Dynamic personalized service degradation based on users' consumption behavior

Juan Sebastián Rojas Meléndez

Tesis de Doctorado en Ingeniería Telemática

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

Álvaro Rendón Gallón
PhD. En Ingeniería de Telecomunicaciones

Universidad Del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Popayán, octubre de 2020

Juan Sebastián Rojas Meléndez

Dynamic personalized service degradation based on users' consumption behavior

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

Doctor en:
Ingeniería Telemática

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

Álvaro Rendón Gallón
PhD. En Ingeniería de Telecomunicaciones

Popayán
2020

Dedicated to my family, friends 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 and PhD. Alvaro Rendón Gallón for their guidance and advisement in the development of this project, MinCiencias for the support given through the PhD scholarship and PhD Adrian Pekar for his guidance and support during the research internship in Budapest, Hungary.

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 data plans to their users with a limited consumption (e.g., 2 GB, 5GB), service degradation is a mechanism 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. This happens when a user exceeds his/her established consumption limit; in order to save resources and ensure the correct performance of the network, limits are applied to the information transfer rate and then, the functionalities of Internet services are negatively affected. Besides, when this mechanism is implemented, the behavior and preferences that the user presents in the consumption of OTT applications are never considered.

Knowledge Defined Networking (KDN) is a paradigm that aims to support network operators by automatizing network-related processes through the support of Machine Learning (ML) algorithms [1], [2]. ML has been proved to be suitable in many application domains of network traffic management and monitoring, including security, fault detection, resource allocation, traffic classification, and user profiling [3]–[7]. The majority of the models used in the KDN paradigm are based on traditional batch learning. However, it presents disadvantages. Firstly, depending on the algorithm and the size of the dataset, the training phase might take a considerable amount of time and demand high computational resources. Secondly, when training with a small dataset with no representative number of samples, the algorithm might present a poor performance. Lastly, once an algorithm has been trained and implemented, it cannot acquire new knowledge from new samples. In consequence, if there is a change in the statistical properties of the data, a new model must be generated.

Having in mind these disadvantages, Incremental Learning (IL) is a suitable alternative to overcome them. IL is built on the premise of continuous learning and adaptation, enabling the autonomous incremental development of complex skills and knowledge. In ML context, it aims to smoothly update the prediction model to account for different tasks and data distributions while still being able to re-use and retain knowledge over time [8], [9].

Considering the previous background, and leveraging the strengths of IL, this research project aims to dynamically assign users to a group of personalized service degradation policies based on their OTT consumption behavior over time.

In order to accomplish this objective, a reference model that offers guidelines in both obtaining users' consumption behavior information and implementing IL algorithms to achieve a model capable of classifying users' consumption will be proposed. Furthermore, the network attributes that enable the characterization of users' OTT consumption behavior will be determined. A performance comparison between traditional ML algorithms and IL algorithms will be performed. Finally, a prototype capable of implementing an IL model for the dynamic assignation of users to a set of personalized service degradation policies will be obtained.

Some of the obtained results in the development of this research project are: a reference model that offers guidelines in the processes needed to obtain information related to the users' consumption behavior and an IL model capable of doing a dynamic classification of such behaviors; 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 an IL classifier that enables the dynamic 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:

- The proposed and implemented reference model is a comprehensive guideline for the analysis and classification of users' OTT consumption behavior in Internet networks.

- An adequate IL model for the dynamic classification of users' consumption behavior was obtained from the comparison of traditional ML and IL algorithms.
- A personalized set of service degradation policies was designated for each group of users (high, medium and low consumption) based on the study performed on the users' OTT consumption behavior and on the QoS parameters involved in the operation of a mobile network.

As future works, it is proposed to develop a framework that facilitates the integration of all the processes proposed on the reference model. 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, Incremental Learning, 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 planes de datos a sus usuarios con un consumo limitado (por ejemplo 2 GB, 5 GB), la degradación del servicio es un mecanismo implementado de forma generalizada para aplicar límites a la cantidad de información que los usuarios pueden transferir durante un período de tiempo. Esto sucede cuando un usuario excede su límite de consumo establecido, y con el objetivo de ahorrar recursos y garantizar un correcto funcionamiento de la red, los operadores aplican límites a la velocidad de transferencia de información afectando de manera negativa la funcionalidad de los servicios de Internet. Además, cuando se implementa este mecanismo, el comportamiento y las preferencias que el usuario presenta en el consumo de aplicaciones OTT nunca son considerados.

