to each flow.

In real world scenarios in traffic network is very difficult to inspect end-to-end communications due to several aspects mainly concerning to privacy matters. Therefore, the authors in [3] created an emulated environment that allows them to acquire complete flows from several end-to-end communications. They used 4 hardware machines, 2 with Windows 7 and 2 with Ubuntu, plus 3 virtual machines with Windows 7, Windows XP, and Ubuntu as data generating stations. A server machine was used for data storage. VBS was used to collect the information about the flows such as start time of the flow, number of packets contained by the flow, local and remote IP addresses, local and remote ports, transport layer

protocol, along with detailed information about each packet. Three dataset were obtained from this process: i. PAM 1 dataset (8.95 GB), ii. PAM 2 dataset (21.14 GB) and ii. PAM 3 dataset (6.73 GB).

4.2 Monitoring

The historical dataset is processed by this module in order to extract valuable information from it. For the present job, the datasets were merged and cleaned getting as a result a new dataset PAM with 177135 flows. Additionally, the data set was processed in order to obtain the name of the application and the category of each flow.

Ten categories were identified within the flows, that is: Web protocols, File sharing (P2P), Social network, Streaming, network communication protocols, real-time communication (VOIP), System level applications, File transport protocol, VPNs and Protocols for database communication. As it was previously mentioned, the most important categories to detect is the streaming and the real-time communication ones for improving the QoS. These categories count with applications such as Netflix, Flash, YouTube, Skype, and among others.

4.3 Analyzer

The ML process is performed by this module as an initial stage as it was remarked previously. The results are detailed below.

4.3.1 Feature extraction. For this particular case, 26 features were implemented, these features are described in Table 1 for the flow directions a and b (server and client, respectively).

Table 1: Features extracted from the packets flows.

Feature		Type	Amount
Protocol		Categorical	1
Server/Client Port		Numeric	2
Duration		Time variable	1
Total flow Flow a Flow b	Amount of packets Total Bytes	Numeric	6
Packet size in flow a and flow b	Minimum Maximum Mean Standard Deviation	Bytes-Numeric	8
Inter-arrival time(IAT) between packets in a and b	Minimum Maximum Mean Standard Deviation	Numeric	8
Category or Application protocol		Categorical	1
		Total	27

4.3.2 Feature selection. We will present the resulting features obtained by each approach defined previously. It is important to mention that the server and client ports are discarded from the beginning of the study, because this information might not be available. Moreover, the ML model can incur in errors due to the dynamically setting of the ports.

- Correlation analysis: after applying the correlation over the 24 original features (where the label is not included in this test), 17 features remain as uncorrelated for this particular dataset. The features discarded were: total packets and bytes, server bytes, client and server mean packets, maximum and mean server IAT.
- Random forest: the resulting dataset form the previous step is fed to a forest with 500 trees. The RF model was trained, and the information degree of each feature was computed. After applying a ranked selection, the most important features were: protocol, client minimum and maximum packet length, server minimum and maximum packet length, client mean IAT and server minimum IAT.

4.3.3 ML results. We present as follows the accuracy of the classifiers trained with: i) the total of flows labeled with their category in Table 2, ii) the flows belonging to the streaming and real-time communication category with their best classifier and feature selection/reduction performed in Table 3, and iii) additionally, the total amount flows labeled with their category but each flow is truncated with 25% (Q1), 50% (Q2) and 75% (Q3) of the packets in Table 4.

In Table 2, the best results are standing in red while the best second one is in bold. It is noticeable that the Random Forest (RF) classifier gives the best results, while at the same time can select 6 significant features and the accuracy is not significantly degraded. The same behavior occurs for the rest of the tests. The truncation of the flows allows us to analyze how the model will behave when facing communications that are not finished.

Table 2: Accuracy of the category classifiers per feature reduction/ selection performed.

	Original 26 features	Correlation Analysis 85% 17 features	Random forest 6 features
RF	**0.994**	**0.994**	**0.957**
DT	0.950	0.923	0.912
SVM	0.653	0.644	0.612
KNN	0.765	0.759	0.821

Table 3: Accuracy of the best classifier and feature reduction/ selection performed to identify the name of application.

	Random forest 6 features
RF streaming	0.931
RF VOIP	0.917

Table 4: Accuracy of the best classifier per quartile using all the features.

	Q1	Q2	Q3	Original
RF	0.972	0.972	0.982	0.994

The results above showed the complete ML process executed using different ML approaches for performing feature selection and building classification models. The selection of the best model for this particular dataset was the RF in the three cases. These results have to be obtained by the *Orchestrating system* in an autonomic manner, and consequently set as a new knowledge to the *Self-optimizing manager*.

5 AUTONOMIC PROPOSAL ANALYSIS

In the previous section, we detail one of the procedures to be implemented in the *Orchestrating system*. In order to integrate all the results and implement them in an automatic manner, we used the platform Pentaho [1] for data integration. Pentaho allowed us to join all the components needed to deployed the experimental example. The aim is to develop and deliver an automatic process, the automated system cover from the features extraction from flow traces in .pcap files to the automated selection of the ML algorithm.

