Personalized Service Degradation on OTT Applications

Juan Sebastián Rojas Meléndez

Tesis de Maestría en Ingeniería Telemática

Director:
Juan Carlos Corrales Muñoz
PhD. En Ciencias de la Computación

Universidad Del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, Abril de 2018

Juan Sebastián Rojas Meléndez

Personalized Service Degradation on OTT Applications

Tesis presentada en la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de

Magister en:
Ingeniería Telemática

Director:
Juan Carlos Corrales Muñoz
PhD. en Ciencias de la Computación

Popayán
2018

Dedicated to my family and all the people that,
with their support,
made the culmination of this work a reality.

Acknowledgements

The author would like to thank University of Cauca and PhD. Juan Carlos Corrales Muñoz for his guidance and advisement in the development of this project, Colciencias for the support given through the PhD scholarship and Mag. Eduardo Rojas Pineda for all his support and collaboration.

Structured abstract

OTT applications are known by their large consumption of network resources for their correct operation. In the mobile networks scope, where operators offer users data plans with limited consumption, service degradation is a measure implemented in a generalized way in order to apply limits to the amount of information that can be transferred by the users over a period of time when a user exceeds his/her established consumption limit, in order to save resources and ensure the correct performance of the network. Therefore, the behavior and preferences presented by the user in the consumption of OTT applications are never considered when this mechanism is implemented.

Considering the previous background, this research project aims personalizing the service degradation policies in mobile Internet based on the OTT applications consumption of the users.

In order to accomplish this objective an identification of the QoS parameters related to the service degradation and the parameters that describe the user consumption behavior will be identified. A selection of classification techniques aimed at characterizing the users OTT consumption behavior will be performed and a set of personalized service degradation policies considering the user behavior will be proposed.

Some of the obtained results in the development of this research project are: a group of datasets that enable the classification of the users based on their consumption behavior; a set of parameters that must be considered in order to propose QoS policies

and to characterize the user consumption behavior; the implementation of a classifier that enables the identification of the consumption trends of OTT applications by users and the proposal of personalized service degradation policies considering the consumption trends of OTT applications of users.

From the obtained results it can be concluded that a set of personalized service degradation policies having in mind the OTT consumption behavior presented by the users were proposed for each of the three groups (high, medium and low consumption) defined after the clustering process, however it is important to mention that the implementation of a scheme that enable this process inside a real mobile network is still to be developed.

As future works it is proposed to develop a framework that facilitates the gathering of data related with the consumption of OTT applications done by users of a mobile network; Create a dataset with a larger number of users and larger quantity of information of different OTT applications (e.g., a month) and develop and integrate a knowledge plane that enables the implementation of Artificial Intelligence techniques inside an Internet network.

Keywords: OTT Applications, Service Degradation, Machine Learning, Classification, Dataset, DPI, QoS Policy, PCC architecture.

Resumen estructurado

Las aplicaciones OTT son conocidas por su gran consumo de recursos de red para su correcto funcionamiento. En el ámbito de las redes móviles, donde los operadores ofrecen a los usuarios planes de datos con consumo limitado, la degradación del servicio es una medida implementada de forma generalizada para aplicar límites a la cantidad de información que los usuarios pueden transferir durante un período de tiempo, cuando el usuario excede su límite de consumo establecido, a fin de ahorrar recursos y garantizar el correcto funcionamiento de la red. Por lo tanto, el comportamiento y las preferencias presentadas por el usuario en el consumo de aplicaciones OTT nunca se tienen en cuenta cuando se implementa este mecanismo.

Teniendo en cuenta lo mencionado anteriormente, este proyecto de investigación tiene como objetivo personalizar las políticas de degradación del servicio en Internet móvil en función del comportamiento de consumo de los usuarios respecto a las aplicaciones OTT.

Para lograr este objetivo, se plantea realizar una identificación de los parámetros de QoS relacionados con la degradación del servicio y los parámetros que describen el comportamiento de consumo de los usuarios. Además se realizará una selección de técnicas de clasificación para caracterizar el comportamiento de consumo de aplicaciones OTT de los usuarios y se propondrá un conjunto de políticas personalizadas de degradación del servicio teniendo en cuenta el comportamiento de los usuarios.

Algunos de los resultados obtenidos en el desarrollo de este proyecto de investigación son: un grupo de conjuntos de datos que permiten la clasificación de los usuarios a partir de su comportamiento de consumo; un conjunto de parámetros que deben considerarse para proponer políticas de QoS y caracterizar el comportamiento de consumo del usuario; la implementación de un clasificador que permite la identificación de las tendencias de consumo de las aplicaciones OTT por parte de los usuarios y la propuesta de políticas personalizadas de degradación de servicio teniendo en cuenta las tendencias de consumo de las aplicaciones OTT por parte de los usuarios.

De los resultados obtenidos se puede concluir que se propuso un conjunto de políticas personalizadas de degradación de servicio teniendo en cuenta el comportamiento de consumo de aplicaciones OTT presentado por los usuarios para cada uno de los tres grupos (alto, medio y bajo consumo) que fueron definidos después de un proceso de agrupamiento. Sin embargo es importante mencionar que aún no se ha desarrollado la implementación de un esquema que permita este proceso dentro de una red móvil real.

Como trabajos futuros, se propone desarrollar un framework que facilite la recopilación de datos relacionados con el consumo de aplicaciones OTT realizado por los usuarios de una red móvil; Crear un conjunto de datos con un mayor número de usuarios y una mayor cantidad de información de diferentes aplicaciones OTT (por ejemplo, un mes) y desarrollar e integrar un plano de conocimiento que permita la implementación de técnicas de Inteligencia Artificial dentro de una red de Internet.

Palabras Clave: Aplicaciones OTT, Degradación de servicio, Aprendizaje automático, Clasificación, Conjunto de datos, DPI, Políticas de QoS, Arquitectura PCC.

Content

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Context

Currently the Information and Communications Technologies (ICT) market is undergoing through extremely rapid changes. The business models traditionally used by ISP (Internet Service Providers) where each company had their own network infrastructure and offered a unique set of services is facing a new challenge the OTT (Over The Top) services.

OTT services is the expression used for services carried over the networks, delivering added value to customers, but without any ISP being involved in planning, selling, provisioning, or servicing them and of course without any traditional telecommunications booking revenue obtained from them. The current generation of service and application companies that use an OTT business model, as a platform for their new products, has begun to generate major hardships in the traditional business model used by ISP. Companies and applications such as Skype, YouTube, Facebook, Netflix, among many others, have emerged to tackle the new needs in communications and functionalities that customers demand [1], [2].

Due to this change the ISPs have found themselves in a scenario that represents great difficulties, where they are no longer the sole competitors in the market and through the scheme proposed by OTT services have become an intermediary that only carries information between OTT applications and the different users who have hired their Internet connection services. For this reason, their traditional business model where the user hired access to an internet connection and different applications deployed through their infrastructure is being remodeled for a more flexible one that considers OTT service providers as allies. This way, ISPs can generate revenue through from the high consumption users of this type of applications and on the other hand, OTT service providers obtain benefits by complying with a Service Level Agreement (SLA) that guarantees the correct operation of their applications.

1.2. Motivation

Nowadays considering the revolution that is being provoked by the OTT services on the ISP traditional business models, cooperation agreements have been established aiming at increasing the benefits and economic revenue obtained from the high consumption rates that costumers present on OTT services. Such cooperation involve the establishment of a Service Level agreement (SLA)

A SLA is defined as an official commitment that involves particular aspects of the service: quality, availability and responsibilities. The most common component of SLA is that the services should be provided to the customer as agreed upon in the contract. Internet service providers and OTT companies will commonly include service level agreements within the terms of their contracts with customers to define the levels of service being sold in QoS (Quality of Service) terms (throughput, delay times, jitter or similar measurable details) [3]. Hence the OTT companies benefit since ISP guarantees a good quality in the provisioning of their service and the ISP benefits from the number of users that access that specific OTT service.

However, even though such changes in the business model are being considered OTT applications are known by their large consumption of network resources for their correct operation, hence a set of resource control mechanisms must be implemented aiming

at maintaining a good performance of the network. For this reason data caps and service degradation are usually implemented in order to apply limits to the amount of information that can be transferred by network users over a period of time [4], [5]. Such mechanisms are most commonly used in mobile networks where the data plans offered by traditional operators establish consumption limits. Nevertheless, when applied, service degradation majorly affects the performance of the OTT services having a negative impact in the perception obtained by the users of the application and besides it fails to comply with the established SLA defined between the ISP and the OTT companies and the ISP and the user that hired the Internet services. Therefore an alternative that enables a good network management for network operators and impacts the least in the user perception of OTT applications must be considered.

1.3. Problem Definition

The ever-increasing adoption of smartphones and the continuous development of mobile networks have resulted into consumers having access to multiple types of applications changing the usage and traffic patterns beyond the traditional voice and messaging. This global trend indicates that with the emergence of internet-based service providers or rather known as Over-the-top (OTT), the business landscape has changed massively [6].

As mentioned before, OTT applications are known by their large consumption of network resources needed for their correct operation and in the mobile networks scope, where operators offer users data plans with limited consumption, data caps and service degradation are resource control mechanisms implemented in order to apply limits to the amount of information that can be transferred by network users over a period of time. It is usually applied following a set of policies defined by the network operator and are implemented when a user exceeds his established consumption limit, in order to save resources and ensure the correct performance of the network. It can be either a degradation in the performance of the accessed applications or a cancelation in the service provisioning by the network operator.

However, this degradation is applied in a generalized way, i.e., once this measure is applied, the performance of all the applications that the user can employ is affected. Therefore the user's behavior and preferences regarding the consumption of OTT applications are never considered. Furthermore it breaches the SLA that the ISP has been able to establish with certain OTT applications. With this in mind and considering the scenario previously described this research project aims at answering the following research question:

How to minimize the impact on OTT applications consumption commonly employed by the user in a mobile internet operator QoS degradation scheme?

1.4. Objectives

Considering the previous motivation and the research question, this research project aims personalizing the service degradation policies in mobile Internet based on the OTT applications consumption of the users.

In order to accomplish this objective an identification of the QoS parameters related to the service degradation and the parameters that describe the user consumption behavior will be identified. A selection of classification techniques aimed at characterizing the users OTT consumption behavior will be performed and a set of personalized service degradation policies considering the user behavior will be proposed.

1.5. Contributions

In the present research project the following contributions can be stated:

A group of datasets that enable the classification of the users based on their consumption behavior of OTT applications.

A set of parameters that must be considered in order to propose QoS policies and to characterize the user consumption behavior based on a traditional network monitoring process.

A comparative study of supervised learning algorithms applied to the datasets related to the OTT application domain under a QoS degradation scheme.

The implementation of a classifier that enables the identification of the consumption trends of OTT applications by users and subsequently facilitates the generation of personalized service degradation policies.

A proposal of personalized service degradation policies considering the consumption trends of OTT applications of users following the technical specifications of a LTE network.

Publishing papers: Within the present research project the paper titled: "Personalized Service Degradation Policies on OTT Applications based on the Consumption Behavior of Users" was accepted to be published on the Mobile Communications Workshop 2018 (MC 2018) which is developed within the International Conference on Computational Science and its Applications (ICCSA 2018) in Melbourne, Australia from July 2nd to July 5th. The paper will be included in the Springer Lecture Notes in Computer Science (LNCS) series.

1.6. Content

The structure of the present document will be described as follows:

Chapter 2: This chapter presents a description of the most relevant concepts within this research project, including: data caps or service degradation, Quality of Service (QoS) and traffic classification focusing on statistical classification and Deep Packet Inspection (DPI); subsequently this chapter presents a set of related works about the service degradation; Quality of Service (QoS); OTT services; Traffic classification and Categorization of users in a mobile network.

Chapter 3: This chapter presents a detailed description of the generated datasets focusing on the attributes, tools and preprocessing procedures that were performed in order to obtain the classification models.

Chapter 4: This chapter presents a detailed description of the modeling process illustrating the supervised learning algorithms that were used and the obtained results on each test.

Chapter 5: This chapter presents the structure and proposed personalized service degradation policies based on the consumption behavior presented by the users stored on the datasets, following the Policy and Charging Control (PCC) architecture proposed for a LTE network.

Chapter 6: This chapter presents the conclusions obtained from the development of this research project along with some possible future works.

Chapter 2

State of the art

This chapter introduces a description of the most relevant concepts within this research project, the explained concepts are: data caps or service degradation, Quality of Service (QoS) and traffic classification focusing on statistical classification and Deep Packet Inspection (DPI); subsequently this chapter presents a set of related works about the service degradation with the objective of identifying how this resource control mechanism is managed in the networks by Internet service providers (ISP); Quality of Service (QoS) focusing on identifying the parameters closely related to the service degradation; OTT services in order to know how this topic has been worked in the research field; Traffic classification focusing on identifying which techniques are used for this process; Categorization of users in a mobile network in order to know how operators manage users inside the network.

2.1. General context

Before describing the proposed solution in this research project it is necessary to define and comprehend some of the most important concepts, surrounding service

degradation, Quality of Service (QoS) and traffic classification which will be briefly described as follows.

2.1.1. Service degradation – Data Cap Models

Faced with increased network congestion from both the rise in bandwidth intensive applications and the growing number of Internet users, many Internet Service Providers (ISP) have imposed a data cap or monthly data limit on their subscribers [7]. These bandwidth caps vary from 1-250 GB and exist in nations such as Australia, Canada, Turkey, South Africa, the U.K., and the United States [8]. With the transition from the flat rate dominated pricing regime towards volume-based tariff, data caps are not restricted to home broadband; they are also part of the pricing model applied to mobile Internet users [9]. Since ISP argue that caps help provide more consistent service to all their users, this pricing model is likely to persist [4].

Currently there are three dominant volume-based tariff schemes at the market. The fair-flat tariff establish volume-thresholds that are used to increase prices for heavy users. Customers that consume below the volume-threshold pay the standard flat rate price, whereas customers that exceed the fair-use level pay a predefined premium in that billing-period. Often providers notify users about their actual consumption and warn them if they are about to exceed the fair-use level. Some providers are even charging the additional fee only after repeated overuse (e.g. two months in a row). However, overall consumption under fair-flat tariffs is not limited.

The second dominant volume-based tariff is known as three-part tariff. A three-part tariff is defined by an access price, an allowance, and a marginal price for any usage in excess of the allowance. Consumers with a three-part tariff pay for any usage in excess of their allowance and can end up with relatively high cost for their additional data consumption. That unexpected high cost makes this tariff unattractive from a customer perspective because it adds a pay-per-use element to the already uncertain and unpredictable demand for data consumption.

Data caps on the other hand are the dominant volume-based tariff scheme in mobile Internet access, but are more and more common in fixed-line Internet access as well. Tariffs with data caps are very often sold under the flat rate label. However, in contrast

to flat rate tariffs, consumption under data caps is strictly limited and overuse requires direct customer action. The enforcement of data caps can either have the form of immediate disruption of the Internet service, or a service quality degradation of the connection. For example, many mobile network operators reduce the bandwidth of the connection to a speed equivalent to the Integrated Services Digital Network (ISDN) when the cap is reached. That form of “soft enforcement” allows operators to make the claim of unlimited Internet usage in their marketing campaigns, without losing the important aspect of volume-based price discrimination. When the cap is reached, customers often have the option to pay an additional fee to continue to be able to use the Internet or to restore the full speed of the connection. Providers either charge customers to reset their original quota-limit, or to buy an additional predefined data-volume [10]. Figure 1 illustrates the different volume based tariff schemes.

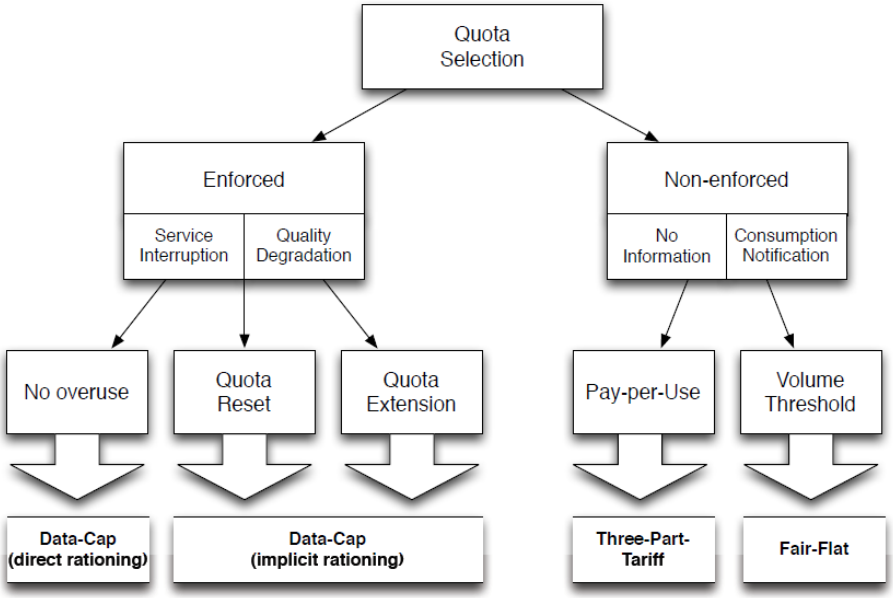

Figure 2.1. Volume based tariff schemes [10].

2.1.2. Quality of Service - QoS

Quality of Service is a topic that has a long background and investigation. Therefore there are various definitions that have been proposed throughout the years. For instance, in its technical report [11], ETSI defines QoS from the network perspective as: “the ability to segment traffic or differentiate between traffic types in order for the

network to treat certain traffic differently from others", and in the ISO definition [12], quality is defined as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs".

However, currently the most recent and used definition is the one given by the ITU in its Quality of Service Regulation Manual [13] where it is defined as: "the totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service". The ITU definition is consistent with the ISO definition. Compared to the ETSI definition from a network perspective, the ITU and ISO definitions focus on the service as the entity under consideration. It is important to notice however, that the various definitions tend to reflect views on the telecommunication/ICT systems, networks, and services from user and network perspectives.

Traditionally, QoS was mainly addressed from the perspective of the end-user being a person with abilities to hear and see and be tolerant to some degradation of services (e.g. low packet loss ratio is acceptable for voice, while end-to-end delay for voice should be less than 400 milliseconds). But with the advent of new types of communications where services may not require real time delivery and where the sender or the end-user may not be a person but a machine, it is important to keep in mind that not all services are the same (e.g. Internet of Things - IoT). Even similar services can be treated in different ways depending on whether they are used by machines or by humans on one or both ends of a given communication session or connection. The end-user perception of a telecommunication/ICT service is also influenced by different factors such as social trends (in terms of popular devices, services, applications, social networks, etc.), advertising, tariffs and costs, which are interrelated to the customer expectation of QoS. The user perception of quality is not limited to objective characteristics at the man-machine interface. For end-users, the quality that they personally experience during their use of a telecommunication service also counts [13].

As illustrated in Figure 2.2, modern QoS not only depends on end-to-end technical aspects, which include network performance and terminal performance, but also on non-technical aspects (not directly related to the equipment), such as point of sale, customer care.

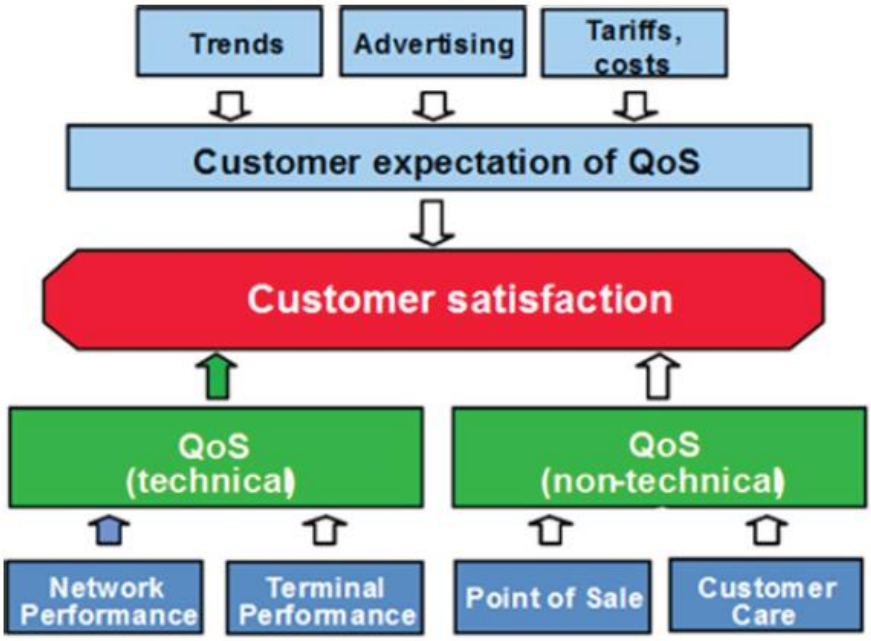

Figure 2.2. QoS technical and non-technical point of view [13].

2.1.3. Traffic Classification

Internet traffic classification has been the subject of intensive study since the birth of the Internet itself. Indeed, the evolution of approaches for traffic classification can be associated with the evolution of the Internet itself and with the adoption of new services and the emergence of novel applications and communication paradigms. Throughout the years many approaches have been proposed for addressing technical issues imposed by such novel services [14]. Among them the following can be found: Port number matching, Deep Packet Inspection (DPI), Classification in the Dark and Active Crawlers which will be briefly described as follows:

Traffic Classification based on port numbers

The classification of network traffic based on the User Datagram Protocol (UDP) or TCP port numbers is a simple approach built upon the assumption that each application protocol always uses the same specific transport-layer port. This method was mostly useful in the identification of well-known protocols like, for example, HTTP or Simple Mail Transfer Protocol (SMTP), which use the port numbers 80 and 25, respectively.

However, many Internet applications easily bypass this identification strategy by simply using random or unknown port numbers, thereby disguising their traffic using port numbers generally used by other well-known protocols (e.g., port 80) that are usually allowed by firewalls. Thereby, nowadays port numbers as a classification mechanism is considered obsolete [15].

Traffic Classification based on Deep Packet Inspection

DPI methods, usually the most accurate, are based on inspection of the packets' payload. They rely on a database of previously known signatures that are associated to application protocols, and search each packet for strings that match any of the signatures. This approach is used not only in the classification of network traffic, but also in the identification of threats, malicious data, and other anomalies. Because of their effectiveness, classification systems based on DPI are especially significant for accounting solutions, charging mechanisms, or other purposes for which the accuracy is crucial. However, the main drawback of DPI techniques is their inability to be used when the traffic is encrypted (communications with TLS or SSL). Since, in these cases, the contents of the packets are inaccessible (encrypted), DPI-based mechanisms are restricted to specific packets of the connection (e.g., when the session is established) or to the cases when UDP and TCP connections are used concurrently and only the TCP sessions are encrypted. Packets with no payload, which may be malicious, cannot be classified as well. DPI methods are also sensitive to modifications in the protocol or to evolution of the application version: any changes in the signatures known by the classifier will most certainly prevent it from identifying the application. Moreover, DPI methods that rely on signatures for specific applications can only identify traffic generated by those applications [15].