Knowledge Defined Networking (KDN) es un paradigma que tiene como objetivo apoyar a los operadores de red mediante la automatización de procesos relacionados con la red a través de la implementación de algoritmos de *Machine Learning* (ML) [1], [2]. Se ha demostrado que el ML es una valiosa herramienta en muchas áreas de aplicación de la gestión y supervisión del tráfico en redes, entre las cuales encontramos: seguridad, detección de fallas, asignación de recursos, clasificación del tráfico y el perfilamiento de usuarios [3]-[7]. La mayoría de los modelos utilizados en el paradigma de KDN se basan en el ML tradicional (batch learning). Sin embargo, esto presenta desventajas. En primer lugar, según el algoritmo y el tamaño del conjunto de datos, la fase de entrenamiento puede tomar una cantidad considerable de tiempo y demandar una gran cantidad de recursos computacionales. En segundo lugar, cuando se entrena con un conjunto de datos sin un número representativo de instancias, el algoritmo puede presentar un bajo rendimiento. Por último, una vez que un algoritmo

ha sido entrenado e implementado, dicho modelo no puede adquirir nuevo conocimiento a partir de nuevas instancias. Por lo tanto, si hay un cambio en las propiedades estadísticas de los datos, es necesario generar un nuevo modelo.

Teniendo en cuenta estas desventajas, el Incremental Learning (IL) se presenta como una alternativa adecuada para superarlas. El IL está basado en la premisa de la adaptación y el aprendizaje continuo, permitiendo el desarrollo autónomo e incremental de habilidades y conocimientos complejos en el modelo. En el contexto de ML, el objetivo del IL es la actualización continua del modelo con la finalidad de tener en cuenta diferentes tareas y distribuciones de datos mientras se reutiliza y retiene el conocimiento adquirido a lo largo del tiempo [8], [9].

Considerando las fortalezas del IL, este proyecto de investigación tiene como objetivo la asignación dinámica de usuarios de una red a un grupo de políticas de degradación del servicio personalizadas basadas en su comportamiento de consumo de aplicaciones OTT a lo largo del tiempo.

Para lograr este objetivo, se propondrá un modelo de referencia que ofrezca pautas tanto para la obtención de información sobre el comportamiento de consumo de los usuarios como en la implementación de algoritmos IL para conseguir un modelo capaz de clasificar el consumo de los usuarios. Además, se determinarán los atributos de red que permiten la caracterización del comportamiento de consumo OTT de los usuarios. Se realizará una comparación del rendimiento entre algoritmos tradicionales del ML y algoritmos del IL. Finalmente, se obtendrá un prototipo capaz de implementar un modelo IL para la asignación dinámica de usuarios a un conjunto de políticas de degradación del servicio personalizadas.

Algunos de los resultados obtenidos en el desarrollo de este proyecto de investigación son: un modelo de referencia que presenta pautas sobre los procesos necesarios para obtener información relacionada con el comportamiento de consumo de los usuarios y un modelo de IL capaz de hacer una clasificación dinámica de tales comportamientos; un grupo de conjuntos de datos que permiten la clasificación de los usuarios en función 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 de IL que permite la identificación dinámica de las tendencias de consumo de las aplicaciones OTT por parte de los usuarios y la propuesta de políticas personalizadas de degradación de

servicio considerando las tendencias de consumo de las aplicaciones OTT de los usuarios.

De estos resultados es posible concluir lo siguiente:

- El modelo de referencia propuesto e implementado es una guía integral para el análisis y clasificación del comportamiento de consumo OTT de los usuarios en las redes de Internet.
- Se obtuvo un modelo de IL adecuado para la clasificación dinámica del comportamiento de consumo de los usuarios a partir de la comparación de los algoritmos del ML tradicional y del IL.
- Se designó un conjunto personalizado de políticas de degradación del servicio para cada grupo de usuarios (consumo alto, medio y bajo) basado en el análisis realizado sobre el comportamiento de consumo OTT de los usuarios y los parámetros de QoS involucrados en la operación de una red móvil.