In this platform a job can be placed at a server with tasks such as scheduling training and tests of several ML models. Figure 5 shows an implementation of the *Orchestrating system* following the process in Section 3.1. Basically, this implementation counts with the jobs that define an autonomous manager such as monitoring, analyzer, planner and executor. Inside these jobs, a transformation is called to perform several operations over the historical dataset. In the figure is shown the transformations belonging to the Analyzer component for training and test different ML models using the historical dataset. In this step the reduced datasets detailed in Section 4.3.2 are obtained with the ML models. *training1*, *training2* and *training3* are nested transformations that are in charge of building the four models defined in Section 4.3.3. Following, the planner takes the test results and define the best classifier to set as knowledge.

An online implementation of the *Traffic analyzer manager* is presented in Figure 6. Firstable, it is assumed that the packet flows are already processed, and the features per flows are coming in a streaming way. An analysis is performed with the ML learning model given by the *Orchestrating system*. Depending on the best model selected and in which software was implemented (R, Python, Matlab, etc) the prediction is performed. The label predicted is used by the QoS manager for planning and executing reconfigurations in the network, if needed.

6 RELATED WORKS

A framework is presented in [1] to large-scale network monitoring and analysis called DBStream. DBStream is a data repository of network monitoring data capable of processing data streams coming from a wide variety of sources. One important component of DBStream is an online classification of traffic using ML techniques. Different classifiers can be placed into the framework for the online classification. [8] presents a Self-Learning Traffic Classifier (SLTC) for P2P identification. The ML classifier is build with statistical approach, and the authors proposed an implementation of the solution based on rules for different scenarios. On the other hand, [11] proposed a system that uses ML models to predict the traffic and improve the QoS. The system programs trainings of a classifier, that is used to identify live data at a monitoring point.

The previous solutions do not include autonomous or adaptive processes to improve the performance of the traffic analyzer. Even though, the classifiers are constantly retrained, it might occur that a ML model is better than another given different scenarios; the same case appears with the features extracted. In consequence, provide an static solution might not be sufficient given the dynamism of the network traffic. Our proposal aim at giving flexibility and adaptability for traffic network analysis.

[1] http://www.pentaho.com/

7 CONCLUSIONS

Traffic analysis to improve the QoS was studied using ML to classify flows, the proposal tries to offer an effective solution for non-experts in the ML and autonomic computing field. We propose autonomic system *Self-optimizing manager* that integrates all the components needed to improve the QoS in an autonomous manner. In particular, we focus our attention of how the knowledge is learned, and how the traffic analysis is performed using this knowledge.

We present the complete procedure that has to be performed to classify traffic, and we show that the combination feature-model can affect the accuracy of the classification results. The results demonstrated the capability of ML models to classify flows, where RF classifier obtained the best accuracy. Moreover, RF based feature selection was used to obtain only six features that can help to predict the traffic without losing performance.

We noticed that the complete ML process is tedious and requires several tests to assure its adequate usage; however, autonomic computing offers a solution to facilitate these procedures to other entities. Therefore, defining the best features-model was the main goal of the *Orchestrating system* in this context. We give an example of how our approach can be implemented, particularity the *Orchestrating system* and the *Traffic analyzer* manager.

As future works, we propose to detail the autonomic architecture with a more fine selection of the pair features-model, that will integrate metrics from the *Traffic analyzer* and *QoS* manager. Additionally, more implementation matters will be treated such as on-line tests, complexity, resource management, time response, etc.

8 ACKNOWLEDGMENTS

This work is sponsored by Thales Alenias Space, TOULOUSE, France.

REFERENCES

[1] Arian Baer, Pedro Casas, Alessandro D'Alconzo, Pierdomenico Fiadino, Lukasz Golab, Marco Mellia, and Erich Schikuta. 2016. DBStream: A holistic approach to large-scale network traffic monitoring and analysis. *Computer Networks* 107 (2016), 5 – 19.

[2] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. 2016. On Internet Traffic Classification: A Two-Phased Machine Learning Approach. *Journal of Computer Networks and Communications* (2016).

[3] Tomasz Bujlow, ValentÃŋn Carela-Español, and Pere Barlet-Ros. 2015. Independent comparison of popular DPI tools for traffic classification. *Computer Networks* 76 (2015), 75 – 89.

[4] Dinil Mon Divakaran, Le Su, Yung Siang Liau, and Vrizlynn L. L. Thing. 2015. SLIC: Self-Learning Intelligent Classifier for network traffic. *Computer Networks* 91 (2015), 283 – 297.

[5] IBM. 2005. *An Architectural Blueprint for Autonomic Computing.* Technical Report. IBM.

[6] Félix Iglesias and Tanja Zseby. 2016. Time-activity footprints in IP traffic. *Computer Networks* 107 (2016), 64 – 75.

[7] IUT. November 2001. *SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS: Quality of service and performance.* Technical Report. International Telecommunication Union.

[8] Ram Keralapura, Antonio Nucci, and Chen-Nee Chuah. 2010. A novel self-learning architecture for p2p traffic classification in high speed networks. *Computer Networks* 54, 7 (2010), 1055 – 1068.