Traffic Classification in the Dark

The inspection of the contents of IP packets, is not always a valid option for the identification of application-level protocols. Therefore, new methods that do not resort to the deep inspection of the packets have been developed. The strategy of this kind of approach, sometimes called in the dark, is to classify the traffic using behavioral or statistical patterns based on flow-level data or generic properties of the packets, like addresses, ports, packet size among others. The main advantage of classification in

the dark is the ability to identify a protocol without the need to examine the contents of the packet. As a consequence, mechanisms based on this approach cannot aspire to the same accuracy level of DPI methods. Their results should be understood as a strong suspicion regarding the probable application protocol. Additionally, classification in the dark can more easily be applied to unknown applications since many methods based on this approach classify the traffic in classes of applications (e.g., Web traffic, email, video streaming, P2P, etc.) instead of specific applications. The existent mechanisms use distinct techniques to correlate the traffic properties and conclude on the application protocol, such as statistical measures, sets of heuristics, or machine learning algorithms [15]. Table 2.1 illustrates a brief comparison of the different traffic classification approaches.

Approaches	Characteristics	Advantages	Weaknesses
Port number matching	- Associates port numbers with applications	- Low Computational requirements - Easy to implement	- Lack of classification performance due to random port numbers (obsolete)
Deep Packet Inspection	- Relies on payload data	- High classification performance	- May not work for encrypted traffic - Requires high processing resources - Can only be used for known applications
Classification in the dark	- Uses only packet header and flow level information	- Usually lighter than DPI - Applicable for encrypted traffic - Can identify unknown applications from target classes	- Usually has lower classification performance when compared to DPI

Table 2.1 Comparison of the traffic classification approaches [15].

2.2. Related works

This section presents an analysis of the most important related works within the framework of this proposal, highlighting that until now there have not been projects that consider a similar or identical approach to the one presented in this research. Therefore, the current state of knowledge will be presented after implementing a

systematic mapping of academic documents, based on the methodology proposed in [16], to provide an overview of the research area and determine the amount and type of works.

The selected topics of interest were: The service degradation with the objective of identifying how this resource control mechanism is managed in the networks by Internet service providers (ISP); Quality of Service (QoS) focusing on identifying the parameters closely related to the service degradation; OTT services in order to know how this topic has been worked in the research field highlighting the fact that it is a very recent investigation area; Traffic classification focusing on identifying which techniques are used for this process; Categorization of users in a mobile network in order to know how operators manage users inside the network. To perform the search, different databases of scientific research (Scopus, IEEE explore, ACM) were consulted, where Scopus was finally selected as the main source of information since it presented the best results related to the implemented search chains, providing a total of 1030 papers related to research topics.

Finally, a review is made discarding the papers that were not useful and separating the rest following the mentioned thematic cores. The remaining papers were distributed like this: 22 papers for service degradation, 28 papers for Quality of Service (QoS), 33 papers for traffic classification, 47 papers for OTT services and 4 papers for categorization of users. These works are separated considering the different approaches found within each topic in question. The following subsections will briefly describe the most representative works around each thematic core.

2.2.1. Service Degradation related works

As mentioned before, each thematic core has different approaches considering the type and objective of the papers that were found. In the case of service degradation two approaches were defined: business models, highlighting works focusing on how ISP business models and users experience is affected by the implementation of data caps. As a second approach a resource control techniques group was defined highlighting works that are mainly focused on how to avoid the service degradation using different mechanisms such as "video prefetching", detection of alternative networks and packet compression in HTTP browsing.

Starting with the papers related with business models in [4] Chetty et.al., describe how monthly bandwidth caps affect households' Internet use in South Africa, a country that prior to February 2010 had all home broadband data subscriptions capped so most users' experience of broadband was of a metered connection. This paper aimed at learning how bandwidth caps affect households' broadband, how households manage a bandwidth cap during the month and what tools and information households desire to monitor and control their bandwidth usage. To do this, a qualitative study was conducted on 12 households living with data caps.

In [17], a work developed by the US company Ixia, an analysis of QoS policies that are generally implemented in an LTE mobile network (dynamic allocation of network resources, priority control, limitation of traffic rates) to manage network congestion, improve the QoS and qualify the services is performed. This analysis highlights which are the QoS parameters that significantly affect the performance of the different services offered on the network (video, voice, games, internet, etc.). It also highlights the importance of an operator making a categorization of the users in such a way that the network resources are managed efficiently.

In [10] considering the on-going transition of the Internet to a universal communications access technology, data pricing becomes the main driver of revenues for infrastructure providers in the future. With this in mind, Krämer et.al., outlines a first approach to understand and systematically analyze current and future business models based on data caps, their impact on customer behavior and on the service provider market. In [18] Joe-Wong et.al., presented the obtained results from the first TDP (Time-Dependent Pricing) trial with a commercial ISP. Time-dependent pricing (TDP) allows the ISP to effectively target network peaks by offering higher prices at those times, incentivizing users to consume data at other times. On this trial 27 customers of a local U.S. ISP were recruited and divided into users into time-independent pricing (TIP) and TDP groups. This trial presents important conclusions around the impact of data usage monitoring apps on cellular and Wi-Fi usage behavior and real costumers' price sensitivity and delay tolerance for different applications.

Now the most relevant papers related to the second defined approach (resource control techniques) include the work proposed by Agababov et.al., [19]. This work presents

Flywheel, a tool developed for Google, with the aim of being a proxy service for HTTP, which has the objective of extending the life time of users' data plans in mobile networks by reducing the size of the packages that are exchanged between servers and user equipment. Flywheel, integrates with Chrome browser and, on average, reduces consumption rates by 50% generated by browsing and loading of web pages. Another paper presented by Chetty et.al., [5] present the design and implementation of a tool called uCap, a data cap management system that was deployed on 21 home networks in three countries (South Africa, India, and the United States) to help home users manage Internet data. Furthermore a qualitative study is applied on ten of the homes to evaluate which aspects of the tool users found most effective.

2.2.2. Quality of Service related works

Now this subsection will focus on describing the QoS related works. For this topic the subsequent categories were defined taking into consideration the type of QoS parameters that each paper used: Performance, Security, Configuration and Data following the classification proposed in [20]. However it is important to mention that at the end only the papers related to the performance profile were used, since this is where the parameters directly related to the service degradation are considered.

Considering that QoS can be a critical element for achieving the business goals of a service provider and for the acceptance of a service by the user, in [20] Kritikos et.al., performs a comparison between the different approaches of QoS description found in the literature where a large spectrum of models and meta-models to describe the service quality, ranging from ontological approaches to define quality measures, metrics, and dimensions are considered. This comparison enables the specification of quality-based service requirements and capabilities as well as of SLA (Service-Level Agreements) and SLA templates for service provisioning. Among the most interesting analysis presented on this paper that has a relation to this research project are the proposed QoS parameters that have a direct relationship with the performance of a service (Response time, Processing time, Latency, Timeliness, Precision and Throughput).

Nowadays smartphones are equipped with at least two access technologies, e.g. Wi-Fi and Cellular (3G or 4G LTE). Therefore, it is possible to use all available technologies

for Internet access at the same time. With this in mind Wamser et.al., [21] evaluated application-aware algorithms which can aggregate multiple access links to provide additional resources if necessary. Specifically the contributions presented in this paper are: the definition of three algorithms in the OTT Virtual Access Network (VAN) architecture which take application type or application quality into account. The aim is to address the application demands to achieve an effective network. Furthermore, an investigation of a user watching a YouTube video clip, while a Wi-Fi and a Cellular network are available is performed in order to evaluate the user perceived quality, cellular usage and device power consumption.

Considering that in In LTE networks, QoS is implemented between the User Equipment and the Packet-Data-Network-Gateway (P-GW), relying on the bearer concept, a virtual resource that maps a certain QoS class (e.g., guaranteed bitrate for video streaming) to particular resource reservations and that usually all OTT traffic on a LTE network is transferred without any specific QoS guarantees, Samdanis et.al., [22] proposes the Service Boost concept which is intended to address two problems: First provide a way for users to request the preferred network services (QoS) for selected applications and second enable operators to monetize these preferred services by charging users and/or Application Content Providers. The implementation of such concept could provide the means for mobile operators to offer users a mechanism to request a better service for selected flows and secondly enable the network to letting the user decide about important applications, so that network operators do not have to implement traffic detection functions like Deep Packet Inspection (DPI).

2.2.3. OTT Services related works

Proceeding with the related works, this subsection presents some of the most relevant papers related with the OTT Services. Considering the objectives of each paper, the next four categories were defined: Business models, holding papers that focused on describing and analyzing how OTT Services have begun to change the information and communication technologies market; Media, containing papers that centered their efforts on video OTT Services, highlighting the fact that these are the most researched services; VoIP, holding the papers that focused on research related with voice communication over Internet; and Messaging, holding the papers that focused on developments related to instant messaging services.

The ICT (Information and Communications Technologies) market is undergoing rapid and dramatic, changes, for this reason in [1] Wesley Clover, an investment management firm with active interests in ICT performs a well-structured analysis of the revolution provoked by OTT Services, highlighting what OTT services are, why they are so important, the ever increasing growth of mobile technologies and which decisions have to be made by Telcos and traditional service providers to avoid becoming a pipeline between OTT services and users. Moving on, aiming to integrate the OTT Services as a fair new competitor into the ICT market some countries have made efforts on implementing regulatory policies in order to control the rise of this kind of services. With this in mind, in [23] Barclay presents the regulatory responses enforced in some countries of the Caribbean and propose a regulatory framework that may aid in the effective management of OTT services and its evolution in the region. The framework considers the perspectives of the multiple stakeholders including regulatory agencies, telecommunications enterprises and customers.

Now, proceeding with the papers centered on video OTT Services and taking into consideration that Netflix and Hulu are the leading Over-the-Top (OTT) content service providers in the United States and Canada, Adhikari et.al., [24] performed an extensive measurement study to uncover their architectures and service strategies aiming at helping in the design and implementation of future systems. To accomplish such objective the authors dissect the basic architecture of the two popular video streaming platforms by monitoring the communications between the client-side player and various components of the two platforms. Furthermore this paper explores alternative strategies for improving video delivery performance using multiple CDN (Content Delivery Networks) while conforming to the business constraints.

In the Voice Over IP (VoIP) scope Zhu et.al., [25] study how QoE (Quality of Experience) can be enhanced for OTT applications over mobile broadband networks, considering only the last-mile radio access network and focusing on UDP based delay-sensitive real-time video call applications such as Skype. Their approach aims at taking advantage of the fact that applications are running on the end-user device over the top of the radio, and allow direct information exchange between applications and radio infrastructure. In [26] Wang et.al., applies Quality Function Deployment (QFD) to explore the customer requirements and identify prospective technologies of VoLTE (Voice over LTE) services. As an interesting conclusion the study illustrates that VoLTE

outperforms Over-the-top (OTT) services in most of the customer requirements, assuring that VoLTE has a competitive advantage when compared to OTT services in the mobile voice call services.

Finally in the Messaging services scope and having in mind that mobile devices change the way we communicate by enabling mobile and ubiquitous learning, Simon So [27] evaluated the use of mobile instant messaging tools to support teaching and learning in higher education. A total of 61 undergraduate students enrolled at a teacher-training institute in Hong Kong who have smartphones with WhatsApp and splitting the students into an experimental and a control group. Besides the traditional classroom learning for both groups, the experimental group was also supported with multimedia materials and teacher-student interaction via WhatsApp outside school hours. The study concluded that the intervention of WhatsApp improved the learning achievement and that the participants showed positive perception and acceptance of the use of OTT services for teaching and learning.

2.2.4. Traffic Classification related works

In this subsection the most relevant works related with traffic classification are highlighted. Regarding the considered categories for this topic, the following two were defined following the objectives of this research project and the given definition of traffic classification on subsection 2.1.3: Statistic classification, focusing on papers closely related to the implementation of machine learning algorithms in traffic classification; Deep Packet Inspection, holding papers centered on the use of this mechanism for the classification process.

One of the primary components of network operations and management is the traffic classification which enables the enhancement of network services and security by the identification of the different applications that are used inside the network. In [28] Al-Naymat et.al., collect first-hand traffic dataset from five different VoIP and Non-VoIP applications that are used by the majority of Internet community (Skype, YouTube, Yahoo Messenger, GTalk and PayPal) using a testbed and subsequently perform a classification step through machine learning algorithms, specifically Random Forest, AdaBoost (J48) and MultiLayer Perceptron are implemented. Their work aims at showing that machine learning algorithms can obtain good results in terms of accuracy

(true positives) when flow statistics features such as packet length, cumulative byte, among others are used. It is important to mention that the dataset used in this paper is not available for download and use in other investigations. In [29] Shafiq et.al., perform a similar approach to the one presented by Al-Naymat and his coworkers, comparing four different machine learning algorithms (Support Vector Machine, C4.5 decision tree, Naive Bays and Bayes Net) using a set of flow features extracted using Netmate [30], a flow statistics calculator and classifying the Internet traffic by types of application represented in five different classes: WWW, DNS, FTP, P2P and Telnet. Their results show a good accuracy and processing times for each algorithm having the decision tree as the best classification model.

Continuing with the papers related with statistic classification, the behavioral segmentation of mobile network users to target different sectors of customers with efficient marketing strategies and ensure customer retention in light of the intense competition has vital importance for mobile operators. With this in mind Ghnemat and Jaser [31] use self-organizing maps (SOM), an unsupervised learning approach, to detect different usage patterns of mobile users. The proposed system is tested using a large sample of customers' data (duration of generated calls, data usage, number of sent SMS, among others) provided by a major mobile operator in Jordan. The study detected six different behavioral segments in this market and highlights the role of data users in modern mobile markets.

Now proceeding with the works related with the Deep Packet Inspection category, considering that traffic classification usually is involved with network security, in [32] Hung and his coworkers proposed a GPU-based multiple-pattern matching algorithm (Graphic Processing Unit) for filtering malicious packets by using a Bloom filter to inspect the packet payload by leveraging the high parallelism computing power of a GPU, demonstrating that the proposed algorithm significantly enhances performance over sequential payload inspection algorithms. In a similar way and having in mind that in terms of DPI hardware solutions are having a better performance than software solutions, Jayashree and Shivaskhankarappa [33] propose a hardware based methodology to improve DPI. Their work introduce Ternary Content Addressable Memory (TCAM), which could perform complete packet inspection (packet header and payload inspection) aimed at the detection of possible security threats inside a network.

2.2.5. Categorization of users in a mobile network - related works

This subsection presents the most relevant works related with the categorization of users inside mobile networks, aiming at knowing how researches and mobile operators categorize the users according to their consumption behavior. With such objective in mind only a categorization models subgroup is defined holding four papers. The most important ones will be briefly highlighted as follows.

Understanding the traffic characteristics and user behaviors in mobile data networks becomes critical for operators in the rapidly evolving market. To help cellular network operators design more appropriate networks and application developers create applications which could better serve user needs, Li, Yang and Ansari [34] characterizes mobile Internet traffic generated by Android, iOS, and Windows Phone platforms devices. Using traffic data collected from a major Chinese mobile operator's network, they explore and compare user behaviors of these three platforms from two aspects: traffic dynamics and user applications. Regarding traffic dynamics, the authors group the users by traffic volume in a time window of five minutes, splitting them in three groups, high, medium and low traffic (HT/MT/LT). On the other hand the user applications are classified in seven categories: Browser, IM, game, webio, download, SNS and appMarket. More in detail, browser refers to the web browser applications like Safari and Chrome. IM refers to the applications which offer instant message service, such as Microsoft messenger. Game refers to the game applications running on mobile devices. Weibo is a particular kind of applications in China which are similar to Twitter. SNS is short for Social Network Site, including a number of websites such as LinkedIn and Facebook; AppMarket refers to the applications which offer app downloading for users like Google Play Store. In terms of application categories, the authors conclude that browser and IM attract more users than the rest. In a similar way Jin et.al., [35] investigate the usage patterns of mobile data users from data gathered from a mobile operator in the United States by applying Markov model and other statistical tools to characterize the data users. Through their analysis a highly uneven behavior is observed across mobile users since most of the users access data services occasionally while a small number of users contribute to the majority of data usage in the network. Therefore the users are classified in two groups: normal users and heavy users. Furthermore the authors conclude that most of the "heavy" traffic is generated by video and audio streaming applications and popular social networks.

Another approach for user categorization is the one used by Yang and his coworkers [36] where the users' categorization is done in three different ways: Data usage, following a similar structure as the previous work based on the traffic volume generation and concluding in two groups (normal and heavy users); Mobility Pattern, considering that a user's mobility pattern has an important impact in the network resource allocation and defining four different groups (Non Mobility, Low Mobility, Normal Mobility and High Mobility users); Finally Application Usage, based on the type of applications accessed by the users and defining a total of eleven categories (social network, e-commerce, reading, video, music, online gaming, news, mail, App Store, search and advertising) highlighting that the most used applications within their dataset was social networks.

As a summary Table 2.2 presents the proposed scheme for the papers classification along with a brief description of each defined thematic group.

The next subsection will focus on the gaps identified after the review of all the papers that were found.

Table 2.2. Related works classification.

Thematic groups	Categories	Related works	Description
SERVICE DEGRADATION	Business Models	[4], [10], [17], [18], [37]–[43]	These works focus on how to improve ISP's business models by implementing "data caps" and also how these types of policies can affect the user.
	Resource Control Techniques	[5], [19], [44]–[50]	These works are mainly focused on how to avoid the service degradation using different mechanisms such as "video prefetching", detection of alternative networks and packet compression in HTTP browsing.
QUALITY OF SERVICE (QoS)	Performance	[4]–[6], [20]–[22], [51]–[78]	These works focus on network analysis systems that generally are centered on parameters like throughput, latency and response time in networks and services.

OTT SERVICES	Business models	[1],[6],[23], [37]–[47]	These works focus on studying how OTT services have revolutionized the traditional ISP business model.
	Media	[86]–[92]	These works present different types of research focused on OTT video services.
	VoIP	[25], [26], [98]	These works present different types of research focused on OTT VoIP services.
	Messaging	[27]	These papers present different types of investigations focused on OTT messaging services.

TRAFFIC CLASSIFICATION	Statistic Classification	[94]–[98]	These works present studies mainly related to the use and comparison of machine learning algorithms in traffic classification.
	Deep Packet Inspection	[32], [33], [99], [100]	These works focus on investigations implementing deep packet inspection for traffic classification
USER CATEGORIZATION	Categorization models	[34]–[36], [101]	These works focus on studying user categorization models according to their consumption behavior.

2.2.6. Gaps

The previous subsections illustrate the different trends and focus of the most relevant related works structured in thematic groups. From the service degradation and QoS groups it can be stated that most of the works analyze the data caps from a business perspective and that the papers that present a functional prototype related with this topics usually try to avoid the implementation of the service degradation, or attempt to save resources in a way that the user do not exceeds his/her consumption limit. The works that belong to the OTT services group present a major trend to discuss the revolution provoked by this new entity in the ICT market and the other works analyze different aspects of video, messaging and VoIP applications highlighting that video applications are usually the focus for most investigations. The works related to traffic classification are divided in DPI and machine learning techniques illustrating different perspectives and results, however there are no works that consider the implementation of these techniques to propose a scheme of personalization of the service degradation policies. Finally the papers related to the categorization of users on mobile network present various models that can be considered and will depend on the users' behavior of the implemented dataset.

The Table 2.3 presents the different gaps identified in some of the highlighted works and subsequently a general summary of the identified gaps is illustrated.

- The works that are related to service degradation and data caps in general aim at creating ways to avoid having to implement this mechanism on the user and do not consider a personalization considering their consumption behavior.

- The papers related to OTT services focus on business models studies with the objective of identifying advantages for the ISP and mobile phone operators when applying this type of strategies without considering studies on consumption trends and categorization of users.

- Some works propose investigations related to OTT applications, however, most of them are centered on OTT media services (video), without considering other types of OTT services.

- Although there are works that implement machine learning algorithms (learning supervised and unsupervised) for the classification of Internet traffic, few of them considered the implementation of classification models that focus on the identification of specific applications as objective variables.

- Throughout the state of the art construction, no available dataset with information about the consumption of OTT applications generated by network users was found.

Service degradation related works		
Related work	**Contributions**	**Gaps**
[4]	- Describes how monthly bandwidth caps affect households' Internet use in South Africa.	- Does not propose any considerations related to the service degradation.
[17]	- Presents an analysis of QoS policies that are generally implemented in an LTE mobile network to manage network congestion, improve the QoS and qualify the services.	- Does not consider a proposal of QoS policies aimed at the personalization of the service degradation.
[10]	- Outlines an approach to understand and systematically analyze current and future business models based on data caps and their impact on customer behavior	- Does not consider a proposal of QoS policies aimed at the personalization of the service degradation within the business model.
[19]	- Presents a tool with the aim of being a proxy service for HTTP, extending the life time of users' data plans in mobile networks by reducing the size of the packages that are exchanged between servers and user equipment.	- Aims at avoiding the service degradation, hence does not consider a course of action after the users exceed their consumption limit.
Quality of Service related works		
Related work	**Contributions**	**Gaps**
[20]	- Performs a comparison between the different approaches of QoS description found in the literature where a large spectrum of models and meta-models to describe the service quality, ranging from ontological approaches to define quality measures, metrics,	- Does not consider analyzing the QoS related to OTT services. - Does not propose any considerations related to the service degradation.
[22]	- Proposes the Service Boost concept which is intended to provide a way for users to request the preferred network services (QoS) for selected applications and enable operators to monetize these preferred services by charging users and/or Application Content Providers	- Does not propose any considerations related to the service degradation.