Palabras Clave: Aplicaciones OTT, Degradación de servicio, Aprendizaje automático, Clasificación, Aprendizaje Incremental, 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 Internet Service Providers (ISP) where each company had their own network infrastructure and offered a unique set of services is facing a new challenge: the Over The Top (OTT) services.

OTT services is the expression used to refer to services carried over Internet networks, delivering added value functionalities to customers, without any ISP being involved in the planning, selling, provisioning, or servicing phases of the service and of course without any traditional telecommunications booking revenue obtained from them. Currently, the constant establishment 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, WhatsApp among many others, have emerged to tackle the new needs in communications and functionalities that customers demand [10], [11].

Due to this change the ISP 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 and delivers 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 between ISPs and OTT service providers aiming at increasing the benefits and economic revenue obtained from the high consumption rates that customers present on OTT services. Such cooperation involves 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) [12]. 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 [13], [14].

Such mechanisms are most commonly used in mobile networks where the data plans offered by traditional operators establish consumption limits (e.g., 2GB, 5GB). 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. 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, laptops, tablets and the continuous development of Internet networks have resulted into users 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 OTT providers, the business landscape has changed massively [15].

As mentioned before, OTT applications are known by their large consumption of network resources 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, service degradation is applied in a generalized way. This means that once it 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 and it breaches the SLA that the ISP has been able to establish with certain OTT applications.

Knowledge Defined Networking (KDN) is a paradigm that aims to support network operators by automatizing network-related processes through the support of Machine Learning (ML) algorithms [1], [2]. ML has been proved to be suitable in many application domains of network traffic management and monitoring, including security, fault detection, resource allocation, traffic classification, and user profiling [3]–[7]. The majority of the models used in the KDN paradigm are based on traditional batch learning. However, it presents disadvantages. Firstly, depending on the algorithm and the size of the dataset, the training phase might take a considerable amount of time and demand high computational resources. Secondly, when training with a small dataset with no representative number of samples, the algorithm might present a poor performance. Lastly, once an algorithm has been trained and implemented, it cannot acquire new knowledge from new samples. In consequence, if there is a change in the statistical properties of the data, a new model must be generated.

In front of these disadvantages, Incremental Learning (IL) is a suitable alternative to overcome them. IL is built on the premise of continuous learning and adaptation, enabling the autonomous incremental development of complex skills and knowledge. In ML context, it aims to smoothly update the prediction model to account for different tasks and data distributions while still being able to re-use and retain knowledge over time [8], [9].

With this in mind, a proposal of personalizing the service degradation policies applied to users has been considered by leveraging the benefits of traditional ML algorithms applied to the characterization of user consumption behavior [6]. Through this approach, a set of personalized service degradation policies can be considered for different types of users based on their consumption behavior. However, since this approach is built through traditional ML algorithms, the implemented model is not capable of considering the swift changes of the Internet and, as a consequence, the changes that a user can present in their consumption behavior over time. Therefore, there is still a need for dynamically adjusting the proposed personalized service degradation policies in order to maintain their usefulness over time.
With this in mind and considering the scenario previously described this research project aims at answering the following research question:

Is it possible to dynamically assign users to a group of personalized service degradation policies based on their OTT consumption behavior over time?

The following statement is proposed as a hypothesis for the research question:

The application of an incremental learning approach should lead to a dynamic assignation of users to a group of personalized service degradation policies

considering the changes presented on their OTT consumption behavior over time.

1.4. Objectives

Considering the previous motivation, the research question, and the strengths of IL this research project aims to dynamically assign users to a group of personalized service degradation policies based on their OTT consumption behavior over time.

In order to accomplish this objective, a reference model that offers guidelines in both obtaining users' consumption behavior information and implementing IL algorithms to achieve a model capable of classifying users' consumption will be proposed. Furthermore, the network attributes that enable the characterization of users' OTT consumption behavior will be determined. A performance comparison between traditional ML algorithms and IL algorithms will be performed. Finally, a prototype capable of implementing an IL model for the dynamic assignation of users to a set of personalized service degradation policies will be obtained.