[9] Guan-Zhou Lin, Yang Xin, Xin xin Niu, and Hui bai Jiang. 2010. Network traffic classification based on semi-supervised clustering. *The Journal of China Universities of Posts and Telecommunications* 17 (2010), 84 – 88.

[10] Zhen Liu and Qiong Liu. 2012. Studying cost-sensitive learning for multi-class imbalance in Internet traffic classification. *The Journal of China Universities of Posts and Telecommunications* 19, 6 (2012), 63 – 72.

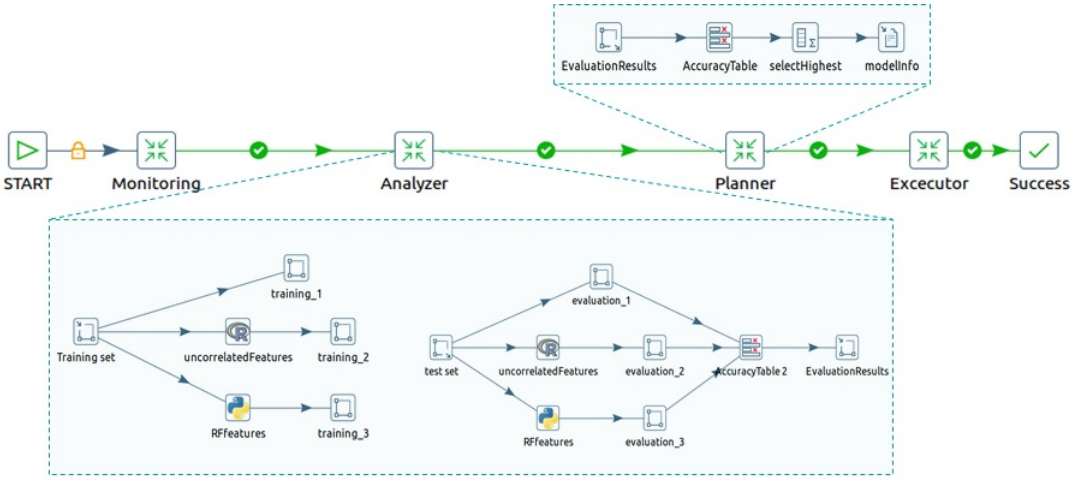

Figure 5: Orchestrating system implementation.

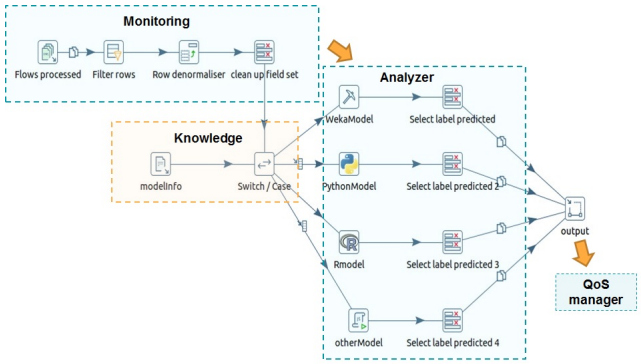

Figure 6: Deployment of the Traffic analyzer manager in an online manner.

[11] Stuart E. Middleton and Stefano Modafferi. 2016. Scalable classification of QoS for real-time interactive applications from IP traffic measurements. *Computer Networks* 107 (2016), 121 – 132.

[12] Andrew Moore, Michael Crogan, Andrew W Moore, Queen Mary, Denis Zuev, Denis Zuev, and Michael L Crogan. 2005. *Discriminators for use in flow-based classification.* Technical Report. University of London.

[13] T. T. T. Nguyen and G. Armitage. 2008. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys Tutorials* 10, 4 (2008), 56–76.

[14] Manish Parashar and Salim Hariri. 2005. *Autonomic Computing: An Overview.* 257–269.

[15] Y Snir, Y Ramberg, J Strassner, R Cohen, and B Moore. November 2003. *Policy Quality of Service (QoS) Information Model.* Technical Report. RFC 3644.

[16] Murat Soysal and Ece Guran Schmidt. 2010. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation* 67, 6 (2010), 451 – 467.

[17] D Tiwari and B Mallick. 2016. SVM and NaÃŕve Bayes Network Traffic Classification using Correlation Information. *International Journal of Computer Applications* 147 (2016), 1–5.

[18] Hastie Trevor, Tibshirani Robert, and Friedman Jerome. 2009. *The elements of statistical learning: data mining, inference, and prediction.* Springer, New York.

[19] Petr Velan, Milan Cermák, Pavel Celeda, and Martin Drasar. 2015. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management* 25, 5 (2015), 355–374.

[20] P. Wang, S. C. Lin, and M. Luo. 2016. A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs. In *2016 IEEE International Conference on Services Computing (SCC).* 760–765.

[21] Jun Zhang, Yang Xiang, Wanlei Zhou, and Yu Wang. 2013. Unsupervised traffic classification using flow statistical properties and IP packet payload. *J. Comput. System Sci.* 79, 5 (2013), 573 – 585.