[21]]	- The definition of three algorithms in the OTT VAN architecture which take application type or application quality into account. The aim is to address the application demands to achieve an effective network - An investigation of a user watching a YouTube video clip, while a Wi-Fi and a Cellular network are available in order to evaluate the user perceived quality, cellular usage and device power consumption.	- Does not propose any considerations related to the service degradation. - Only focuses on the consumption of OTT video applications.
OTT Services related works		
Related work	**Contributions**	**Gaps**
[24]	- Performs an extensive measurement study of Netflix and Hulu to uncover their architectures and service strategies aiming at helping in the design and implementation of future systems.	- Only focuses on the consumption of OTT video applications. - Does not propose any considerations related to the service degradation.
[23]	- Presents the regulatory responses enforced in some countries of the Caribbean - Proposes a regulatory framework that may aid in the effective management of OTT services and its evolution in the region	- The proposed framework aimed at the management ot OTT services does not specify details about the QoS needed for this services. - Does not propose any considerations related to the service degradation.
Traffic Classification related works		
Related work	**Contributions**	**Gaps**
[28]	- Collects a dataset from five different VoIP and Non-VoIP applications that are used by the majority of Internet community (Skype, YouTube, Yahoo Messenger, GTalk and PayPal). - Performs a classification process through machine learning algorithms: Random Forest, AdaBoost (J48) and MultiLayer Perceptron	- The dataset implemented in the paper is not available for further experimentation. - The classification process does not consider the user consumption behavior
[29]	- Compares four different machine learning algorithms (Support Vector Machine, C4.5 decision tree, Naive Bays and Bayes Net) using a set of flow features extracted using Netmate flow calculator.	- The dataset implemented in the paper is not available for further experimentation. - The classification process is aimed at types of applications i.e., WWW, DNS, FTP, P2P. - The classification process does not consider the user consumption behavior.
[31]	- Implements self-organizing maps (SOM) to detect different usage patterns of mobile users based on:	- The dataset implemented in the paper is not available for further experimentation.

	duration of generated calls, data usage, and number of sent SMS, among others.	- The consumption rate of OTT applications is not considered in the analysis.
Categorization of users in a mobile network – related works		
Related work	**Contributions**	**Gaps**
[36]	- Performs a categorization of users in three different ways: Data usage (normal and heavy users); Mobility Pattern (Non Mobility, Low Mobility, Normal Mobility and High Mobility users) and Application Usage based on the type of applications	- The dataset implemented in the paper is not available for further experimentation. - Does not propose any considerations related to the service degradation.
[34]	- Characterizes mobile Internet traffic generated by Android, iOS, and Windows Phone platforms. - Explores and compares user behaviors from two aspects: traffic volume and user applications.	- The dataset implemented in the paper is not available for further experimentation. - Does not propose any considerations related to the service degradation.
[31]	- Implements self-organizing maps (SOM) to detect different usage patterns of mobile users based on: duration of generated calls, data usage, and number of sent SMS, among others.	- The dataset implemented in the paper is not available for further experimentation. - The consumption rate of OTT applications is not considered in the analysis.

Table 2.3. Identified gaps.

Summary

This chapter presented the concepts related with this research project such as data caps or service degradation, Quality of Service (QoS) and Traffic Classification. Furthermore an analysis of the related works was performed dividing them into five groups: Service degradation, divided in two subcategories - Business models, with works focusing on how to improve ISP business models by implementing data caps and also how these types of policies can affect the user, and Resource Control Techniques, holding works mainly focused on how to avoid the service degradation using different mechanisms; Quality of Service, holding one subcategory – Performance, holding works focusing on network analysis systems that generally are centered on parameters like throughput, latency and response time in networks and services; OTT Services, divided into four subcategories – Business models, holding works that focus on studying how OTT services have revolutionized the traditional ISP business model – Media, holding works focused on different investigations applied on

OTT video applications – Messaging, illustrating works aimed at the research of instant messaging OTT applications and VoIP, showing works aimed at the investigation of Voice over IP OTT services; Traffic Classification, divided in two subcategories – Statistic classification, presenting studies mainly related to the use of machine learning algorithms in traffic classification, and Deep Packet, Inspection focusing on investigations implementing DPI for traffic classification; Finally Categorization of users in mobile networks illustrating works aimed at studying user categorization models according to their consumption behavior inside a mobile network.

Subsequently the different identified gaps are illustrated highlighting that even though there are works that contribute to the objectives of this research project, none of them considers the same final purpose of personalization of service degradation policies.

Chapter 3

Dataset Generation and Preprocessing

This chapter presents a detailed description of the different approaches that were implemented to obtain a dataset that enabled the study of users' OTT consumption behavior. A brief description of the gathering process of IP flows of four different OTT applications obtained as a first attempt on a controlled environment will be illustrated. Later on a description of the gathering process applied on a network section from Universidad Del Cauca that allow the generation of the main dataset that enabled the further results of this research project will be shown. Furthermore all the preprocessing applied to the dataset will be described highlighting the cleaning process and the clustering process.

3.1. Dataset on a controlled environment

This section presents the first attempt aimed at creating a dataset that would enable the study and identification of users' consumption behavior related to OTT applications. With this in mind, to create the dataset four OTT applications: YouTube, Sipp, Spotify and a chat application were used as the main traffic generators on four different user devices. In order to capture the IP packets and convert them into IP flows two software applications were used: Wireshark as the sniffer capable of capturing the IP packets

generated by each application in capture sessions with defined time intervals and Netmate as the flow statistics calculator. Both applications are briefly described as follows:

- **Wireshark:** is a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues. Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap (packet capture file format) to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows. There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License [102].

- **NetMate:** NETwork Measurement and AccounTing systEm is a network traffic meter and monitoring tool. Its job is to listen to network traffic, to classify packets into flows and to compute metrics for flows. The system architecture depicted on figure 3.1 basically follows the principle of the RTFM architecture with some slight changes. Netmate is the meter which listens on a network interface, classifies packets and computes metrics [30], [103].

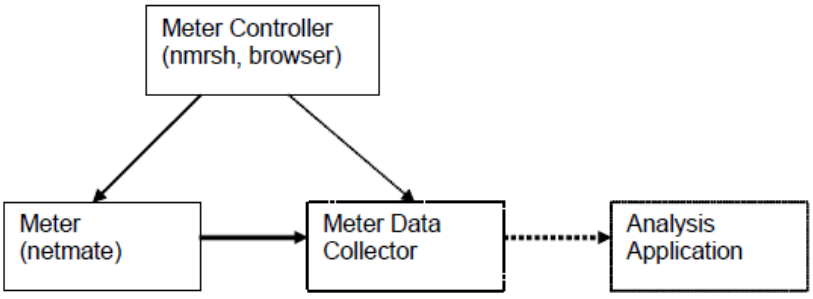

Figure 3.1. Netmate architecture [103].

Now with both tools described, Figure 3.2 illustrates the implemented scheme to capture the traffic generated by each application on the different user devices.

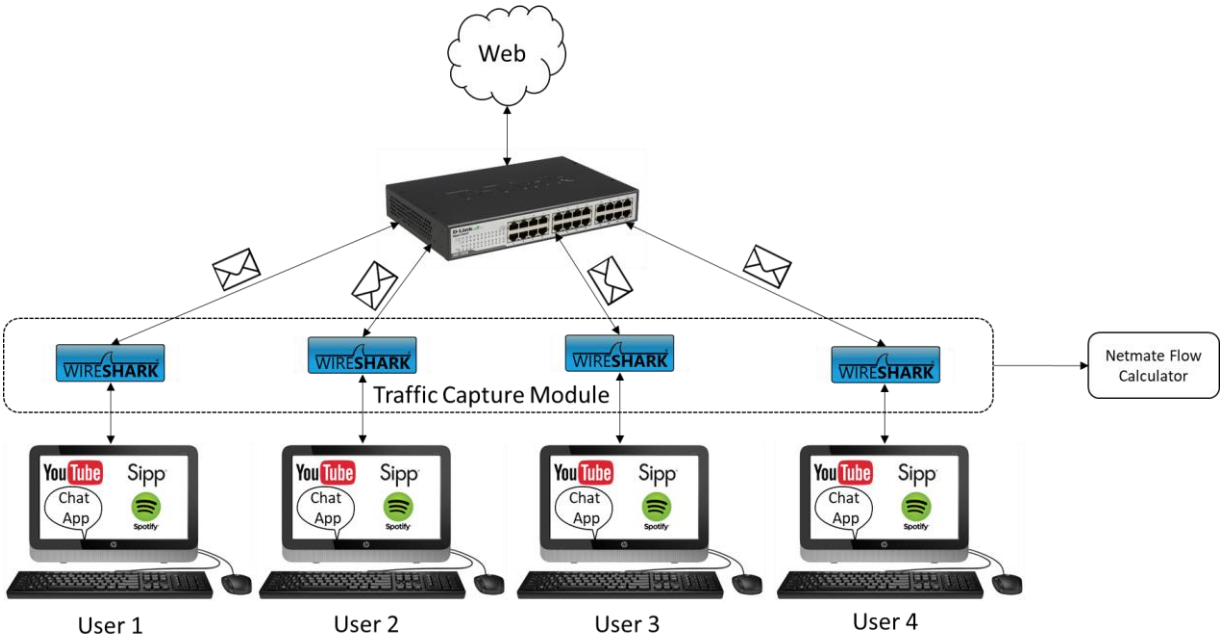

Figure 3.2. Controlled environment scheme.

As can be observed on the figure each device represented a user and had Wireshark installed to capture the packets and later on the obtained pcap files where processed with Netmate to obtain the flow statistics. Each user device had a static IP address and two groups of datasets where generated (two datasets per user device). The first group named original datasets contained all the IP flows exactly as they were captured during the experiment.

The second group named filtered datasets were obtained after applying a filter focusing on the flows that were originated by and directed to a user device i.e., all network control flows and communications between other devices among the network were removed. Besides on that same filter all the encrypted communications (Secure Socket Layer – SSL flows) were removed. Tables 3.1 and 3.2 illustrate the number of instances obtained for each dataset.

Dataset	Number of instances
User 1	10.388
User 2	13.011
User 3	13.765
User 4	11.913
Total	49.077

Table 3.1. Number of instances in the original datasets.

Dataset	Number of instances
User 1	4.589
User 2	4.017
User 3	3.953
User 4	3.509
Total	16.068

Table 3.2. Number of instances in the filtered datasets.

As illustrated before, among all the 8 datasets, a total of 65.145 instances were gathered, on each a dataset with a total of 44 attributes were generated using Netmate which will be described as follows.

3.1.1. Attributes List

This subsection describes all the 44 attributes generated by Netmate as they were stored on the datasets:

- ***srcip***: The source IP address of the flow.
- ***srcport***: The source port number of the flow.
- ***dstip:*** The destination IP address of the flow.
- ***dstport***: The destination port number of the flow.
- ***proto***: The transport layer communication protocol (i.e., TCP = 6, UDP = 17).
- ***total_fpackets***: The total number of packets sent in the forward direction i.e., from origin to destination.

- ***total_fvolume***: The total number of bytes sent in the forward direction.
- ***total_bpackets***: The total number of packets sent in the backward direction.
- ***total_bvolume***: The total number of bytes sent in the backward direction.
- ***min_fpktl***: The size of the smallest packet sent in the forward direction (in bytes).
- ***mean_fpktl***: The mean size of packets sent in the forward direction (in bytes).
- ***max_fpktl***: The size of the largest packet sent in the forward direction (in bytes).
- ***std_fpktl***: The standard deviation from the mean of the packets sent in the forward direction (in bytes).
- ***min_bpktl***: The size of the smallest packet sent in the backward direction (in bytes).
- ***mean_bpktl***: The mean size of packets sent in the backward direction (in bytes).
- ***max_bpktl***: The size of the largest packet sent in the backward direction (in bytes).
- ***std_bpktl***: The standard deviation from the mean of the packets sent in the backward .direction (in bytes).
- ***min_fiat***: The minimum amount of time between two packets sent in the forward direction (in microseconds).
- ***mean_fiat***: The mean amount of time between two packets sent in the forward direction (in microseconds).
- ***max_fiat***: The maximum amount of time between two packets sent in the forward direction (in microseconds).
- ***std_fiat***: The standard deviation from the mean amount of time between two packets sent in the forward direction (in microseconds).
- ***min_biat***: The minimum amount of time between two packets sent in the backward direction (in microseconds).
- ***mean_biat***: The mean amount of time between two packets sent in the backward direction (in microseconds).
- ***max_biat***: The maximum amount of time between two packets sent in the backward direction (in microseconds).
- ***std_biat***: The standard deviation from the mean amount of time between two packets sent in the backward direction (in microseconds).
- ***duration***: The duration of the flow (in microseconds).
- ***min_active***: The minimum amount of time that the flow was active before going idle (in microseconds).
- ***mean_active***: The mean amount of time that the flow was active before going idle (in microseconds).
- ***max_active***: The maximum amount of time that the flow was active before going idle (in microseconds).

- ***std_active***: The standard deviation from the mean amount of time that the flow was active before going idle (in microseconds).
- ***min_idle***: The minimum time a flow was idle before becoming active (in microseconds).
- ***mean_idle***: The mean time a flow was idle before becoming active (in microseconds).
- ***max_idle***: The maximum time a flow was idle before becoming active (in microseconds).
- ***std_idle***: The standard deviation from the mean time a flow was idle before becoming active (in microseconds).
- ***sflow_fpackets***: The average number of packets in a sub flow in the forward direction.
- ***sflow_fbytes***: The average number of bytes in a sub flow in the forward direction.
- ***sflow_bpackets***: The average number of packets in a sub flow in the backward direction.
- ***sflow_bbytes***: The average number of bytes in a sub flow in the backward direction.
- ***fpsh_cnt***: The number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP).
- ***bpsh_cnt***: The number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP).
- ***furg_cnt***: The number of times the URG flag was set in packets travelling in the forward direction (0 for UDP).
- ***burg_cnt***: The number of times the URG flag was set in packets travelling in the backward direction (0 for UDP).
- ***total_fhlen***: The total bytes used for headers in the forward direction.
- ***total_bhlen***: The total bytes used for headers in the backward direction.

Considering that the flows were gathered in a controlled way i.e., when the packet capture was being performed only one of the four applications was being used on the network by the user devices, all the information that was gathered was manually labeled with the four applications as the objective class (YouTube, Sportify, Chat, VoIP-Sipp).

3.1.2. Groups of Attributes

In order to observe in an easier way all the attributes gathered on this dataset a set of 6 groups were defined considering the information that each attribute presents. The Table 3.3 illustrates each group with the attributes assigned to it along with a brief description.

Groups of attribues	Attributes	Description
Network Identifiers	srcip; srcport; dstip; dstport; proto	These attributes hold all the information related to the source and destination of an Internet packet, i.e., IP addresses, communication protocol and ports.
Packet Descriptors	total_fpackets; total_fvolume; total_bpackets; total_bvolume; min_fpktl; mean_fpktl; max_fpktl; std_fpktl; min_bpktl; mean_bpktl; max_bpktl; std_bpktl	These attributes hold all the information related to the number of packets, volume, standard deviation, among others in the forward and backward direction.
Interarrival Times	min_fiat; mean_fiat; max_fiat; std_fiat; min_biat; mean_biat; max_biat; std_biat	These attributes hold all the information related to the interarrival times in the forward and backward direction
Flow Descriptors	duration; min_active; mean_active; max_active; std_active; min_idle; mean_idle; max_idle; std_idle	These attributes hold all the information related to the establishment and ending of the communication flow (minimum active duration, mean active duration, etc)
Subflow descriptors	sflow_fpackets; sflow_fbytes; sflow_bpackets; sflow_bbyte	If there were subflows, these attributes present all the information related to their duration, number and volume of packets, etc.
Header descriptors	fpsh_cnt; bpsh_cnt; furg_cnt; burg_cnt; total_fhlen; total_bhlen	Among these attributes the information related to the header flags is stored, i.e., the number of times the PSH flag was set, the number of times the URG flag was set and the total of header bytes used in the forward and backward direction.

Table 3.3. Groups of attributes - Controlled environment.

It is important to mention that at the end this dataset was discarded since the number of users was really small as well as the number of applications and instances. Furthermore it was not possible to guarantee that the manual labeling process of the flows was correct since there could be network control information being sent among the user and network devices, hence the labeling process was imprecise. However this first approach provided important feedback information to extend the experiment scheme resulting in the dataset captured in a network section from Universidad Del Cauca which will be described in the next section.

3.2. Dataset Generated on Unicauca Network

In order to gather the required information to build the dataset, with the collaboration of the network support division from Universidad Del Cauca, a computer equipped with Wireshark was placed inside the campus network core and was configured as a mirror of a main router in order to capture all the IP traffic from a network section of the campus during six days – April 26, 27, 28 and May 9, 11 and 15 of 2017. All the captured data was stored on pcap files that were processed implementing the architecture illustrated in Figure 3.3.

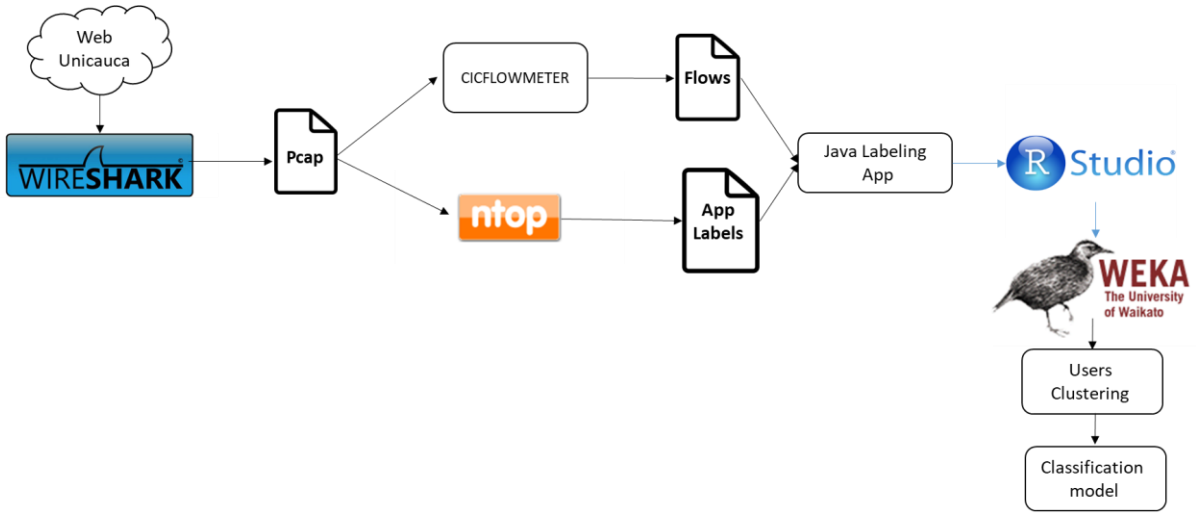

Figure 3.3. General architecture.

As can be observed on the figure, after the pcap files were obtained using Wireshark, all the files were processed using five different applications: CICFlowmeter, ntopng, Java labeling application, RStudio and Weka. In order to obtain a better understanding of the architecture a brief description of all the implemented software applications and their aims will be given in the next subsection with the exception of Wireshark which was already defined in section 3.1.

3.2.1. Software Applications implemented

- **CICFlowmeter:** is an open source network traffic flow generator which has been written in Java by the Canadian Institute of Cybersecurity from the University of New Brunswick. This software application offers high flexibility in terms of choosing the features that want to be calculated, adding new ones, and also having a better control of the duration of the flow timeout.

  CICFlowmeter generates Bidirectional Flows using pcap files as the source file. The first packet determines the forward (source to destination) and backward (destination to source) directions, therefore all the statistical features such as Duration, Number of packets, Number of bytes, Length of packets, among others are also calculated separately in the forward and backward direction.The output of the application delivers files is in the CSV file format where each flow has 85 different attributes. Each flow can be uniquely identified using six attributes: FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort and Protocol [104].

  It is important to mention that, although in the previous section where a dataset was created within a controlled environment the implemented tool to generate the flows statistics was Netmate, after a careful comparison and analysis, CICFlowmeter was selected over Netmate considering its better efficiency and larger number of attributes.

- **Ntopng:** is the next generation version of the original ntop, a network traffic probe that monitors network usage. Ntopng is based on Libpcap [105] and it has been written in a portable way in order to virtually run on every Unix platform, MacOSX and Windows as well. It provides an intuitive web user interface for the exploration of real-time and historical traffic information. It comes in three different versions: Community, Professional and Enterprise. The Community version is free to use and open source code can be found on Github [106]. The Professional and Enterprise versions offer some extra features that are particularly useful for larger organizations. Considering such features the Enterprise version of ntopng was used within this research project through an

academic license granted to Universidad Del Cauca by the developers. Some of the main features of this software application are stated as follows [107]:

- Discover application protocols (Facebook, YouTube, BitTorrent among other 224 protocols) by leveraging nDPI, ntop Deep Packet Inspection (DPI) technology.
- Support for MySQL, ElasticSearch and LogStash export of monitored data.
- Interactive historical exploration of monitored data exported to MySQL.
- Sort network traffic according to many criteria including IP address, ports, Layer 7 protocol (application) and throughput.
- Show real-time network traffic and active hosts.
- Monitor and report live throughput, network and application latencies, Round Trip Time (RTT), TCP statistics (retransmissions, out of order packets, packet lost), and bytes and packets transmitted.
- Store on disk persistent traffic statistics to allow future explorations and post-mortem analyses
- Analyse IP traffic and sort it according to the source and destination.

Specifically ntopng was necessary since CICFlowmeter was limited to calculating the flow statistics only, therefore ntopng was used with the aim of obtaining the layer 7 protocols (e.g., YouTube, Facebook, WhatsApp, etc.) of each flow generated with CICFlowmeter from the pcap files. This was accomplished using Deep Packet Inspection (DPI) through the nDPI feature embedded in ntopng. The process consisted on uploading the pcap file to ntopng, apply nDPI on each file, import the obtained layer 7 protocols to a MySQL database and convert those databases to CSV files.

- **Java Labeling Application:** This application was developed within the framework of this research project. It is a Java desktop application that enables the labeling of each flow obtained with CICFlowmeter with their respective layer 7 protocol (application) label obtained with nDPI from ntopng [108]. The development of this application was needed considering that the application labels were in CSV files apart from the ones obtained with CICFlowmeter. Therefore an algorithm capable of comparing and labeling each flow with their respective application was needed. The Figure 3.4 illustrates the algorithm developed to accomplish such purpose.

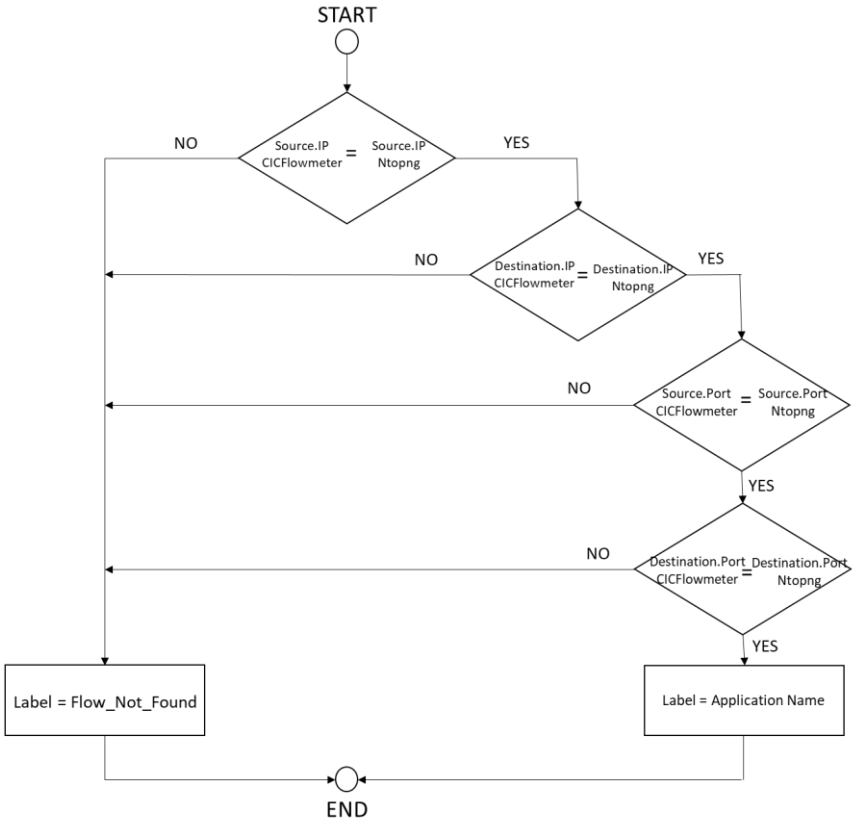

Figure 3.4. Labeling algorithm.

As can be observed on Figure 3.4 the labeling process performs a comparison between four different values (source and destination IP addresses and source and destination ports) contained in the CSV files generated by both CICFlowmeter and Ntopng. If all the conditions are met then two new attributes are added to the CSV file from CICFlowmeter: Application code, an integer that represents the application within the nDPI source code varying from 0 to 226 and Application name, a string illustrating the application that is being consumed on that Internet flow (e.g., Google, Facebook, YouTube, etc.). If one of the four conditions are not met the flow was labeled with code 226 and name Flow_Not_Found. After this process the dataset ended up with 87 attributes.

- **RStudio:** is a free and open-source Integrated Development Environment (IDE) for R, a programming language for statistical computing and graphics. It was founded by Joseph Allaire. RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application and

RStudio Server, which allows access to the application by using a web browser while it is running on a remote Linux server and each are available in open source and commercial editions. It is written in C++ programming language and uses the Qt framework for its graphical user interface. Its latest version (1.1) was released on October 9, 2017 [109].

Within this research project, RStudio was utilized as the main preprocessing application which enabled all the analysis activities that were performed on the dataset including cleaning procedures (removal of missing and irrelevant data), the clustering analysis and the generation of ARFF files needed to create the classification models with Weka.

- **Weka:** It is a well-known suite or collection of machine learning algorithms for data mining tasks developed by the University of Waikato, New Zealand. The algorithms can either be applied directly to a dataset or called from Java code. It contains tools for data pre-processing, classification, regression, clustering, feature selection, association rules, and visualization. Furthermore it is also well-suited for developing new machine learning schemes [110].

  As illustrated in Figure 3.4, within this project's framework, Weka is essentially used together with RStudio for the clustering analysis and the comparison and generation of the classification models applied to the dataset by testing a set of different algorithms that will be illustrated in a subsequent section.

With this a brief description of all the software applications used has been given. Now the following subsection will be focused on presenting the total number of instances, the attributes description and the classes' distribution after the labeling process was completed.

3.2.2. Attributes Description & Classes Distribution

As mentioned at the beginning of section 3.2, this datasets were gathered by capturing packets on a section of the network from the campus in Universidad Del Cauca. The captured data was divided in six CSV files (one per day) captured on: April 26, April 27, April 28, May 9, May 11 and May 15 of 2017, doing capture sessions of thirty minutes. A total 16.545.768 instances were captured distributed as illustrated in Table 3.4.

Dataset	Number of attributes	Number of instances
26-04-2017	87	1.405.590
27-04-2017	87	5.037.080
28-04-2017	87	927.987
09-05-2017	87	2.308.883
11-05-2017	87	3.709.524
15-05-2017	87	3.156.704
Total	-	16.545.768

Table 3.4. Instances per day.

Each file has 87 attributes. It is worth mentioning that the units of each attribute were consulted via email with Dr. Arash Habibi Lashkari from University of New Brunswick [111], one of the developers of CICFlowmeter. The time related attributes are measured in seconds and the digital information units are measured in bytes. All of the attributes, with their names as stored on the first version of the dataset, are described as follows:

- **Flow ID - (nominal):** a flow identifier following the next format:

  SourceIP-DestinationIP-SourcePort-DestinationPort-Protocol

  It is important to remember that CICFlowMeter generates bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions.

- **Source IP - (nominal):** The source IP address of the flow.

- **Source Port - (numeric):** The source port number.
- **Destination IP - (nominal):** The destination IP address.
- **Destination Port - (numeric):** The destination port number.
- **Protocol - (numeric):** The transport layer protocol number identification (i.e.,TCP = 6, UDP = 17).
- **Timestamp - (nominal):** The instant the packet was captured stored in the next date format:

  Dd/mm/yyyy HH:MM:SS

- **Flow Duration - (numeric):** The total duration of the flow (in seconds).
- **Total Fwd Packets - (numeric):** The total number of packets in the forward direction.
- **Total Backward Packets - (numeric):** The total number of packets in the backward direction.
- **Total Length of Fwd Packets - (numeric):** The total quantity of bytes in the forward direction obtained from all the flow (all the packets transmitted). This is obtained from the Total Length field stored on the packets header.
- **Total Length of Bwd Packets - (numeric):** The total quantity of bytes in the backward direction obtained from all the flow (all the packets transmitted). This is obtained from the Total Length field stored on the packets header.
- **Fwd Packet Length Max - (numeric):** The maximum value in bytes of the packets length in the forward direction.
- **Fwd Packet Length Min - (numeric):** The minimum value in bytes of the packets length in the forward direction.
- **Fwd Packet Length Mean - (numeric):** The mean value in bytes of the packets length in the forward direction.
- **Fwd Packet Length Std - (numeric):** The standard deviation in bytes of the packets length in the forward direction.
- **Bwd Packet Length Max - (numeric):** The maximum value in bytes of the packets length in the backward direction.
- **Bwd Packet Length Min - (numeric):** The minimum value in bytes of the packets length in the backward direction.
- **Bwd Packet Length Mean - (numeric):** The mean value in bytes of the packets length in the backward direction.

- **Bwd Packet Length Std - (numeric):** The standard deviation in bytes of the packets length in the backward direction.
- **Flow Bytes S - (numeric):** The number of bytes per second in the flow.
- **Flow Packets S - (numeric):** The number of packets per second in the flow.
- **Flow IAT Mean - (numeric):** The mean value of the interarrival time of the flow (in both directions).
- **Flow IAT Std - (numeric):** The standard deviation of the interarrival time of the flow (in both directions) (in seconds).
- **Flow IAT Max - (numeric):** The maximum value of the interarrival time of the flow (in both directions) (in seconds).
- **Flow IAT Min - (numeric):** The minimum value of the interarrival time of the flow (in both directions) (in seconds).
- **Fwd IAT Total - (numeric):** The total Interarrival time in the forward direction (in seconds).
- **Fwd IAT Mean - (numeric):** The mean interarrival time in the forward direction (in seconds).
- **Fwd IAT Std - (numeric):** The standard interarrival time in the forward direction (in seconds).
- **Fwd IAT Max - (numeric):** The maximum value of the interarrival time in the forward direction (in seconds).
- **Fwd IAT Min - (numeric):** The minimum value of the interarrival time in the forward direction (in seconds).
- **Bwd IAT Total - (numeric):** The total Interarrival time in the backward direction.
- **Bwd IAT Mean - (numeric):** The mean interarrival time in the backward direction (in seconds).
- **Bwd IAT Std - (numeric):** The standard interarrival time in the backward direction (in seconds).
- **Bwd IAT Max - (numeric):** The maximum value of the interarrival time in the backward direction (in seconds).
- **Bwd IAT Min - (numeric):** The minimum value of the interarrival time in the backward direction (in seconds).
- **Fwd PSH flags - (numeric):** The number of times the packets sent in the flow had the pushing flag bit set as 1 in the forward direction. The Pushing flag allows to send information immediately without filling all the buffer size from a packet,

notifying the receptor to pass the packet to the application at once, it is very useful for real time applications.

- **Bwd PSH flags - (numeric):** The number of times the packets sent in the flow had the PSH (pushing) flag bit set as 1 in the backward direction.
- **Fwd URG flags - (numeric):** The number of times the packets sent in the flow had the URG (Urgent) flag bit set as 1 in the forward direction. The URG flag is used to inform a receiving station that certain data within a segment is urgent and should be prioritized. If the URG flag is set, the receiving station evaluates the urgent pointer, a 16-bit field in the TCP header. This pointer indicates how much of the data in the segment, counting from the first byte, is urgent.
- **Bwd URG flags - (numeric):** The number of times the packets sent in the flow had the URG (Urgent) flag bit set as 1 in the backward direction.
- **Fwd Header Length - (numeric):** The header length of the packets flow in the forward direction.
- **Bwd Header Length - (numeric):** The header length of the packets flow in the backward direction.
- **Fwd Packets S - (numeric):** The number of packets per second in the forward direction.
- **Bwd Packets S - (numeric):** The number of packets per second in the backward direction.
- **Min Packet Length - (numeric):** The minimum length of the packets registered in the flow (both forward and backward directions).
- **Max Packet Length - (numeric):** The maximum length of the packets registered in the flow (both forward and backward directions).
- **Packet Length Mean - (numeric):** The mean value of the length of the packets registered in the flow (both forward and backward directions).
- **Packet Length Std - (numeric):** The standard deviation of the length of the packets registered in the flow (both forward and backward directions).
- **Packet Length Variance - (numeric):** The variance of the length of the packets registered in the flow (both forward and backward directions).
- **FIN Flag Count - (numeric):** The number of times the packets sent in the flow had the FIN flag bit set as 1. In the normal case, each side terminates its end of the connection by sending a special message with the FIN (finish) bit set. This message, sometimes called a FIN, serves as a connection termination request to the other device, while also possibly carrying data like a regular segment. The

device receiving the FIN responds with an acknowledgment to the FIN to indicate that it was received. The connection as a whole is not considered terminated until both sides have finished the shutdown procedure by sending a FIN and receiving an ACK. Furthermore, the connection termination phase uses a four-way handshake, with each side of the connection terminating independently. When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After the side that sent the first FIN has responded with the final ACK, it waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections; this prevents confusion due to delayed packets being delivered during subsequent connections. A connection can be "half-open", in which case one side has terminated its end, but the other has not. The side that has terminated can no longer send any data into the connection, but the other side can. The terminating side should continue reading the data until the other side terminates as well. It is also possible to terminate the connection by a 3-way handshake, when host A sends a FIN and host B replies with a FIN & ACK (merely combines 2 steps into one) and host a replies with an ACK.

- **SYN Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the SYN (Synchronize) flag bit set as 1. The SYN (Synchronize) flag is the TCP packet flag that is used to initiate a TCP connection. A packet containing solely a SYN flag is the first part of the "three-way handshake" of TCP connection initiation. It is responded to with a SYN-ACK packet. Packets setting the SYN flag can also be used to perform a SYN flood and a SYN scan.
- **RST Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the RST (Reset) flag bit set as 1 - (An RST says reset the connection. It must be sent whenever a segment arrives which apparently is not intended for the current connection - FIN says, "I finished talking to you, but I'll still listen to everything you have to say until you're done" (Wait for an ACK) RST says, "There is no conversation. I am resetting the connection!").
- **PSH Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the PSH (Pushing) flag bit set as 1.
- **ACK Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the ACK (Acknowledged) flag bit set as 1. As mentioned before to establish a connection, TCP uses a three-way handshake. Before a client

attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may initiate an active open.

- **URG Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the URG (Urgent) flag bit set as 1.
- **CWR Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the CWR (Congestion Window Reduced) TCP flag set as 1. During the synchronization phase of a connection between client and server, the TCP CWR and ECE (Explicit Congestion Notification - Echo) flags work in conjunction to establish whether the connection is capable of leveraging congestion notification. In order to work, both client and server need to support ECN (Explicit Congestion Notification). To accomplish this, the sender sends a SYN packet with the ECE and CWR flags set, and the receiver sends back the SYN-ACK with only the ECE flag set. Any other configuration indicates a non-ECN setup.
- **ECE Flag Count - (numeric):** The number of times the packets sent in the flow (in both directions) had the ECE (Explicit Congestion Notification Echo) TCP flag set as 1.
- **Down Up Ratio - (numeric):** Download and upload ratio i.e., the rate of packets in terms of speed exchanged in both directions. Download for backward direction and upload for forward direction.
- **Average Packet Size - (numeric):** The average size of each packet. It is important to notice that Packet Length specify the size of the whole packet including the header, trailer and the data that send on that packet. But Packet Size specify only the size of the header on the packet.
- **Avg Fwd Segment Size - (numeric):** The average segment size observed in the forward direction. A TCP segment is the Protocol Data Unit (PDU) which consists of a TCP header and an application data piece which comes from the upper Application Layer. Transport layer data is generally named as segment and network layer data unit is named as datagram but when UDP is used as transport layer protocol the data unit is called UDP datagram since the UDP data unit is not segmented (segmentation is made in transport layer when TCP is used).
- **Avg Bwd Segment Size - (numeric):** Average Segment size observed in the backward direction.

- **Fwd Header Length 1 - (numeric):** The header length of the packets flow in the forward direction. This attribute has the exact same values than the attribute Fwd Header Length, hence it can be a bug on the CICFlowmeter software.
- **Fwd Avg Bytes Bulk - (numeric):** The average number of bytes in the different bulks of information sent in the forward direction. Bulk data transfer is a software-based mechanism designed to move large data file using compression, blocking and buffering methods to optimize transfer times.
- **Fwd Avg Packets Bulk - (numeric):** Average number of packets in the different bulks of information sent in the forward direction.
- **Fwd Avg Bulk Rate - (numeric):** Average number of bulks (groups of data) sent in the forward direction.
- **Bwd Avg Bytes Bulk - (numeric):** Average number of bytes in the different bulks of information sent in the backward direction.
- **Bwd Avg Packets Bulk - (numeric):** Average number of packets in the different bulks of information sent in the backward direction.
- **Bwd Avg Bulk Rate - (numeric):** Average number of bulks (groups of data) sent in the backward direction.
- **Subflow Fwd Packets - (numeric):** The average number of packets in a subflow in the forward direction. The core idea of multipath TCP is to define a way to build a connection between two hosts and not between two interfaces (as standard TCP does). In standard TCP, the connection should be established between two IP addresses. Each TCP connection is identified by a four-tuple (source and destination addresses and ports). Given this restriction, an application can only create one TCP connection through a single link. Multipath TCP allows the connection to use several paths simultaneously. For this, Multipath TCP creates one TCP connection, called subflow, over each path that needs to be used. The detailed protocol specification is provided in RFC 6824 [112].
- **Subflow Fwd Bytes - (numeric):** The average number of bytes in a subflow in the forward direction.
- **Subflow Bwd Packets - (numeric):** The average number of packets in a subflow in the backward direction.
- **Subflow Bwd Bytes - (numeric):** The average number of bytes in a subflow in the backward direction.
- **Init Win bytes forward - (numeric):** The total number of bytes sent in the initial window in the forward direction. TCP uses a sliding window flow control protocol. In

each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to buffer for the connection. The sending host can send only up to that amount of data, before it must wait for an acknowledgment and window update from the receiving host.

- **Init Win bytes backward - (numeric):** The total number of bytes sent in the initial window in the backward direction.
- **act data pkt fwd - (numeric):** Count of packets with at least one byte of TCP data payload in the forward direction.
- **min seg size forward - (numeric):** Minimum segment size observed in the forward direction.
- **Active Mean - (numeric):** The mean time a flow was active before becoming idle.
- **Active Std - (numeric):** Standard deviation time a flow was active before becoming idle.
- **Active max - (numeric):** Maximum time a flow was active before becoming idle.
- **Active min - (numeric):** Minimum time a flow was active before becoming idle.
- **Idle Mean - (numeric):** Mean time a flow was idle before becoming active.
- **Idle std - (numeric):** Standard deviation time a flow was idle before becoming active.
- **Idle max - (numeric):** The maximum time a flow was idle before becoming active.
- **Idle min - (numeric):** The minimum time a flow was idle before becoming active.
- **Label - (nominal):** The state of the flow (benign or not).
- **L7Protocol – (numeric):** This attribute represents the code number of the layer 7 protocol as obtained from nDPI in Ntopng application. It is a number that varies from 0 to 226 (e.g., 0 is labeled as Unknown application).
- **ProtocolName - (nominal):** This attribute is the objective class of the dataset. It holds the application name following the code number stored in the L7Protocol attribute (e.g., YouTube, Yahoo, Facebook, etc.).

With this all of the 87 attributes have been characterized. In order to offer an easier perspective of the attributes, seven groups have been defined as illustrated in Table 3.5. Each group holds different attributes taking into consideration the type of information represented by each of them.

Now, proceeding with the dataset description, after the labeling process was completed using a combination of procedures between CICFlowmeter (obtain flows statistics), Ntopng (obtain application labels) and the Java labeling application (label each flow

with their respective application), a total of 80 different layer 7 protocols (applications) were identified among the IP flows. Figure 3.5 depicts the number of flows that were identified for each application.

As can be appreciated on the figure there is a high number of flows labeled as “Flow_Not_Found” (12.230.352 flows) and “Unknown” (738.120 flows). After analyzing in detail it can be concluded that the instances labeled as “Flow_Not_Found” are flows that hold network control information only, therefore the Deep Packet Inspection engine implemented with Ntopng (nDPI) was unable to find a layer 7 protocol on this flows. For the “Unknown” labels it can be concluded that these are flows that have application layer information however nDPI does not have a pattern that represents such application on its database, hence such flows are identified as unknown for the DPI engine. On the other hand, several flows are labeled as “SSL” (Secure Socket Layer – 404.883 flows) the cryptographic protocol most used to provide security over internet communications before it was preceded by TLS (Transport Layer Security). Since these flows are encrypted, nDPI is unable to obtain the layer 7 protocol to recognize the application that is being used, leaving them in a similar situation as the ones labeled as unknown. This analysis illustrates that even though Deep Packet Inspection was implemented on the captured flows, there are still several shortcomings in the recognition of applications. Besides the analysis shows that a cleaning process is needed on the dataset, which will be illustrated in the subsequent subsection.

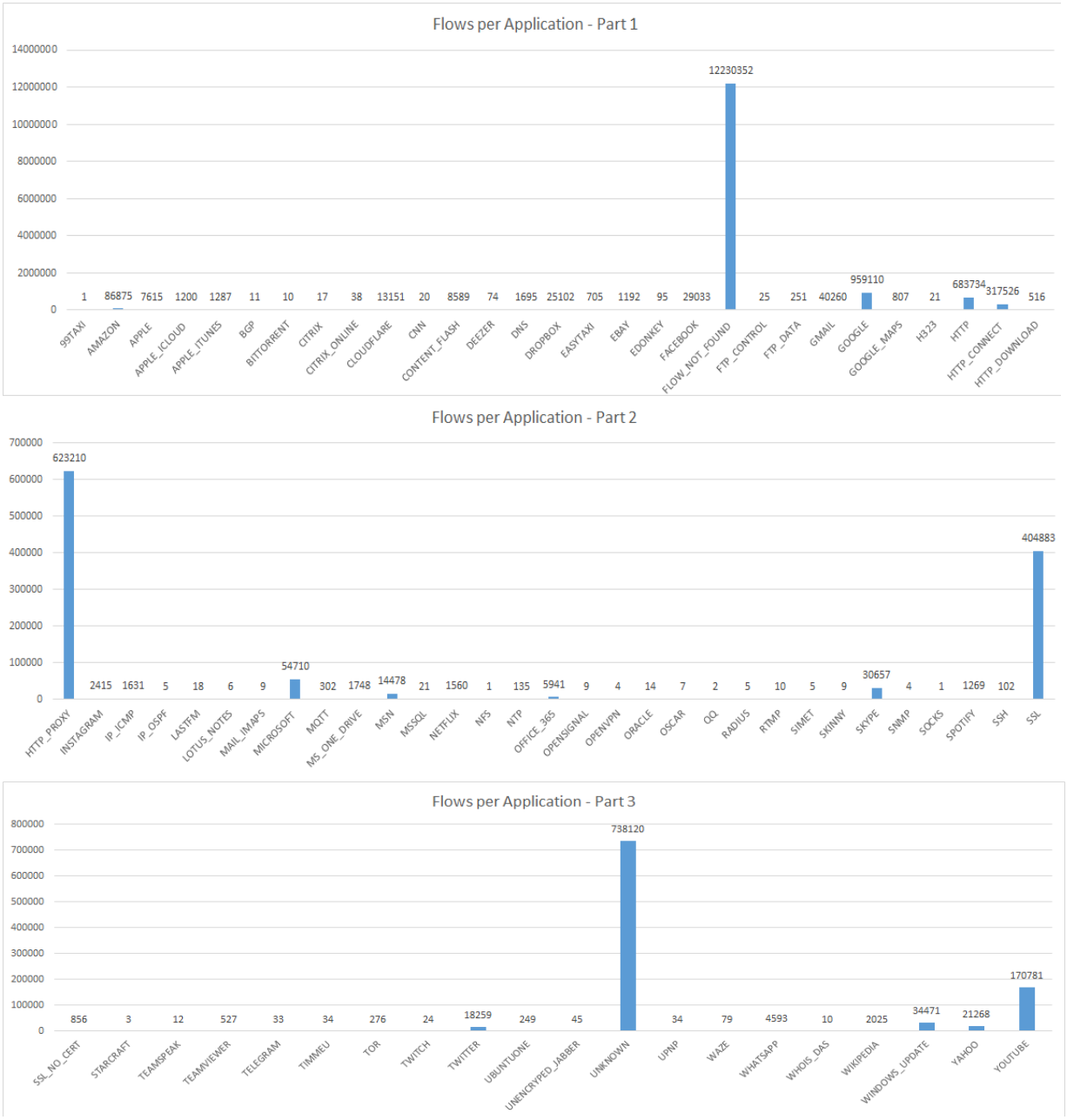

Figure 3.5. Flows per application - Dataset version 1.

Groups of attributes	Attributes	Description
Network Identifiers - 7 attributes	FlowID;Source IP; Source Port; Destination IP; Destination Port; Protocol; Timestamp	These attributes hold all the information related to the source and destination of an Internet flow, i.e., IP addresses, communication protocol and ports.
Flow Descriptors - 35 attributes	Total Fwd Packets; Total Backward Packets; Total Length of Fwd Packets; Total Length of Bwd Packets; Fwd Packet Length Max; Fwd Packet Length Max; Fwd Packet Length Mean; Fwd Packet Length Std; Bwd Packet Length Max; Bwd Packet Length Min; Bwd Packet Length Mean; Bwd Packet Length Std; Flow Bytes S; Flow Packets S; Min Packet Length; Max Packet Length; Packet Length Mean; Packet Length Std; Packet Length Variance; Down Up Ratio; Avg Fwd Segment Size; Avg Bwd Segment Size; Fwd Avg Bytes Bulk; Fwd Avg Packets Bulk; Fwd Avg Bulk Rate; Bwd Avg Bytes Bulk; Bwd Avg Packets Bulk; Bwd Avg Bulk Rate; Init Win bytes forward; Init Win bytes backward; act data pkt fwd; min seg size forward; Label; L7Protocol; ProtocolName	These attributes hold all the information related to the internet flow, i.e., number of packets, volume, standard deviation and application name among others in the forward and backward direction.
Interarrival Times - 15 attributes	Flow Duration; Flow IAT Mean; Flow IAT std; Flow IAT Max; Flow IAT Min; Fwd IAT Total; Fwd IAT Mean; Fwd IAT Std; Fwd IAT Max; Fwd IAT Min; Bwd IAT Total; Bwd IAT Mean; Bwd IAT Std; Bwd IAT Max; Bwd IAT Min	These attributes hold all the information related to the interarrival times in the forward and backward direction
Flag Features - 12 attributes	Fwd PSH flags; Bwd PSH flags; Fwd URG flags; Bwd URG flags; FIN Flag Count; SYN Flag Count; RST Flag Count; PSH Flag Count; ACK Flag Count; URG Flag Count; CWE Flag Count; ECE Flag Count	These attributes show the information related to all the flags contained in the header of the packets, i.e., Push flags, Urgent flags, Finish flags, among others.
Subflow descriptors - 4 attributes	Subflow Fwd Packets; Subflow Fwd Bytes; Subflow Bwd Packets; Subflow Bwd Bytes	If there were subflows, these attributes present all the information related to their number of packets per flow and volume in the forward and backward direction.
Header descriptors - 5 attributes	Fwd Header Length; Bwd Header Length; Average Packet Size; Fwd Header Length 1	Among these attributes the information related to the header is stored
Flow Timers - 8 attributes	Active Mean; Active Std; Active max; Active min; Idle Mean; Idle std; Idle max; Idle min	These attributes store the information related with the time each flow was active and inactive.

Table 3.5. Groups of attributes - Unicauca dataset.

3.2.3. Cleaning Procedures

Considering that at this point the dataset holds different flows that do not provide useful information for the proposal of personalized service degradation policies, a cleaning process was carried out aiming at four objectives listed as follows:

- Identify and remove missing values.
- Remove flows labeled as “Unknown”.
- Remove flows labeled as “Flow_Not_Found”.
- Remove flows labeled as “SSL”

Each procedure will be briefly described as follows:

Identify and remove missing values

Leveraging the tools provided by RStudio an analysis was performed on the datasets in order to find out if there were any missing values or values that hold no particular meaning among the instances. As consequence, it was identified that there were certain cases where the attributes “Flow.bytes.s” and “Flow.Packets.s” had instances with missing values (N.A.), infinite values (inf) and not a number (NaN).

After consulting the cause of these uncommon values with Dr. Arash Habibi Lashkari via email, it was concluded that this could happen if at the moment of the capture the number of packets and bytes per second was so small that CICFlowmeter was unable to perform the respective calculations. Therefore the stored value was assigned randomly between missing value, infinite and not a number.

With this in mind and implementing an R program in RStudio all the anomalous instances were removed from the dataset. Table 3.6 illustrates the number of anomalous instances found on each dataset.

Dataset	Number of anomalous instances
26-04-2017	67.903
27-04-2017	222.207
28-04-2017	30.574
09-05-2017	98.845
11-05-2017	131.427
15-05-2017	124.580
Total	675.536

Table 3.6. Anomalous instances per day.

Remove unsuitable labeled instances

Following a similar procedure as the one implemented for the missing values, using RStudio the instances that presented “Flow_Not_Found”, “Unknown” and ”SSL” labels were identified and removed from the dataset since these instances did not present accurate information about the application that was being consumed in that Internet communication.

At the end of the cleaning procedures a second version of the dataset was created leaving a total of 3.577.296 instances.

Moving on and aiming at the proposal of personalized service degradation policies it is necessary to perform an analysis to the dataset in order to identify users with similar behavior among the flows that can be grouped together. With this in mind the next subsection will present the clustering analysis that was performed.

3.2.3. Clustering analysis

After the second version of the dataset was obtained a clustering analysis was performed with the objective of identifying groups that gathered users with similar consumption behavior. As an approach to accomplish this, Self-Organized Maps (SOM) were implemented considering the analysis proposed by Ghnemat et.al., [31], however it is important to mention that there are other alternatives such as the elbow method, the average silhouette method or the gap statistic method which can be consulted here [113].

SOM is a methodology that belongs to the wider framework of unsupervised neural computing networks. It is applied to extract useful information from raw data via

mapping data samples presented in multi-dimensional space to a reduced space of two or three dimensions to help visualize hidden relations between the different samples. This is achieved by clustering samples that have common traits. The network consists of two layers of nodes; input and output where the input layer is fully connected to the output layer. Each node in the input layer is associated with one of the attributes that the data samples are comprised of. The nodes in the output layer are called neurons and they are put together in a two dimensional grid. Each neuron corresponds to an information vector with the same dimensions of the data samples [31].

With this in mind and following the clustering process presented in [31] an SOM of a sample (3731 instances) of the second version of the dataset was created with the Kohonen library implemented in RStudio. Considering the size of the sample an 18x17 SOM was created following the standard recommendation: $5\sqrt{N}$ [114], where N is the number of instances of the sample dataset and the result of this operation is the number of nodes of the SOM (306 nodes). With this the U-Matrix, a graph that can be used to identify the number of clusters within the SOM map, is obtained and it is illustrated on Figure 3.6. This visualization allows to observe the distance between each node and its neighbors.

Figure 3.6. U-Matrix.

By analyzing the U-Matrix it can be seen that there is only one cluster within the dataset with some anomalies that can be associated to some flows that had an unusual long time duration and mean packet size. Furthermore, by observing the correlation between the different attributes of the dataset through the elaboration of heatmaps it can be observed that in general most of the attributes maintain the same values regardless of the other variables which demonstrates that with this kind of distribution among the data there is at most one cluster only. The figures 3.7 and 3.8 illustrate eight heatmaps that demonstrates such uniform behavior among the attributes (considering the high number of plots and the extension of this document only some graphs are illustrated).

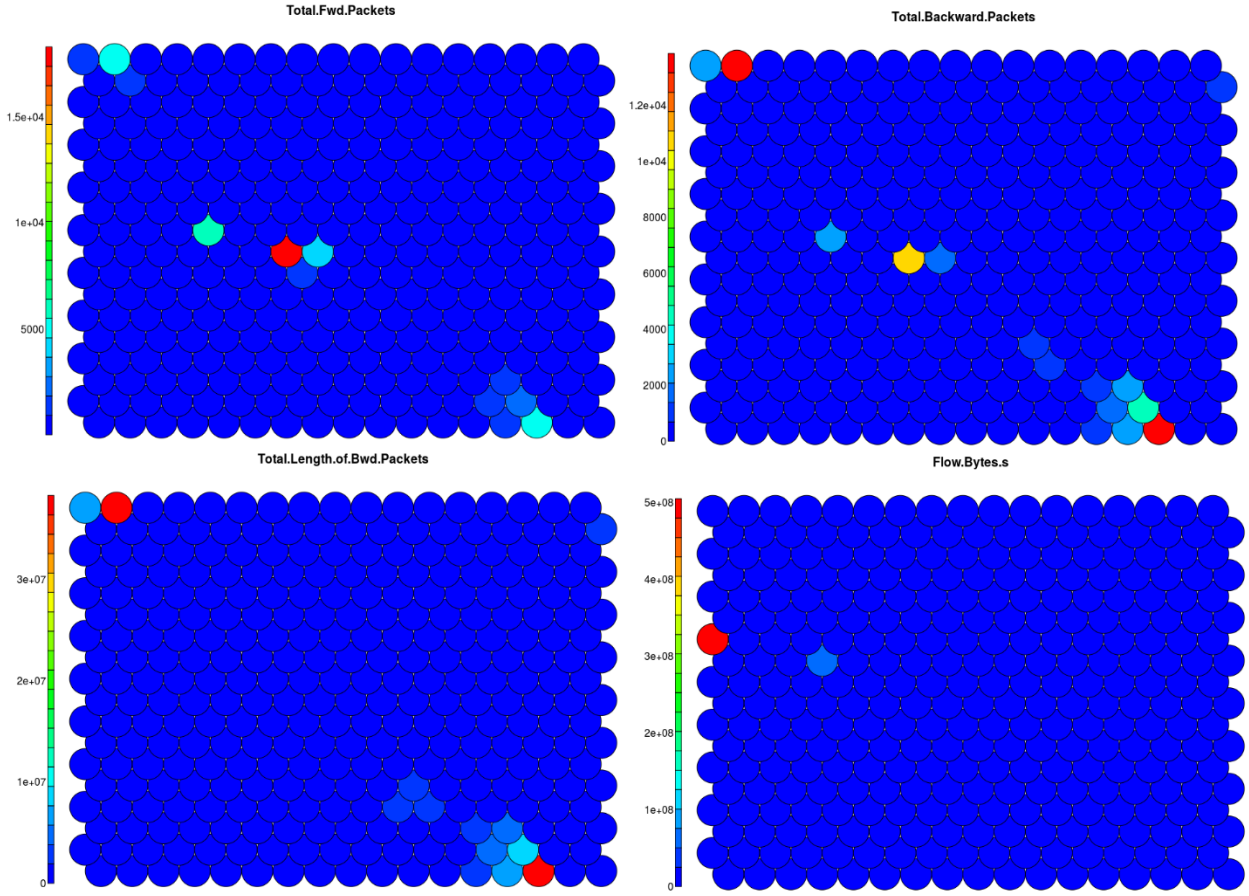

Figure 3.7. Heatmaps (a).

It is important to notice that in Figure 3.8 the heatmap of the consumed applications is presented (L7Protocol). This heatmap uses the code numbers from nDPI to identify the applications (a number between 0 and 200 where each number represents a specific application). By observing the figure it can be noticed that most the flows have an important trend towards the consumption of the same applications, specifically: HTTP

browsing (code 7 in nDPI), Facebook (code 119), Twitter (code 120), Google (code 126), YouTube (code 124), Spotify (code 156), Wikipedia (code 176) and Amazon (code 178).

Furthermore after observing the distribution of the other attributes it is not possible to find a pattern that allows to group users according to their consumption behavior since, as can be observed on each heatmap, most of the nodes have the same color i.e., most of the attributes on each flow maintain the same values making it impossible to differentiate some kind of trend that can conclude in the definition of multiple clusters. For this reason the U-matrix delivers only one cluster as a result.

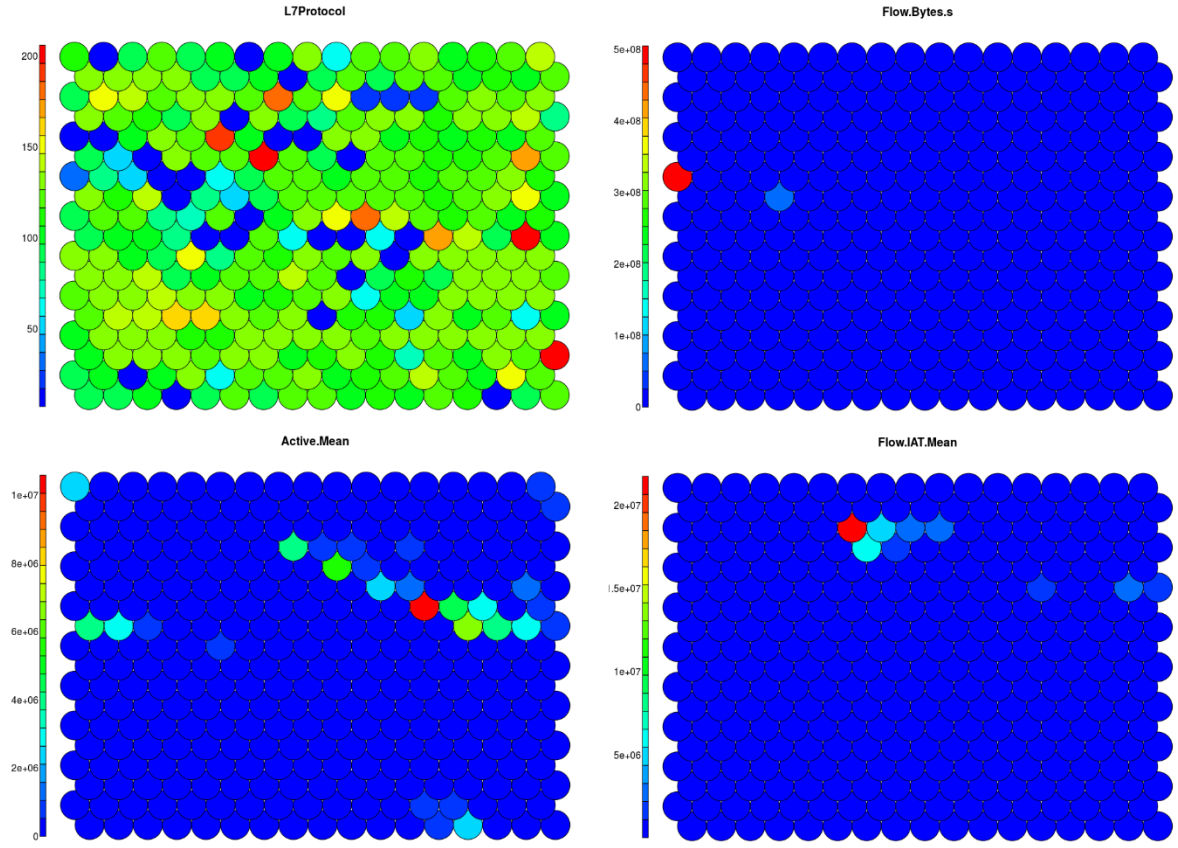

Figure 3.8. Heatmaps (b).

On the other hand, after a detailed analysis of the second version of the dataset it can be observed that among the instances there are flows generated not only by user devices but by network devices as well (e.g., servers, routers, switches, etc.). With this in mind and considering the results obtained with the Self-Organized Map it is concluded that:

- Clustering IP flows can provoke that flows generated by the same user are assigned to different groups.
- Having IP flows generated by network devices can harm the clustering aimed at the consumption behavior of users.
- It is not possible to create groups that represent the consumption behavior of a user using IP flow statistics since there is almost no difference among the features of two or more flows.

Therefore, as an alternative, it is concluded that a new dataset leveraging the gathered information must be created following the dataset used in [31] where each user is represented by an instance and their behavior is summarized.
To accomplish this task the first step was to differentiate the IP flows generated by users from the flows generated by network devices. In order to do this, following the recommendation of one of the experts from the network support division from Universidad Del Cauca, only the flows between the IP addresses 192.168.0.0 and 192.168.255.255 were considered since all these flows belong to user devices inside the campus network. Doing this a total of 1581 users were identified to create the new dataset.

As a second step it was necessary to define the attributes and applications that should be considered. With this in mind, from the 80 applications identified, a total of 32 applications were considered for the new dataset focusing on OTT applications only. Now, regarding the attributes that should be considered to summarize the consumption behavior of a user, leveraging the information already gathered, the following attributes were defined:

- User IP address to differentiate each user (in decimal format and in network format).
- Number of flows generated per application, since this allows to visualize which application is used the most by a user.
- Mean flow duration per application, aiming at observing on average how much time a user spends consuming a specific application.
- Average packet size per application, aiming at observing the impact an application can have in the data plan of a user.

- Average bytes per second per application, aiming at analyzing the average speed for each application.

With this a new dataset having 1581 instances (one per user) and 130 attributes is created. Figure 3.9 illustrates the number of flows per application in the new dataset illustrating a trend of consumption aimed mostly at Browsing, Google and YouTube.

Now as final step a new clustering analysis is performed on this new dataset to find out groups of users with similar behavior in order to propose personalized service degradation policies for each group, however it is not clear which is the optimal number of clusters. To solve this, the average silhouette method is implemented. This method measures the quality of a clustering process by determining how well each object lies within its cluster object. A high average silhouette width indicates a good clustering. Average silhouette method computes the average silhouette of observations for different values of k.

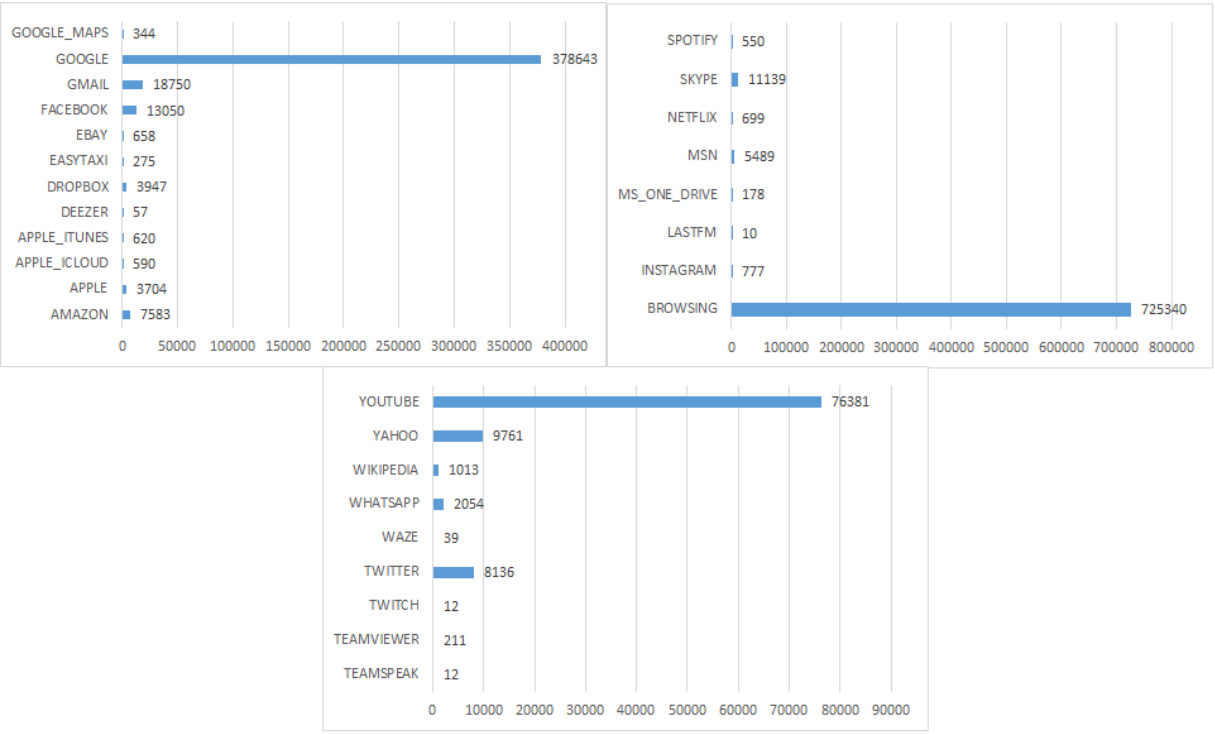

Figure 3.9. Flows per application - Final dataset.

The optimal number of clusters k is the one that maximize the average silhouette over a range of possible values for k [115], [113]. The analysis is performed with k between 1 and 20 clusters. Figure 3.10 illustrates the average silhouette width for each value of K.

The graph illustrates that the maximum width is between 2 and 3 clusters. However considering that the difference is infinitesimal among them, the optimal number of clusters selected for the new dataset is 3 clusters. Finally having the optimal number of clusters identified the clustering process is implemented with WEKA using a KMeans algorithm with K=3. With this process, a class label attribute is added to the dataset resulting in 131 attributes.

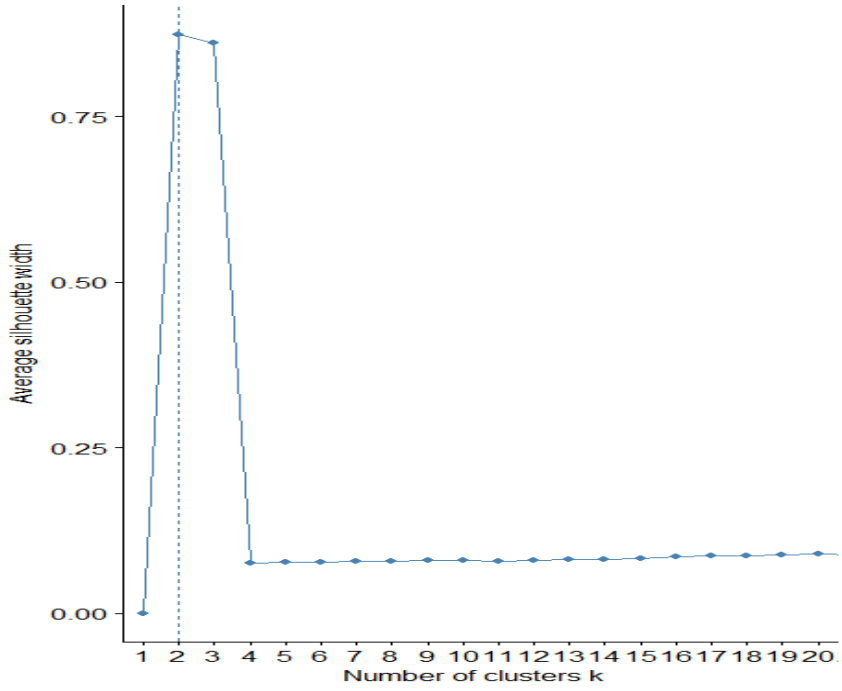

Figure 3.10. Silhouette method - Final dataset.

After analyzing the results obtained with WEKA it can be observed in the clusters distribution that although mostly all the users access the same applications, they vary in the intensity of their consumption. The users of cluster 1 consume the higher number of applications (14 applications) for the longest time: Amazon, Apple, Browsing,

Dropbox, Facebook, Gmail, Google, MSN, Skype, Twitter, Whatsapp, Wikipedia, Yahoo and YouTube; the users of clusters 2 consume 13 applications for a slightly minor time than cluster 1 and the consumed applications are: Amazon, Apple, Browsing, Dropbox, Ebay, Facebook, Gmail, Google, MSN, Skype, Twitter, Yahoo, YouTube; Finally cluster 3 contains the users that exhibit the behavior with the least consumption intensity in time and quantity of applications, consuming 5 applications: Amazon, Browsing, Facebook, Google and YouTube. Considering the previous statements and taking into consideration the characterization models presented in [35] and [36], cluster 1 was defined as the High consumption users, cluster 2 as the medium consumption users and cluster 3 as the low consumption users. The Figure 3.11 illustrates the number of instances (users) on each cluster and the Figure 3.12 illustrates the average generated flows by application on each cluster.

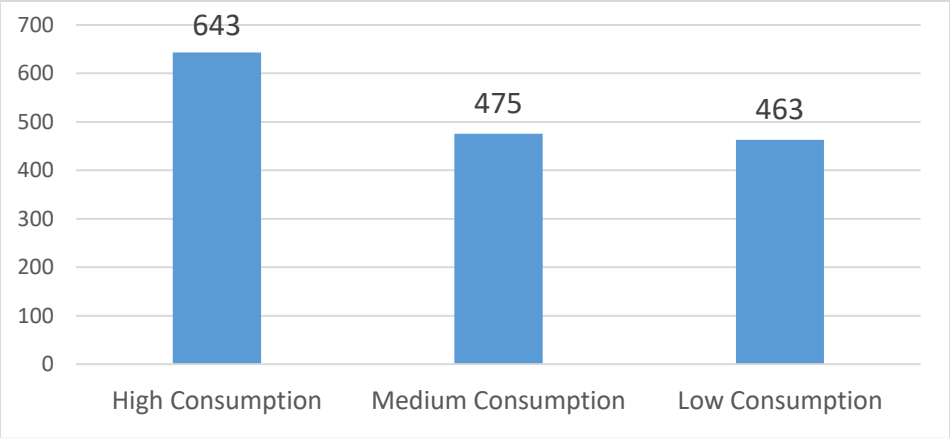

Figure 3.11. Users per cluster.

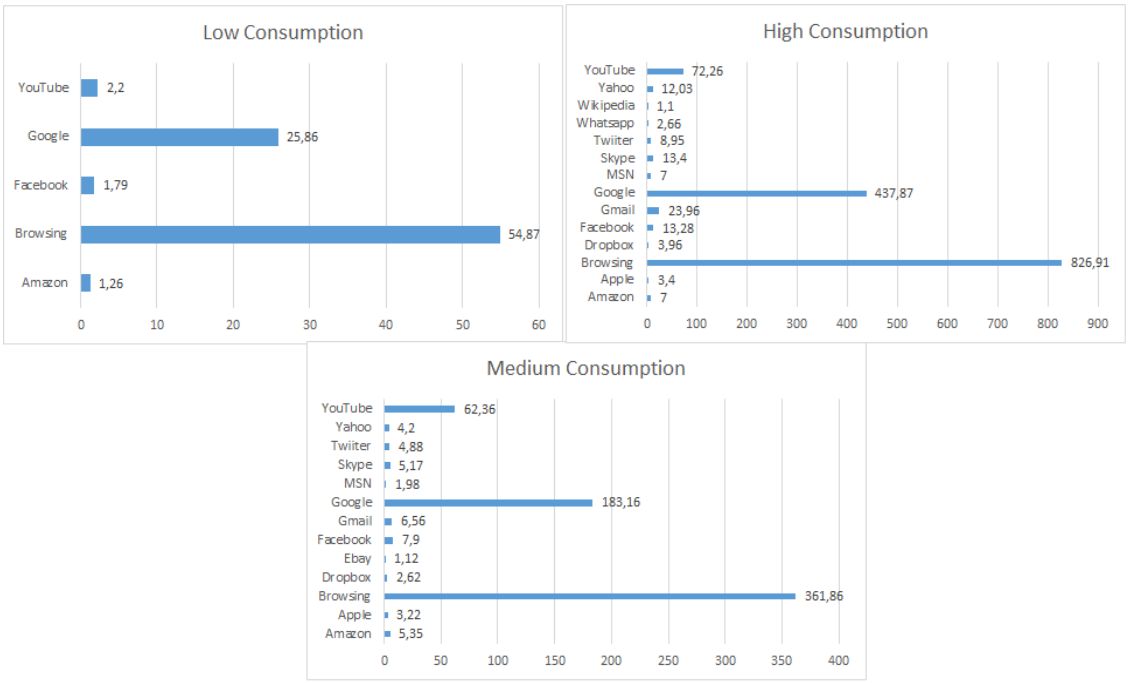

Figure 3.12. Average flows generated by application on each cluster.

With this the clustering analysis is concluded highlighting that at this point and starting from a traditional network monitoring, the parameters that describe the user consumption behavior have been defined (flows generated per application, mean flow duration per application, average packet size per application, average bytes per second per application) and that now it is possible to proceed with the classification modeling presented in the next chapter. All the dataset versions can be found in [116].

Summary

This chapter presented a detailed description of the different approaches and procedures that were implemented to create a dataset that enable the analysis and establishment of a set of groups based on the users consumption behavior of OTT applications in order to propose a set of personalized service degradation policies.

First the dataset generated on a controlled environment is presented highlighting all the tools implemented for its construction and the labeling process performed manually on

all the instances. This section concludes by stating that the number of users and OTT applications is not enough to achieve the goals considered in this research project. Later on the dataset generated on a network section from the campus of Universidad Del Cauca is presented highlighting: the software applications implemented to gather the required information; the characterization of the 87 attributes stored on the dataset and the cleaning process implemented which enabled the creation of two versions of the dataset.

Subsequently a clustering analysis is performed on the second version of the dataset by implementing Self-Organized Maps (SOM) on a sample of the dataset. Through this analysis the U-Matrix and the heatmaps are obtained enabling the observation of only one possible cluster considering that there are almost no differences between two or more IP flows. Furthermore it is concluded that a new dataset, leveraging the gathered information, is needed since the second version holds flows generated by user and network devices and such combination can harm the clustering process aimed at users behavior only. Finally the new dataset is created by focusing on the gathered information of 1581 users and summarizing their behavior using the flows generated per application, the mean flow duration per application, the average packet size per application and the average bytes per second per application. Then a new clustering process using the average silhouette method is performed obtaining 3 groups as the optimal number of clusters. Later on, by analyzing the clusters distribution the users are divided into 3 groups: High consumption, medium consumption and low consumption users.

Chapter 4

Classification Modeling

This chapter presents a detailed description of all the implemented procedures aimed at creating the best classification model using a set of different machine learning algorithms on the dataset described on the previous chapter. First a brief description of each algorithm is given, continuing with an illustration of the tests results and finally a discussion and conclusion based on those results.

4.1. Algorithms and Metrics

This section presents a brief definition of the implemented algorithms in all the classification tests along with the metrics that enable the analysis and conclusion on which model presents the best performance after the tests.

4.1.1 Algorithms definition

Boosting

Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms

that convert weak learners to strong ones [117]. Boosting is based on a simple question: Can a set of weak learners create a single strong learner?

A weak learner is defined to be a classifier that is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Specifically, the algorithm implemented is Adaptive Boosting (Adaboost) a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, it can be used in conjunction with many other types of learning algorithms to improve performance. The output of learning algorithms defined as weak learners is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner [117].

Bootstrap Aggregating – Bagging

Bootstrap aggregating, also called bagging, is an ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method.

It applies bootstrap sampling to obtain the data subsets for training the base learners. In detail, given a training dataset containing *m* number of training examples, a sample of *m* training examples will be generated by sampling with replacement i.e., the examples are taken randomly without changing the size of the original training dataset, therefore an example can be selected more than once. With this in mind, some original examples appear more than once, while some original examples are not present in the sample. By applying the process *T* times, *T* samples of *m* training examples are obtained. Then, from each sample a base learner can be trained by applying the base learning algorithm. Bagging adopts strategies for aggregating the outputs of the base learners, that is, voting for classification and averaging for regression. To predict a test

instance, in a classification problem for example, Bagging feeds the instance to its base classifiers and collects all of their outputs, and then votes the labels and takes the winner label as the prediction, where ties are broken arbitrarily [117]. Specifically the implemented algorithm is the bagging algorithm provided for classification tests in WEKA.

Support Vector Machines – SVM

In machine learning, Support Vector Machines (SVM) are supervised learning models that analyze data used for the construction of classification and regression models. Specifically, given a set of training examples, each labeled as belonging to one of two classes, an SVM training algorithm builds a model that assigns new examples to one class or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall [118].

In addition to performing linear classification, SVM can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Precisely, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier.

For the tests performed within this research project two algorithms used for SVM and embedded in WEKA's framework where implemented: LibSVM [119] and Sequential Minimal Optimization (SMO) [120]. All the parameters of each algorithm were optimized using Gridsearch [121] and linear, polynomial and radial based function kernels were tested.

Decision Tree – J48

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules [122]. Furthermore a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values of competing alternatives are calculated. A decision tree consists of three types of nodes:

- Decision nodes – typically represented by squares
- Chance nodes – typically represented by circles
- End nodes – typically represented by triangles

For the tests performed within this research project the J48 algorithm, a type of C4.5 decision tree deployed for classification purposes [123], was implemented to create and test the classification model.

K Nearest Neighbors – KNN

In pattern recognition, the k-nearest neighbors algorithm is a non-parametric method used for classification and regression [124]. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

For the tests performed, the IBK classification algorithm, provided by WEKA's framework, was implemented to visualize its performance on the final dataset exposed in the previous chapter.

Naive Bayes Classifier

Naive Bayes classifiers are a family of simple probabilistic classifiers based on the application of the Bayes' theorem assuming a strong (naive) independence between the features stored in the dataset [125]. A simple explanation of its operation process can be found in [126].

For the tests performed in this research project the Naïve Bayes implementation provided by WEKA is applied to the dataset and its performance is compared against the other classifiers.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification and regression, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (in classification) or mean prediction (in regression) of the individual trees [127].

It is an extension of Bagging, where the major difference is the incorporation of randomized feature selection. During the construction of a component decision tree, at each step of split selection, first random forests randomly selects a subset of features, and then carries out the conventional split selection procedure within the selected feature subset [117].

In a similar way to the other types of algorithms random forests is implemented for the tests using the tools provided by WEKA's framework.

The previous description of algorithms was intended to provide some context about the background and operation process of the classification models implemented in the tests of this research project. Furthermore each algorithm was selected with the intention of

testing several classification approaches and compare its performance when applied to the obtained dataset.

4.1.2 Classification metrics

Moving on and keeping the intention of providing some context about the classification scope in supervised learning algorithms, in this subsection a brief description of some of the most important and relevant classification metrics that should be considered when comparing different algorithm approaches will be presented.

Confusion Matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix [128], is a specific table layout that allows visualization of the performance of an algorithm, typically in supervised learning. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly misclassifying one as another). Figure 4.1 illustrates the structure of a confusion matrix along with the common performance metrics that can be calculated from it.

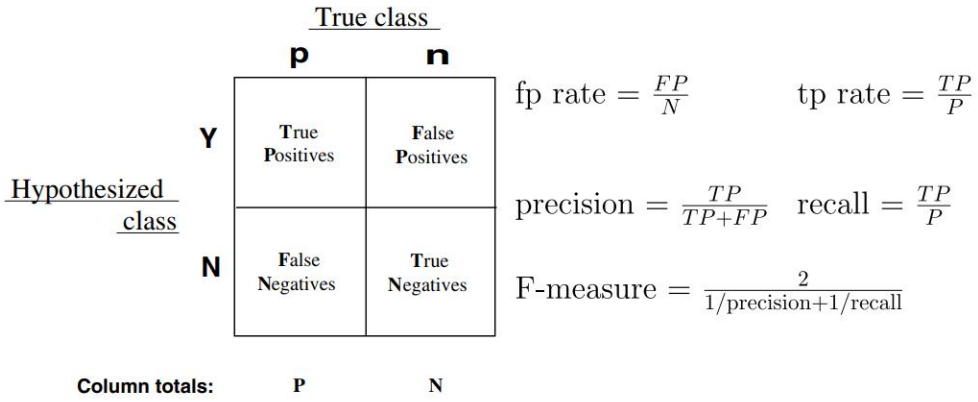

Figure 4.1. Confusion matrix and its performance metrics [128].

In a more precise way, given a binary classifier (positive and negative classes) and an instance, there are four possible outcomes. If the instance is positive and it is classified as positive, it is counted as a *true positive*; if it is classified as negative, it is counted as a *false negative*. If the instance is negative and it is classified as negative, it is counted

as a *true negative*; if it is classified as positive, it is counted as a *false positive*. As can be observed in figure 4.1 the true positive rate (TP - also called hit rate and recall) of a classifier is estimated as:

$$TP_{rate} = \frac{Positives\ Correctly\ Classified}{Total\ Positives}$$

The false positive rate (also called false alarm rate) of the classifier is:

$$FP_{rate} = \frac{Negatives\ Incorrectly\ Classified}{Total\ Negatives}$$

Additional terms associated with the performance of classification models include: Precision that addresses the question - Given a positive prediction from the classifier, how likely is it to be correct? Hence it can be interpreted as how many of the instances were correctly classified; Recall which addresses the question - Given a correctly classified instance, how likely is it for the classifier to detect it? i.e., how many of the true positives were recalled; and the F-measure which is related to a test's accuracy. It considers both the precision and the recall being the harmonic average of both of them and it reaches its best value at 1 (perfect precision and recall) and its worst at 0. All of this metrics were considered in the classification tests performed on the dataset and the obtained results will be presented in the subsequent section.

4.2. Classification tests & results

This section will present the results obtained from the classification tests performed on the final dataset illustrated on subsection 3.2.3. As mentioned before, a total of 8 algorithms were tested with the objective of identifying the best classification model in terms of performance. At first, each algorithm was tested individually and as a last step a corrected T test was performed in order to observe if there was any significant statistical difference. It is important to mention that, in some cases, the same algorithm was tested multiple times with different configuration settings aiming at finding the best

possible performance. With this in mind, the results obtained on every test will be illustrated as follows:

4.2.1. Adaboost with J48 classifier

The first approach implemented was a boosting algorithm with a decision tree as the main classifier. The algorithm was tested using 10, 100 and 1000 iterations having the best results with 100 iterations and was evaluated using a 10-fold cross validation method. The obtained results for each class from the dataset as well as the average of each metric with 100 iterations are illustrated on table 4.1 and table 4.2 depicts its confusion matrix.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.983	0.028	0.960	0.983	0.972
Low Consumption	0.888	0.005	0.988	0.888	0.936
Medium Consumption	0.935	0.056	0.873	0.935	0.903
Weighted Average	0.941	0.029	0.943	0.941	0.941

Table 4.1. Adaboost with J48 results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	632	0	11
Low Consumption	1	422	52
Medium Consumption	25	5	433

Table 4.2. Adaboost with J48 - Confusion Matrix.

Considering the total amount of instances for each class (643 for High Consumption, 475 for Medium Consumption and 463 for Low Consumption) both tables show that the classifier presented a good overall performance even though the highest classification errors are obtained for the low consumption instances which are classified as medium consumption instances, however such error rate can be considered small.

4.2.2. Bagging with J48 classifier

As a second approach a combination between the bagging algorithm and the J48 decision tree was considered. As in the previous case the algorithm was evaluated with a 10-fold cross validation method and was tested for 10, 100 and 1000 iterations obtaining the best results in the latter. Table 4.3 and table 4.4 illustrate the obtained results for the algorithm configured with 1000 iterations.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.966	0.031	0.955	0.966	0.961
Low Consumption	0.920	0.014	0.967	0.920	0.943
Medium Consumption	0.909	0.052	0.879	0.909	0.894
Weighted Average	0.935	0.032	0.936	0.935	0.936

Table 4.3. Bagging with J48 results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	621	0	22
Low Consumption	2	437	36
Medium Consumption	27	15	421

Table 4.4. Bagging with J48 - Confusion Matrix.

Similar to the previous case, in general, this algorithm presents good results with smaller precision and recall values and having a better differentiation between the low and medium consumption instances. However it presents a higher error rate when differentiating between high and medium consumption instances.

4.2.3. LibSVM and SMO optimized with Gridsearch algorithm

As a third approach LibSVM and SMO (Sequential Minimal Optimization) algorithms were implemented aiming at obtaining the best support vector machine. Considering that SVM have various parameters to be considered in the configuration settings Gridsearch was used to optimize the values that were best suited for the definition of the hyperplane. Each algorithm was tested with linear, polynomial and RBF (Radial Basis Function) kernels for the hyperplane, obtaining the best results with polynomial kernels and the worst results with RBF kernels. The optimized parameters with

Gridsearch were: gamma which defines how far the influence of a single training example reaches, with low values meaning a far reach and high values meaning close reach; and the cost parameter which tells the SVM optimization how much to avoid misclassifying each training example. For large values of the cost parameter, the optimization will choose a smaller-margin hyperplane. On the other hand, a very small value of this parameter will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points [129]. The obtained results for the best kernel functions (polynomial) of each algorithm are illustrated as follows.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.958	0.036	0.948	0.958	0.953
Low Consumption	0.962	0.021	0.952	0.962	0.957
Medium Consumption	0.883	0.038	0.907	0.883	0.895
Weighted Average	0.937	0.032	0.937	0.937	0.937

Table 4.5. LibSVM with polynomial kernel results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	616	1	26
Low Consumption	2	457	16
Medium Consumption	32	22	409

Table 4.6. LibSVM with polynomial kernel - Confusion Matrix.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.992	0.014	0.980	0.992	0.986
Low Consumption	0.981	0.006	0.985	0.981	0.983
Medium Consumption	0.961	0.011	0.974	0.961	0.967
Weighted Average	0.980	0.011	0.980	0.980	0.980

Table 4.7. SMO with polynomial kernel results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	638	0	5
Low Consumption	2	466	7
Medium Consumption	11	7	445

Table 4.8. SMO with polynomial kernel - Confusion Matrix.

As can be observed on the obtained results, both SVM algorithms present really good performances for the dataset highlighting that SMO is better than LibSVM. Both algorithms present a lower precision when classifying medium consumption users however by observing the confusion matrices it can be concluded that their error rate is small and in the case of SMO algorithm the best among all the algorithms so far.

4.2.4. J48 decision tree – KNN – Naive Bayes Algorithm

Moving on with the tests, in this subsection all the individual approaches are presented (i.e., algorithms without any kind of combination with other classification tools). For this approach J48 decision tree, K Nearest and the Naïve Bayes algorithm were considered. The number of neighbors for KNN was automatically optimized obtaining K = 23 neighbors. The obtained results for each model are presented below.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.956	0.038	0.945	0.956	0.951
Low Consumption	0.928	0.032	0.926	0.928	0.927
Medium Consumption	0.855	0.052	0.872	0.855	0.864
Weighted Average	0.918	0.040	0.918	0.918	0.918

Table 4.9. J48 decision tree results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	615	2	26
Low Consumption	2	441	32
Medium Consumption	34	33	396

Table 4.10. J48 decision tree - Confusion Matrix.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.983	0.025	0.965	0.983	0.974
Low Consumption	0.977	0.035	0.922	0.977	0.949
Medium Consumption	0.868	0.019	0.950	0.868	0.907
Weighted Average	0.948	0.026	0.948	0.948	0.947

Table 4.11. KNN results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	632	0	11
Low Consumption	1	464	10
Medium Consumption	22	39	402

Table 4.12. KNN - Confusion Matrix.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.605	0.091	0.821	0.605	0.697
Low Consumption	0.863	0.079	0.825	0.863	0.844
Medium Consumption	0.650	0.276	0.493	0.650	0.561
Weighted Average	0.696	0.141	0.726	0.696	0.701

Table 4.13. Naive Bayes results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	389	0	254
Low Consumption	10	410	55
Medium Consumption	75	87	301

Table 4.14. Naive Bayes - Confusion Matrix.

The results for this set of algorithms illustrate that KNN presents the best results among them and that Naïve Bayes algorithm presents the worst results among all the tests. Furthermore it is important to notice that even though J48 and KNN can be defined as good classifiers none of them overcome the SVM algorithms.

4.2.5. Random Forest

As the last approach the Random Forest algorithm was tested with the dataset obtaining the best performance with a total of 100 iterations. The results are illustrated as follows.

Objective class	TP Rate	FP Rate	Precision	Recall	F-Measure
High Consumption	0.997	0.081	0.894	0.997	0.943
Low Consumption	0.895	0.021	0.949	0.895	0.921
Medium Consumption	0.795	0.043	0.885	0.795	0.837
Weighted Average	0.907	0.052	0.908	0.907	0.905

Table 4.15. Random Forest results.

Classes / Classified as	High Consumption	Low Consumption	Medium Consumption
High Consumption	641	1	1
Low Consumption	3	425	47
Medium Consumption	73	22	368

Table 4.16. Random Forest - Confusion Matrix.

The results illustrate a similar behavior as the one obtained by the J48 algorithm highlighting a lower precision when classifying medium consumption users.

4.2.6. Statistical significance test

As final step all the algorithms were compared by performing a corrected T test in order to observe which model is the best classifier and if there are any significant statistical difference among their performances with a defined significance level of 0.05. With this in mind, table 4.17 illustrate the obtained results focusing on the percentage of correctly classified instances obtained by each algorithm compared with the SMO algorithm since it presented the best classification performance. The statistical difference is illustrated with three possible situations: V indicating a victory for the target algorithm (SMO); N indicating no significant statistical difference between the algorithms and L indicating a loss for the target algorithm.

Algorithms	SMO	LibSVM	Adaboost with J48	Bagging with J48	J48	KNN	Naïve Bayes	Random Forest
Correctly classified instances	97.97	93.73	94.05	93.54	91.84	94.75	69.57	90.70
Statistical difference (V/N/L)	-	N	N	N	V	N	V	V

Table 4.17. Statistical significance test results.

As can be observed in the previous table even though the SMO algorithm has the highest percentage of correctly classified instances, only in three cases out of eight, there is a significant statistical difference in the classification performance of the algorithms considering the set significance level of 5 percent.

Therefore it is possible to conclude that any algorithm among: SMO, LibSVM, Adaboost, Bagging and KNN can be considered as an adequate classification model to sort users on one of the three groups (high, medium and low consumption) considering their consumption behavior of OTT applications. In addition it is important to notice that two out of tree algorithms that have a significant statistical difference present a good performance behavior, leaving the Naïve Bayes algorithm as the only option that can be considered as a bad classification model for this kind of data.

Furthermore it is important to mention that it was not necessary to perform an speed analysis in the training and testing phase of any algorithms considering that the final dataset has 1581 instances only, therefore the processing of each algorithm did not take a considerable amount of time that provoked the need to perform such analysis.

Hence, the obtained results validate that machine learning algorithms are a feasible classification technique to differentiate a user's OTT consumption behavior, without leaving aside all the efforts and preprocessing steps that were needed to obtain the information that enable such classification.

Summary

This chapter presented a detailed description of all the classification modeling that was performed on the final dataset that was presented on chapter 3. As a first step, this chapter presents a brief description of the implemented algorithms along with a definition of some of the classification metrics that are usually considered in supervised learning tests.
On the second section of this chapter all the obtained results from each test are presented considering that initially each algorithm was tested individually and in some cases each algorithm was tested several times with different configuration settings aiming at identifying the best configuration possible. Subsequently a statistical significance test, with a significance level of 5 percent, is illustrated comparing all the algorithms against the one that presented the best classification results (SMO – Sequential Minimal Optimization – a support vector machine with a linear kernel function).

From these results it is possible to conclude that 5 of the 8 algorithms that were considered are a suitable option to classify a user's OTT consumption behavior and that a supervised learning approach from machine learning tools is a suitable classification technique for this kind of task without leaving aside all the efforts and preprocessing steps that were needed on the dataset to obtain the information that enabled the classification process.

Chapter 5

Personalized Service Degradation Policies

This chapter presents the proposal of personalized service degradation policies considering the results of the clustering analysis performed on the dataset and illustrated in chapter 3. All the proposal structure is based on the Policy and Charging Control (PCC) architecture developed for LTE (Long Term Evolution) networks, published on 3GPP technical specification 23.203 version 15 [130]. In order to provide some context the first section of this chapter presents a brief description of the PCC architecture along with the definition of the PCC rules, the mechanism that enables the implementation of QoS policies inside the network. Later on, the proposal of personalized service degradation policies aimed at the users' consumption behavior from the gathered dataset is presented.

5.1. Policy & Charging Control Architecture

An LTE service provider should be able to make services with different QoS requirements available to its users with different subscription levels. So, the service provider needs to be aware of subscription levels of each user (e.g. premium, best effort, etc.) and requested service types (e.g. Internet, voice, etc.) in order to assign radio and network resources to the traffic of each user and manage them properly.

With this need in mind, the Policy Control and Charging (PCC) architecture comprises seven elements: Policy and Charging Rules Function (PCRF), Policy and Charging Enforcement Function (PCEF), Bearer Binding and Event Reporting Function (BBERF), Online Charging System (OCS), Offline Charging System (OFCS), Subscription Profile Repository (SPR) and Application Function (AF). These functional entities are shown in Figure 5.1 with their logical reference points. PCC enables a centralized control to ensure that the service sessions (also called IP-CAN sessions) are provided with appropriate bandwidth and QoS. PCC also provides a means to control charging on a per-service basis [130].

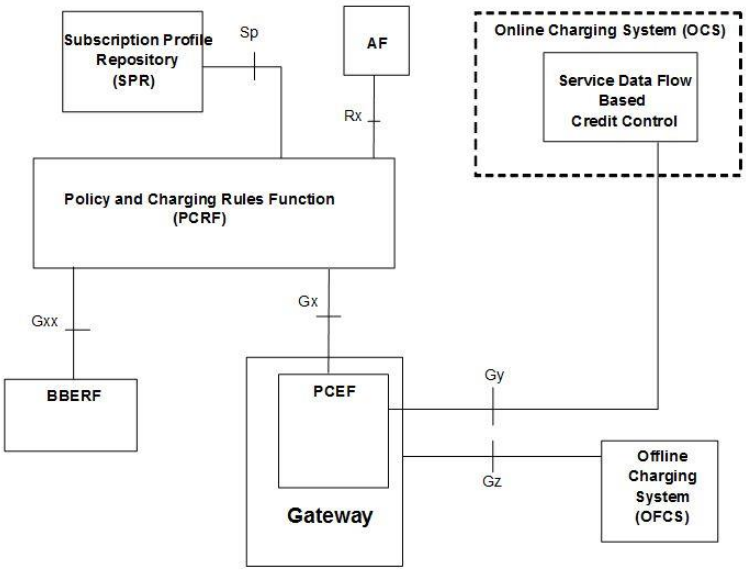

Figure 5.1. PCC Architecture [130].

Considering the objectives of this research project, the elements that are more relevant around QoS policies and service degradation are the PCRF (Policy and Charging Rules Function), the PCEF (Policy and Charging Enforcement Function), the SDF (Service Data Flow), EPS bearer (Evolved Packet System) and the PCC rules, which will be defined as follows.

5.1.1. Policy and Charging Rules Function

The Policy and Charging Rules Function (PCRF) provides network control regarding service data flow detection, gating (blocking or allowing packets), QoS control and flow-

based charging towards the PCEF. It can also apply security procedures before accepting information from the AF. The PCRF ensures that the PCEF user plane traffic mapping and treatment is in accordance with the user's subscription profile which it receives from SPR over Sp interface. The PCRF may reject the request received from the AF when the service information is not consistent with subscription information (either locally configured or received from SPR) and the PCRF responds to the AF with appropriate reason [130].

The PCRF accepts input for PCC decision making from the PCEF over Gx interface, the BBERF (if available), the SPR and the AF (if available) as well as its own pre-defined information. These nodes provide the following information to the PCRF:

- Subscriber Identifier
- IP address of the UE
- IP-CAN bearer attributes
- Request Type (Initial, Modification, etc)
- Type of IP-CAN (GPRS, etc)
- Location of Subscriber
- PDN ID
- PLMN Identifier
- IP-CAN bearer establishment mode

5.1.2. Policy and Charging Enforcement Function

The PCEF is a DPI (Deep Packet Inspection) device responsible for the enforcement of rules which have been configured statically or supplied to the PCEF dynamically from the PCRF (Policy and Charging Rules Function). The PCEF, located at the network gateway, provides policy enforcement as well as charging functionalities. The PCEF performs the role of a traffic controller distributing appropriate bandwidth as and when required. It also provides QoS at the gateway including service data flow detection, including varied interactions between online and offline charging. PCEF has a major role to play in monitoring service data flow regulated by policy control by allowing the service data to flow through the gateway only upon accessibility of the corresponding gate [130].

The PCEF along with the PCRF enable operators to provide differentiated service offerings in order to increase revenue earnings. Every service offering by the operator needs a different bandwidth which is where the PCEF and PCRF elements come into the picture. Their job is to ensure that network resources are utilized efficiently with minimum wastage by making sure required bandwidth is available to each service dynamically in real time.

5.1.3. Service Data Flow and EPS bearer

Figure 5.2 illustrates SDF and EPS bearers, and their relationship. In an LTE network, user traffic (IP flows or IP packets) is classified into SDF traffic and EPS bearer traffic. An SDF refers to a group of IP flows associated with a service that a user is using, while an EPS bearer refers to IP flows of aggregated SDFs that have the same QoS class.

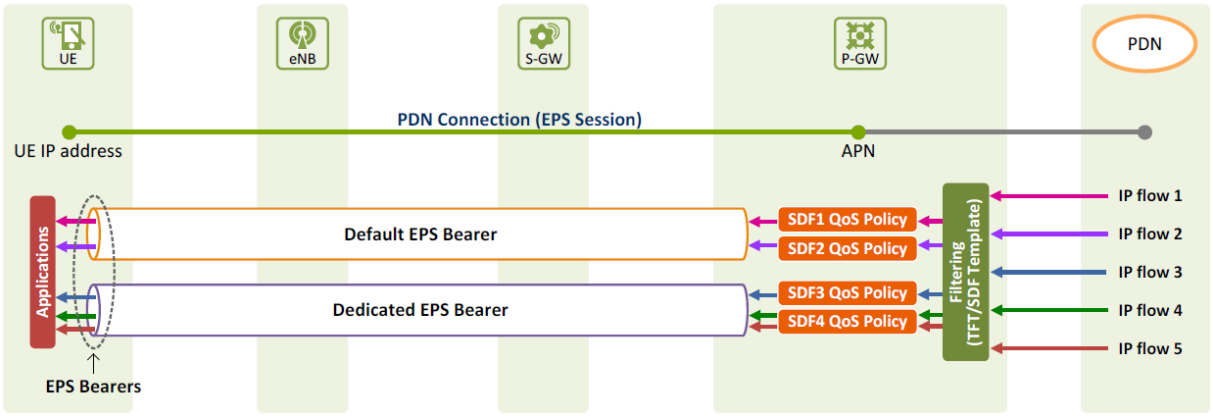

Figure 5.2. SDF and EPS bearer [131].

The SDF and EPS bearer are detected by matching the IP flows against the packet filters (SDF templates for SDF and Traffic Flow Templates (TFT) for EPS bearers). These packet filters are pre-configured by network operators in accordance with their policy, and each of them typically consists of 5-tuple data composed by the Source IP address, Destination IP address, Source port number, Destination port number and Protocol ID. In other words, in the LTE network, IP flows with the same service characteristics that match the packet filters of a SDF template are designated a SDF. SDF that match the packet filters of a TFT are mapped to an EPS bearer, to be finally delivered to a UE (User Equipment). SDFs with the same QoS class are delivered,

through an EPS bearer, whereas ones with different QoS class are delivered through different EPS bearers.

In a more precise way, A SDF is an IP flow or an aggregation of IP flows of user traffic classified by the type of the service in use. Different SDFs have different QoS class and hence a SDF serves as a unit by which QoS rules are applied in accordance with the Policy and Charging (PCC) procedure in the LTE network, each SDF is mapped to an EPS bearer that satisfies its QoS requirement, and then delivered to the User Equipment. On the other hand, there are two types of EPS bearers: default and dedicated. When a UE attaches to the LTE network, an IP address used in a PDN (Packet Data Network) is assigned and a default EPS bearer is established all at the same time. When a user who has been using a service through a default bearer (e.g. Internet browsing) attempts to use a service which requires higher QoS that the current default bearer cannot provide (e.g. Video on Demand), a dedicated bearer is established on demand. Thus, the dedicated bearer is established with QoS different settings from the one already sent in the existing bearer. A UE can be connected to more than one PDN, which must have one mandatory default EPS bearer and can have none to many optional EPS bearers. The number of EPS bearers a UE can have cannot exceed eleven bearers [131].

5.1.4. PCC rule

A PCC rule is an element defined by the PCRF for each SDF. The purpose of the PCC rule is to detect a packet belonging to a SDF, identify the service the SDF contributes to, provide applicable charging parameters for the SDF, and provide policy control for the SDF. Each rule, as defined by the PCRF, is forwarded to the PCEF over the Gx interface to be enforced for each SDF. After enforcing the PCC rules, when IP packets arrive, it detects the SDF that each packet belongs to, and applies a PCC rule to each packet according to their SDF.

There are two types of PCC rules: Dynamic PCC rules or pre-defined PCC rules. Dynamic PCC rules are dynamically provisioned by the PCRF to the PCEF when an EPS session is established or modified. Pre-defined ones are preconfigured in the PCEF, and thus can be activated or deactivated by PCRF [132]. A PCC rule consists

of: a policy rule name, service ID, SDF template, gate status, QoS parameters and charging parameters and varies depending on the operator's policy.

Considering the objectives stated on this research project, the charging parameters are not within its scope, however the QoS parameters are of vital importance in the definition of PCC rules related with service degradation. Among the parameters that must be taken into consideration when defining PCC rules are:

Element	Definition
Policy Rule Name	Is the name given to the PCC rule in order to be easily identified.
SDF Template	This is the packet filter pre-configured by network operators in accordance with their policy, and each of them typically consists of 5-tuple (Source IP address, Destination IP address, Source port number, Destination port number, and Protocol ID). The Protocol ID is checked with DPI.
GBR & Non-GBR (Guaranteed Bit Rate)	For an EPS bearer, having a GBR resource type means the bandwidth of the bearer is guaranteed, i.e., a GBR type EPS bearer has a "guaranteed bit rate" associated as one of its QoS parameters. Only a dedicated EPS bearer can be a GBR type bearer and no default EPS bearer can be GBR type. The QCI of a GBR type EPS bearer can range from 1 to 4. On the other hand, having a non-GBR resource type means that the bearer is a best effort type bearer and its bandwidth is not guaranteed. A default EPS bearer is always a Non-GBR bearer, whereas a dedicated EPS bearer can be either GBR or non-GBR. The QCI of a non-GBR type EPS bearer can range from 5 to 9.
MBR (Maximum Bit Rate)	This parameter indicates the maximum bit rate allowed in the LTE network. Any packets arriving at the bearer exceeding the specified MBR will be discarded.
	It's an integer from 1 to 9 that indicates nine different QoS performance characteristics of each IP packet. QCI values are standardized to reference

QCI (QoS Class Identifier)	specific QoS characteristics [130], and each QCI contains standardized performance characteristics (values), such as resource type (GBR or non-GBR), priority (1~9), Packet Delay Budget (allowed packet delay shown in values ranging from 50 ms to 300 ms), Packet Error Loss Rate (allowed packet loss shown in values from 10^{-2} to 10^{-6}.
ARP (Allocation and Retention Priority)	An integer ranging from 1 to 15, with 1 being the highest level of priority. When a new EPS bearer is needed in an LTE network with insufficient resources, the PCEF decides, based on the ARP, whether to: remove the existing EPS bearer and create a new one (e.g. removing an EPS bearer with low priority ARP to create one with high priority ARP) or refuse to create a new one.
SDF Gating Status	This parameter defines the gating status for the LTE bearer i.e., if a bearer is permitted to pass through (open) or if it is blocked (closed).
SDF Charging	This parameter indicates if the charging policies associated to the user equipment is online or offline.

Table 5.1. Elements of a PCC rule.

With this it is possible to proceed with the definition of the personalized service degradation policies presented in the following section.

5.2. Proposal of personalized service degradation policies

By analyzing the structure of a PCC rule and considering that in the technical specification the QCI parameter value defines the ARP and if a bearer is GBR (QCI 1 to 4) or non-GBR (QCI 5 to 9), there are only two possibilities that a service degradation policy can take: degrade the service by affecting the Maximum Bit Rate (MBR)

associated to the SDF (Service Data Flow) or block the service by modifying the gating control parameter.

Therefore, having in mind Figure 3.12 where the most consumed applications by each group are illustrated (5 applications for the low consumption group, 13 applications for the medium consumption group and 14 applications for the high consumption group), table 5.2 illustrates how is recommended to perform the service degradation for each cluster considering the consumption behavior.

Applications\Clusters	Low Consumption		Medium Consumption		High Consumption	
	Degrade service	Block service	Degrade service	Block service	Degrade service	Block service
Amazon		X		X		X
Apple		X		X		X
Browsing	X		X		X	
Dropbox		X		X		X
Ebay		X		X		X
Facebook		X	X		X	
Gmail		X	X		X	
Google	X		X		X	
MSN		X		X		X
Skype		X		X	X	
Twitter		X		X	X	
Whatsapp		X		X	X	
Wikipedia		X		X		X
Yahoo		X		X		X
YouTube		X	X		X	

Table 5.2. Policies recommendation.

When a service degradation is identified, i.e., when a user exceeds his/her consumption limit the recommendations for personalized QoS policies are: for the low consumption group only the most used applications are still functional with their bit rate degraded. For the Medium Consumption group the 5 most used applications are still functional and the rest are blocked. Finally for the High Consumption group the 8 most used applications are degraded in the bit rate and the others are blocked. This way the network administrator can save network resources and the degradation process is performed considering the behavior of the users. It is important to mention that, besides this set of recommendations, several possibilities that are better suited for a certain operator needs can be considered and that for this specific case each group of users is analyzed individually, therefore the criteria (higher average number of generated

flows per application) that decides if an application is degraded or blocked in one group is different for the other two.

As an example Tables 5.3 and 5.4 illustrates the structure of the dynamic PCC rules for the low and medium consumption groups respectively. The maximum bit rates defined for each policy can be considered degraded, since the speeds that can be offered in average on a LTE network according to the report presented by OpenSignal are 45.62 Mbps in Singapore and 19.07 Mbps in Colombia, besides that speed is not the sum of both upload and download links but one link only [133]. Furthermore it is important to notice three considerations: first, only one example for the blocking of a service is illustrated per group considering that the structure of the policy is identical except for the application protocol and the policy name; second, since all the traffic is from OTT applications, i.e., Internet traffic, the 3GPP recommendation states that this applications do not need a guaranteed bit rate (GRB) for their performance, therefore all the policies are for non-GBR bearers (best effort) and third the policies for the high consumption group would be similar to the ones proposed for the medium consumption group with a major number of degraded applications in their MBR permitted, therefore the illustration of the medium consumption group is enough to show the policies structure of both high and medium groups.

Policy Rule Name	SDF Template	SDF GBR	SDF MBR	SDF QCI/ARP	SDF Gating Status	SDF Charging	EPS Bearer QoS
Browsing Degradation	UL:(192.168.10.24, *,*,*, HTTP) DL:(*, 192.168.10.24,*,*, HTTP)	N.A.	UL: 2 Mbps DL: 2 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 2Mbps MBR-DL: 2Mbps
Google Degradation	UL:(192.168.10.24, *,*,*, Google) DL:(*, 192.168.10.24,*,*, Google)	N.A.	UL: 2 Mbps DL: 2 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 2Mbps MBR-DL: 2Mbps
YouTube Degradation	UL:(192.168.10.24, *,*,*, YouTube) DL:(*, 192.168.10.24,*,*, YouTube)	N.A.	UL: Unlimited DL: Unlimited	QCI = 8 ARP = 8	Closed (not permitted)	Online	QCI= 8 ARP = 8 MBR-UL: Unlimited MBR-DL: Unlimited

Table 5.3. Policies structure - Low Consumption group.

It is worth mentioning that the asterisk means that the SDF template considers any port or IP address value for that space of the 5 tuple. Besides the UL and DL abbreviations represent the Upload and Download links bit rates respectively.

Policy Rule Name	SDF Template	SDF GBR	SDF MBR	SDF QCI/ARP	SDF Gating Status	SDF Charging	EPS Bearer QoS
Browsing Degradation	UL:(192.168.10.15, *,*,*, HTTP) DL:(*,192.168.10.15,*, *, HTTP)	N.A.	UL: 2 Mbps DL: 2 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 2Mbps MBR-DL: 2Mbps
Facebook Degradation	UL:(192.168.10.15, *,*,*, Facebook) DL:(*,192.168.10.15,*, *,Facebook)	N.A.	UL: 3 Mbps DL: 3 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 3Mbps MBR-DL: 3Mbps
Gmail Degradation	UL:(192.168.10.15, *,*,*, Gmail) DL:(*,192.168.10.15,*, *, Gmail)	N.A.	UL: 1 Mbps DL: 1 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 1Mbps MBR-DL: 1Mbps
Google Degradation	UL:(192.168.10.15, *,*,*, Google) DL:(*,192.168.10.15,*, *, Google)	N.A.	UL: 2 Mbps DL: 2 Mbps	QCI = 9 ARP = 9	Open (permit)	Online	QCI= 9 ARP = 9 MBR-UL: 2Mbps MBR-DL: 2Mbps
YouTube Degradation	UL:(192.168.10.15, *,*,*, YouTube) DL:(*,192.168.10.15,*, *, YouTube)	N.A.	UL: 5 Mbps DL: 5 Mbps	QCI = 8 ARP = 8	Open (permit)	Online	QCI= 8 ARP = 8 MBR-UL: 5 Mbps MBR-DL: 5 Mbps
Whatsapp Degradation	UL:(192.168.10.15, *,*,*, Whatsapp) DL:(*,192.168.10.15,*, *, Whatsapp)	N.A.	UL: Unlimited DL: Unlimited	QCI = 9 ARP = 9	Closed (not permitted)	Online	QCI= 8 ARP = 8 MBR-UL: Unlimited MBR-DL: Unlimited

Table 5.4. Policies structure - Medium Consumption group.

The policies are defined for two different users, taken from the dataset, each belonging to the low and medium consumption groups.

With this it can be concluded that a set of parameters that must be considered in order to propose QoS policies have been identified and that a set of personalized service degradation policies considering the consumption trends of OTT applications of users has been proposed, following the PCC architecture presented in the 3GPP technical specification 23.203.

Summary

This chapter, divided in two sections, first presented a detailed description of the PCC architecture as described in the 3GPP technical specification 23.203, focusing on the components closely related to the implementation of QoS policies inside a LTE network: Policy and Charging Rules Function (PCRF), Policy and Charging Enforcement Function (PCEF) and the PCC rules divided in their two types predefined and dynamic, and highlighting the different parameters that must be considered in order to propose QoS policies inside a LTE network.

In the second section a set of personalized service degradation policies is presented having in mind the behavior observed from the clustering analysis illustrated at the end of chapter 3. Considering that most of the parameters of a QoS policy (PCC rule) are standardized through the QoS Class Identifier (QCI), it is concluded that the service degradation policies can be managed in two ways: degrade the service performance by setting a specific value to the Maximum Bit Rate (MBR) of the Service Data Flows (SDF), where each bit rate is proposed having in mind that the maximum average bit rate of Colombian LTE networks is 19.07 Mbps, or block the service flows by setting the gating control parameter to closed. With this two possibilities a set of dynamic PCC rules are presented, having in mind the consumption behavior of each group (high, medium and low consumption).

Chapter 6

Conclusions and future works

This chapter presents the conclusions obtained from the development of this research project along with some possible future works, proposed to obtain further results on this area of investigation.

6.1. Conclusions

In this section the main conclusions obtained from the development of this research project are presented as follows:

- After an extensive investigation and analysis of the related works of this research project, it can be concluded that although there are multiple analysis and approaches of data caps, all of the works that were found, aim at creating ways that avoid having to implement this mechanism on the user and do not consider a personalization considering their consumption behavior.

- By implementing a set of different software tools, a dataset of IP flows, divided on three versions, holding information related to the consumption behavior of

OTT applications done by different users was created by leveraging the network inside the campus of Universidad Del Cauca.

- After the preprocessing stage of the dataset presented on chapter 3, it can be concluded that the flow statistics obtained from a traditional network monitoring are not enough information to identify specific applications being consumed on the different IP flows.

- A set of attributes, taken from the flow statistics of a traditional network monitoring, were defined in order to characterize the consumption behavior of a user related to OTT applications.

- After the clustering analysis performed on the dataset it is possible to conclude that, inside the campus of Universidad Del Cauca, the users consume the same applications, however vary on the intensity of their consumption.

- From the classification modeling illustrated in chapter 4, it is possible to conclude that the support vector machines are the best suited algorithm for the classification of new users among the three identified groups (high, medium and low consumption), however by observing the obtained precision and recall values seven of the eight algorithms can be considered as adequate options to create a classification model.

- After analyzing the PCC architecture on chapter 5, a set of attributes that must be considered in order to propose QoS policies are presented highlighting the fact that there are two possibilities that can be implemented in case of service degradation: The degradation of the service by setting limits to the Maximum Bit Rate or blocking the service by closing the gating control to the specific flow.

- A set of personalized service degradation policies having in mind the consumption behavior presented by the users were proposed for each of the three groups (high, medium and low consumption) defined after the clustering process.

6.2. Future works

Considering the investigation area of this research project, the following future works are proposed:

- Develop a framework that facilitates the gathering of data related with the consumption of OTT applications done by users of a mobile network.

- Create a dataset with a larger number of users and larger quantity of information of different OTT applications (e.g., a month).

- Develop and integrate a knowledge plane that enables the implementation of Artificial Intelligence techniques inside an Internet network.

- Develop a proposal of charging policies that are suited for a user in a current state of service degradation.

- Develop a framework that enable the integration of the software tools implemented within this research project.

- Develop an application that leverages the classification models of users based on their OTT consumption behavior in order to facilitate the classification of new instances.

References

[1] Wesley Clover, “Over-The-Top (OTT) a dramatic makeover of global communications.” 2014.

[2] Wedge Green and Barbara Lancaster, “Over The Top Services,” p. 9, 2006.

[3] K. T. Kearney and F. Torelli, “The SLA Model,” in *Service Level Agreements for Cloud Computing*, Springer, New York, NY, 2011, pp. 43–67.

[4] M. Chetty, R. Banks, A. J. Brush, J. Donner, and R. Grinter, “You’Re Capped: Understanding the Effects of Bandwidth Caps on Broadband Use in the Home,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 3021–3030.

[5] M. Chetty, H. Kim, S. Sundaresan, S. Burnett, N. Feamster, and W. K. Edwards, “uCap: An Internet Data Management Tool For The Home,” 2015, pp. 3093–3102.

[6] U. Mahola and L. Erasmus, “Emerging revenue model structure for mobile industry: The case for traditional and OTT service providers in Sub-Sahara,” in *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, 2015, pp. 1485–1494.

[7] J. K. MacKie-Mason and H. R. Varian, “Some FAQs about usage-based pricing,” *Comput. Netw. ISDN Syst.*, vol. 28, no. 1, pp. 257–265, Dec. 1995.

[8] M. Lasar, “It could be worse: data caps around the world,” *Ars Technica*, 04-Apr-2011. [Online]. Available: https://arstechnica.com/tech-policy/news/2011/04/how-internet-users-are-disciplined-around-the-world.ars. [Accessed: 03-Apr-2018].

[9] J. Wortham, “As Mobile Networks Speed Up, Data Gets Capped,” *The New York Times*, 14-Aug-2011.

[10] Jan Krämer, “Data Caps and Two-Sided Pricing: Evaluating Managed Service Business Models.” Proceedings of the European Conference on Information Systems (ECIS), 2014.

[11] *ETSI TR 102 157 - Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia*. 2003.

[12] I. O. for Standardization, *ISO 8402: 1994: Quality Management and Quality Assurance - Vocabulary*. International Organization for Standardization, 1994.

[13] “Quality of Service Regulation Manual.” [Online]. Available: https://www.itu.int/pub/D-PREF-BB.QOS_REG01-2017. [Accessed: 02-Mar-2018].

[14] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller, and K. Hanssgen, “A Survey of Payload-Based Traffic Classification Approaches,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 2, pp. 1135–1156, Second 2014.

[15] J. V. Gomes, P. R. M. Inácio, M. Pereira, M. M. Freire, and P. P. Monteiro, “Detection and Classification of Peer-to-peer Traffic: A Survey,” *ACM Comput Surv*, vol. 45, no. 3, p. 30:1–30:40, Jul. 2013.

[16] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic Mapping Studies in Software Engineering,” in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Swinton, UK, UK, 2008, pp. 68–77.

[17] “Quality of Service (QoS) and Policy Management in Mobile Data Networks | Ixia.” [Online]. Available: https://www.ixiacom.com/resources/quality-service-qos-and-policy-management-mobile-data-networks. [Accessed: 01-Dec-2016].
[18] C. Joe-Wong, S. Ha, S. Sen, and M. Chiang, “Do Mobile Data Plans Affect Usage? Results from a Pricing Trial with ISP Customers,” in *Passive and Active Measurement*, J. Mirkovic and Y. Liu, Eds. Springer International Publishing, 2015, pp. 96–108.
[19] V. Agababov *et al.*, “Flywheel: Google’s Data Compression Proxy for the Mobile Web,” 2015.
[20] K. Kritikos *et al.*, “A Survey on Service Quality Description,” *ACM Comput Surv*, vol. 46, no. 1, p. 1:1–1:58, Jul. 2013.
[21] F. Wamser, T. Zinner, P. Tran-Gia, and J. Zhu, “Dynamic Bandwidth Allocation for Multiple Network Connections: Improving User QoE and Network Usage of YouTube in Mobile Broadband,” in *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, New York, NY, USA, 2014, pp. 57–62.
[22] K. Samdanis, F. G. Mir, D. Kutscher, and T. Taleb, “Service Boost: Towards on-demand QoS enhancements for OTT apps in LTE,” in *2013 21st IEEE International Conference on Network Protocols (ICNP)*, 2013, pp. 1–6.
[23] C. Barclay, “Is regulation the answer to the rise of over the top (OTT) services? An exploratory study of the Caribbean market,” in *2015 ITU Kaleidoscope: Trust in the Information Society (K-2015)*, 2015, pp. 1–8.
[24] V. K. Adhikari *et al.*, “Measurement Study of Netflix, Hulu, and a Tale of Three CDNs,” *IEEEACM Trans. Netw.*, vol. 23, no. 6, pp. 1984–1997, Dec. 2015.
[25] J. Zhu, R. Vannithamby, C. Rödbro, M. Chen, and S. V. Andersen, “Improving QoE for Skype video call in Mobile Broadband Network,” in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1938–1943.
[26] Y. H. Wang, A. J. C. Trappey, and T. h Chow, “Incorporating quality function deployment to e-discovery system exploring Voice-over-LTE service technology,” in *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2015, pp. 224–228.
[27] S. So, “Mobile instant messaging support for teaching and learning in higher education,” *Internet High. Educ.*, vol. 31, pp. 32–42, Oct. 2016.
[28] “(3) Classification of VoIP and non-VoIP traffic using machine learning approaches,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/309592737_Classification_of_VoIP_and_non-VoIP_traffic_using_machine_learning_approaches. [Accessed: 03-Apr-2018].
[29] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, “Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms,” in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 2451–2455.
[30] D. Arndt, *netmate-flowcalc: A package of tools for calculating flow statistics from network traffic*. 2017.
[31] R. Ghnemat and E. Jaser, “Classification of Mobile Customers Behavior and Usage Patterns using Self-Organizing Neural Networks,” *Int. J. Interact. Mob. Technol. IJIM*, vol. 9, no. 4, pp. 4–11, Sep. 2015.

[32] C.-L. Hung, C.-Y. Lin, and P.-C. Wu, “An Efficient GPU-Based Multiple Pattern Matching Algorithm for Packet Filtering,” *J. Signal Process. Syst.*, pp. 1–12, Apr. 2016.
[33] S. Jayashree and N. Shivashankarappa, “Deep packet inspection using ternary content addressable memory,” in *International Conference on Circuits, Communication, Control and Computing*, 2014, pp. 441–447.
[34] Y. Li, J. Yang, and N. Ansari, “Cellular smartphone traffic and user behavior analysis,” in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 1326–1331.
[35] Y. Jin *et al.*, “Characterizing Data Usage Patterns in a Large Cellular Network,” in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, New York, NY, USA, 2012, pp. 7–12.
[36] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng, “Characterizing User Behavior in Mobile Internet,” *IEEE Trans. Emerg. Top. Comput.*, vol. 3, no. 1, pp. 95–106, Mar. 2015.
[37] Wei Dai and S. Jordan, “Design and impact of data caps,” 2013, pp. 1650–1656.
[38] K. Poularakis, I. Pefkianakis, J. Chandrashekar, and L. Tassiulas, “Pricing The Last Mile: Data Capping For Residential Broadband,” in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, New York, NY, USA, 2014, pp. 295–306.
[39] L. Zheng, C. Joe-Wong, C. W. Tan, S. Ha, and M. Chiangs, “Secondary markets for mobile data: Feasibility and benefits of traded data plans,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 1580–1588.
[40] S. Sen, C. Joe-Wong, and S. Ha, “The economics of shared data plans,” presented at the 22nd Workshop on Information Technologies and Systems, WITS 2012, 2012.
[41] W. Dai and S. Jordan, “The Effect of Data Caps Upon ISP Service Tier Design and Users,” *ACM Trans Internet Technol*, vol. 15, no. 2, p. 8:1–8:28, Jun. 2015.
[42] W. Dai, J. Baek, and S. Jordan, “The impact of data caps on ISP competition,” in *The 2014 5th International Conference on Game Theory for Networks*, 2014, pp. 1–6.
[43] X. Wang, R. T. B. Ma, and Y. Xu, “The role of data cap in two-part pricing under market competition,” *IEEE Netw.*, vol. 30, no. 2, pp. 12–17, Mar. 2016.
[44] C. Rossi *et al.*, “3GOL: Power-boosting ADSL Using 3G Onloading,” in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, New York, NY, USA, 2013, pp. 187–198.
[45] T. Casey and G. M. Muntean, “An integrated data offloading approach for mobile users in urban environments,” in *2015 26th Irish Signals and Systems Conference (ISSC)*, 2015, pp. 1–6.
[46] D. Stohr, S. Wilk, and W. Effelsberg, “Monitoring of User Generated Video Broadcasting Services,” in *Proceedings of the First International Workshop on Internet-Scale Multimedia Management*, New York, NY, USA, 2014, pp. 39–42.
[47] S. Wilk, D. Schreiber, D. Stohr, and W. Effelsberg, “On the effectiveness of video prefetching relying on recommender systems for mobile devices,” in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, pp. 429–434.

[48] A. El Bouchti, A. Haqiq, and A. Benjelloun, “Performance analysis of admission control and degradation service mechanism for multiclass services in LTE system,” 2014, pp. 231–236.
[49] S. Wilk, J. Rückert, T. Thräm, C. Koch, W. Effelsberg, and D. Hausheer, “The potential of social-aware multimedia prefetching on mobile devices,” in *2015 International Conference and Workshops on Networked Systems (NetSys)*, 2015, pp. 1–5.
[50] Z. Wang, M. Zhou, and H. Mei, “Towards an Adaptive Service Degradation Approach for Handling Server Overload,” in *2012 19th Asia-Pacific Software Engineering Conference*, 2012, vol. 1, pp. 52–60.
[51] S. X. Chen and G. Y. Li, “A mathematical theory of compressed video buffering: Traffic regulation for end-to-end video network QoS,” *APSIPA Trans. Signal Inf. Process.*, vol. 4, 2015.
[52] Y. Nagai, T. Okamawari, and T. Fujii, “A Streaming Method for Efficient Bandwidth Utilization Using QoS Control Function of LTE,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1–5.
[53] V. Mann, A. Tripathi, and S. Ghosal, “cStream: Cloud based high performance video delivery network,” in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, 2016, pp. 1–8.
[54] F. Wamser, T. Zinner, L. Iffländer, and P. Tran-Gia, “Demonstrating the Prospects of Dynamic Application-aware Networking in a Home Environment,” in *Proceedings of the 2014 ACM Conference on SIGCOMM*, New York, NY, USA, 2014, pp. 149–150.
[55] Weijian Sun and Xiaowei Qin, “End-to-End Delay Analysis of WeChat Video Call Service in Live DC-HSPA+Network,” presented at the Sixth International Conference On Wireless Communications and Signal Processing (WCSP), 2014, pp. 1–5.
[56] J. Famaey, S. Latré, T. Wauters, and F. D. Turck, “End-to-End Resource Management for Federated Delivery of Multimedia Services,” *J. Netw. Syst. Manag.*, vol. 22, no. 3, pp. 396–433, Jul. 2014.
[57] C. J. Liu, “Enhanced IPv6 ping and traceroute,” in *2012 21st Annual Wireless and Optical Communications Conference (WOCC)*, 2012, pp. 51–58.
[58] Y. Gourhant, A. Gouta, and V. D. Philip, “Fair Usage and Capping for Providing Internet for All in Developing Countries,” in *e-Infrastructure and e-Services for Developing Countries*, R. Popescu-Zeletin, K. Jonas, I. A. Rai, R. Glitho, and A. Villafiorita, Eds. Springer Berlin Heidelberg, 2011, pp. 35–48.
[59] T. Kärkkäinen and J. Ott, “Flexible QoS Provisioning for SIP Telephony over DVB-RCS Satellite Net-works,” 2007.
[60] N. Bouten, S. Latré, J. Famaey, W. V. Leekwijck, and F. D. Turck, “In-Network Quality Optimization for Adaptive Video Streaming Services,” *IEEE Trans. Multimed.*, vol. 16, no. 8, pp. 2281–2293, Dec. 2014.
[61] W. Dai, J. W. Baek, and S. Jordan, “Modeling the impact of QoS pricing on ISP integrated services and OTT services,” in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 85–91.
[62] I. M. Stephanakis and I. P. Chochliouros, “Multimedia Content Distribution over Next-Generation Heterogeneous Networks Featuring a Service Architecture of

Sliced Resources," in *Artificial Intelligence Applications and Innovations*, Springer Berlin Heidelberg, 2012, pp. 300–310.
[63] F. Oueslati and J. C. Grégoire, "Network Quality Adaptive Video Transmission," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 469–476.
[64] Y. Han, M. Zhao, and W. Zhou, "Optimization of OTT small data services: Network capacity and cost analysis," in *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, 2014, pp. 1–6.
[65] M. Rahayu, S. Haryadi, and D. R. Aryanti, "Over the Top chat service key performance indicator," in *2015 1st International Conference on Wireless and Telematics (ICWT)*, 2015, pp. 1–5.
[66] V. Aggarwal, E. Halepovic, J. Pang, S. Venkataraman, and H. Yan, "Prometheus: Toward Quality-of-experience Estimation for Mobile Apps from Passive Network Measurements," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, New York, NY, USA, 2014, p. 18:1–18:6.
[67] D. Rivera, N. Kushik, C. Fuenzalida, A. Cavalli, and N. Yevtushenko, "QoE Evaluation Based on QoS and QoBiz Parameters Applied to an OTT Service," in *2015 IEEE International Conference on Web Services*, 2015, pp. 607–614.
[68] H. Feng-Hui, Z. Wen-An, and D. Yu, "QoE Issues of OTT Services over 5G Network," in *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications*, 2014, pp. 267–273.
[69] A. E. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-Based Traffic and Resource Management for Adaptive HTTP Video Delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 988–1001, Jun. 2015.
[70] M. Leszczuk, M. Hanusiak, M. C. Q. Farias, E. Wyckens, and G. Heston, "Recent developments in visual quality monitoring by key performance indicators," *Multimed. Tools Appl.*, vol. 75, no. 17, pp. 10745–10767, Sep. 2016.
[71] S. Haryadi and F. Niramaya, "Study of unfair competition between regulated and unregulated VoIP providers in the mixed of non and all-IP network era," in *2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2014, pp. 1–5.
[72] H. Nam, K. H. Kim, B. H. Kim, D. Calin, and H. Schulzrinne, "Towards dynamic QoS-aware over-the-top video streaming," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014, pp. 1–9.
[73] W. Li, P. Spachos, M. Chignell, A. Leon-Garcia, L. Zucherman, and J. Jiang, "Understanding the relationships between performance metrics and QoE for Over-The-Top video," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.
[74] J. Hart and R. Brown, "What LTE policy control features can operators execute to differentiate themselves from OTT players?," in *2013 17th International Conference on Intelligence in Next Generation Networks (ICIN)*, 2013, pp. 16–22.

[75] E. Bertin, N. Crespi, and M. L'Hostis, "A few myths about telco and OTT models," in *2011 15th International Conference on Intelligence in Next Generation Networks*, 2011, pp. 6–10.
[76] X. Qiao, S. Xue, J. Chen, and A. Fensel, "A Lightweight Convergent Personal Mobile Service Delivery Approach Based on Phone Book," *Int J Commun Syst*, vol. 28, no. 1, pp. 49–70, Jan. 2015.
[77] J. Kibilda, F. Malandrino, and L. A. DaSilva, "Incentives for infrastructure deployment by over-the-top service providers in a mobile network: A cooperative game theory model," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.
[78] A. Nandhiasa and S. Haryadi, "Indonesian regulation management recommendation for Over-the-Top services," in *2015 1st International Conference on Wireless and Telematics (ICWT)*, 2015, pp. 1–4.
[79] I. D. Lutilsky and M. Ivić, "Influence of OTT service providers on Croatian telecomunication market," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 711–715.
[80] M. El-Sayed, A. Mukhopadhyay, C. Urrutia-Valdés, and Z. J. Zhao, "Mobile Data Explosion: Monetizing the Opportunity Through Dynamic Policies and Qos Pipes," *Bell Lab Tech J*, vol. 16, no. 2, pp. 79–99, Sep. 2011.
[81] H. C. Lai, Y. C. Yu, Y. M. Tuan, and H. S. Kuo, "Multi-screen services adoption and use-diffusion: The BEST model perspective," in *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, 2014, pp. 783–787.
[82] D. Saucez, S. Secci, and C. Barakat, "On the incentives and incremental deployments of ICN technologies for OTT services," *IEEE Netw.*, vol. 28, no. 3, pp. 20–25, May 2014.
[83] M. Škrbić, N. Dervišević, J. Mušović, A. Hebibović, and L. Kasumagić, "OTT services in Bosnia and Herzegovina," in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, 2014, pp. 47–50.
[84] Ilsa Godlovitch, Bas Kotterink, Scott Marcus, Pieter Nooren, Jop Esmeijer, and Arnold Roosendaal, "Over The Top Players - Market Dynamics and Policy Challenges." 2015.
[85] G. Li, "Regulating Over-the-Top Services in Australia – From Universal Service Obligation Scheme to OTT Regulation," Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 2773397, 2015.
[86] W. Robitza and A. Raake, "(Re-)actions speak louder than words? A novel test method for tracking user behavior in web video services," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, pp. 1–6.
[87] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. D. Turck, "A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 131–138.

[88] W. Sun, X. Qin, S. Tang, and G. Wei, “A QoE anomaly detection and diagnosis framework for cellular network operators,” in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 450–455.
[89] D. Ghadiyaram, J. Pan, and A. C. Bovik, “A time-varying subjective quality model for mobile streaming videos with stalling events,” 2015, p. 959911.
[90] P. Spachos, W. Li, M. Chignell, A. Leon-Garcia, L. Zucherman, and J. Jiang, “Acceptability and Quality of Experience in over the top video,” in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1693–1698.
[91] D. R. Villagra and A. R. Cavalli, “Analysis and Influence of Economical Decisions on the Quality of Experience of OTT Services,” *IEEE Lat. Am. Trans.*, vol. 14, no. 6, pp. 2773–2776, Jun. 2016.
[92] S. A. Jensen, M. Jensen, and J. M. Gutierrez, “Caching over-the-top services, the Netflix case,” in *2015 International Conference on Computing, Networking and Communications (ICNC)*, 2015, pp. 1081–1086.
[93] S. Ramadona, S. Haryadi, and D. R. Aryanti, “Over the top call service key performance indicator,” in *2015 1st International Conference on Wireless and Telematics (ICWT)*, 2015, pp. 1–4.
[94] S. Ohzahata, Y. Hagiwara, M. Terada, and K. Kawashima, “A Traffic Identification Method and Evaluations for a Pure P2P Application,” in *Proceedings of the 6th International Conference on Passive and Active Network Measurement*, Berlin, Heidelberg, 2005, pp. 55–68.
[95] V. Ilayaraja and R. Venkatesan, “Traffic Differentiation and QoS Provisioning for IEEE 802.11e Wireless LAN,” *Int J Mob Netw Innov*, vol. 6, no. 2, pp. 114–120, Nov. 2015.
[96] Y. R. Qu, S. Zhou, and V. K. Prasanna, “Packet Classification on Multi-core Platforms,” in *Handbook on Data Centers*, S. U. Khan and A. Y. Zomaya, Eds. Springer New York, 2015, pp. 425–447.
[97] J. Seppänen and M. Varela, “QoE-driven network management for real-time over-the-top multimedia services,” in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, 2013, pp. 1621–1626.
[98] N. Williams, S. Zander, and G. Armitage, “A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification,” *SIGCOMM Comput Commun Rev*, vol. 36, no. 5, pp. 5–16, Oct. 2006.
[99] C.-L. Hung, P.-C. Wu, H.-H. Wang, and C.-Y. Lin, “Efficient Parallel Muti-pattern Matching Using GPGPU Acceleration for Packet Filtering,” *DeepDyve*, Aug. 2015.
[100] A. Molavi Kakhki *et al.*, “Identifying Traffic Differentiation in Mobile Networks,” in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, New York, NY, USA, 2015, pp. 239–251.
[101] L. Wu, Y. Li, C. Zhou, H. Jiang, and X. Qian, “Statistic analysis of data access behavior in the mobile Internet,” in *2013 IEEE/CIC International Conference on Communications in China (ICCC)*, 2013, pp. 89–93.
[102] “Wireshark · Frequently Asked Questions.” [Online]. Available: https://www.wireshark.org/faq.html#q1.1. [Accessed: 03-Apr-2018].

[103] "(3) NetMate-User and Developer Manual," *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/246926554_NetMate-User_and_Developer_Manual. [Accessed: 03-Apr-2018].
[104] "Flowmeter | Datasets | Research | Canadian Institute for Cybersecurity | UNB." [Online]. Available: http://www.unb.ca/cic/datasets/flowmeter.html. [Accessed: 30-Nov-2017].
[105] tcpdump, "Tcpdump/Libpcap public repository." [Online]. Available: http://www.tcpdump.org. [Accessed: 03-Apr-2018].
[106] *ntopng source code repository*. ntop, 2018.
[107] "ntopng," *ntop*, 04-Aug-2011. .
[108] jsrojas, *NtopngDataEditor: A java application that loads 3 csv files obtained from ntopng, CICFlowmeter, and nDPI. It compares the flows statistics obtained from pcap files with CICFlowmeter and ntopng and..* 2018.
[109] "RStudio," *RStudio*. .
[110] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed: 16-Aug-2017].
[111] "Faculty of Computer Science | Arash Habibi Lashkari | UNB." [Online]. Available: https://www.cs.unb.ca/people/alashkar. [Accessed: 30-Nov-2017].
[112] M. Handley, O. Bonaventure, C. Raiciu, and A. Ford, "TCP Extensions for Multipath Operation with Multiple Addresses." [Online]. Available: https://tools.ietf.org/html/rfc6824. [Accessed: 05-Dec-2017].
[113] A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA, 2017.
[114] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.
[115] "(3) Finding Groups in Data: An Introduction To Cluster Analysis," *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/220695963_Finding_Groups_in_Data_An_Introduction_To_Cluster_Analysis. [Accessed: 03-Apr-2018].
[116] "Dataset Unicauca - 2018 - Google Drive." [Online]. Available: https://drive.google.com/drive/folders/1FcnKUlSqRb4q5PkGfAGHz-g7bVKL8jmu?usp=sharing.
[117] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.
[118] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[119] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans Intell Syst Technol*, vol. 2, no. 3, p. 27:1–27:27, May 2011.
[120] "(3) Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/234786663_Fast_Training_of_Support_Vector_Machines_Using_Sequential_Minimal_Optimization. [Accessed: 03-Apr-2018].

[121] “Gridsearch.” [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV. [Accessed: 03-Apr-2018].
[122] B. Kamiński, M. Jakubczyk, and P. Szufel, “A framework for sensitivity analysis of decision trees,” *Cent. Eur. J. Oper. Res.*, vol. 26, no. 1, pp. 135–159, Mar. 2018.
[123] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[124] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, Aug. 1992.
[125] “(3) An Empirical Study of the Naïve Bayes Classifier,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier. [Accessed: 03-Apr-2018].
[126] “(3) The Optimality of Naive Bayes,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/221439320_The_Optimality_of_Naive_Bayes. [Accessed: 03-Apr-2018].
[127] T. K. Ho, “Random Decision Forests,” in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, Washington, DC, USA, 1995, p. 278–.
[128] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recogn Lett*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[129] V. Cherkassky and Y. Ma, “Practical selection of SVM parameters and noise estimation for SVM regression,” *Neural Netw.*, vol. 17, no. 1, pp. 113–126, Jan. 2004.
[130] “ETSI TS 23.203: Policy and charging control architecture,” *ITU*. [Online]. Available: http://www.itu.int/itu-t/workprog/wp_a5_out.aspx?isn=6084. [Accessed: 07-Dec-2017].
[131] “LTE QoS: SDF and EPS Bearer QoS,” *Network Manias*. [Online]. Available: https://www.netmanias.com/en/?m=view&id=techdocs&no=5908. [Accessed: 03-Apr-2018].
[132] “LTE Policy and Charging Control (PCC),” *Network Manias*. [Online]. Available: https://www.netmanias.com/en/?m=view&id=techdocs&no=6562. [Accessed: 03-Apr-2018].
[133] “The State of LTE - OpenSignal.” [Online]. Available: https://opensignal.com/reports/2017/06/state-of-lte. [Accessed: 03-Apr-2018].