1.5. Contributions

It is important to mention that this doctoral thesis is preceded by a master thesis titled "Personalized Service Degradation on OTT Applications" [16] which contributed the essential elements that enable the establishment of the current research project. Specifically, the contributions of the master thesis were:

- A set of two datasets, currently published on Kaggle [17], [18], divided in two versions: the first one holding IP flows labeled with the OTT application that is being consumed and the second version holding the OTT consumption behavior of 1581 users studied on the campus of Universidad Del Cauca.
- The combination of a set of software tools that enable the capture, analysis and labeling process of IP flows captured inside an Internet network.
- A comparative study of traditional supervised learning algorithms applied to the classification of users' OTT consumption behavior.

- A set of personalized service degradation policies, applied to three user consumption profiles (high, medium and low consumption), based on the Policy Charging and Control Architecture proposed by the 3GPP [19].

With this in mind the contributions of this doctoral thesis proposal are:

- **A reference model describing the procedures and actors** needed to obtain datasets and the IL model for network operators' decision-making process support.
- **A set of datasets that characterize the OTT consumption behavior of users on a time frame**, having as data sources the traffic gathered at Universidad Del Cauca network.
- **A comparative study of IL algorithms** applied to a set of datasets related to the characterization of the OTT consumption behavior of users inside a network.
- **A comparative study between incremental learning models and traditional supervised learning models** applied to a set of datasets related to the characterization of the OTT consumption behavior of users inside a network.
- **Publishing papers:** Within the present research project the following papers were published:
    - A 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 was developed within the International Conference on Computational Science and its Applications (ICCSA 2018) in Melbourne, Australia from July 2nd to July 5th, 2018. The paper was included in the Springer Lecture Notes in Computer Science (LNCS) series [7].
    - A paper titled **"Consumption Behavior Analysis of Over The Top Services: Incremental Learning or Traditional Methods?"** was published on the IEEE Access journal on September 20, 2019 [20].
    - A paper titled **"Smart User Consumption Profiling: Incremental Learning-based OTT Service Degradation"** was accepted for publication on the IEEE Access Journal on November 9, 2020 and is currently under publication process.

1.6. Content

The structure of the present document is described below:

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), incremental learning, and traffic classification focusing on statistical classification and Deep Packet Inspection (DPI); subsequently this chapter presents a set of related works mainly focused on reference models for user consumption characterization. Furthermore, some related work related to the following topics are also highlighted: service degradation; Quality of Service (QoS); OTT services; Traffic classification related to IL, and Categorization of users in a mobile network.

Chapter 3: This chapter presents a detailed description of the reference model proposed for analyzing users' OTT consumption behavior and obtaining an IL capable of classifying such behavior to support the decision making process of network administrators around network resource management and service degradation policies. The actors, components and tasks are explained in detail aiming at proposing a conceptual model that can be replicable on other scenarios.

Chapter 4: This chapter presents the construction of the different datasets that were obtained after applying the reference model proposed on chapter 3 and were used for the performance comparison and experimental scenarios of IL and batch learning algorithms. All the details regarding techniques, technologies and results that were applied on each stage are presented here.

Chapter 5: This chapter presents the evaluation and performance comparison of all the ML algorithms that were considered and applied to the obtained datasets. First a comparison between traditional batch learning algorithms and IL algorithms is presented. Then a comparison between IL algorithms is presented obtaining the IL model that is exhibits best results when classifying users' OTT consumption behavior.

Chapter 6: This chapter presents the application of the decision making stage of the reference model. Specifically, it presents the proposal of personalized service degradation policies based on the analysis obtained from the users' OTT consumption behavior, the IL model and the policy and control architecture proposed for a 5G network.

Chapter 7: 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), Incremental Learning (IL), network flow monitoring and traffic classification focusing on statistical classification and Deep Packet Inspection (DPI); subsequently this chapter presents a set of related work mainly focused on the topic of user profiling and model proposals for the characterization of users' consumption behavior. Furthermore, some works related to the following topics are highlighted: 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 and specially in identifying how incremental learning has been implemented within network traffic classification; and 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), incremental learning, network flow monitoring 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 [21]. 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 [22]. 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 [23]. Since ISP argue that caps help provide more consistent service to all their users, this pricing model is likely to persist [13].

Currently there are three dominant volume-based tariff schemes at the market. The fair-flat tariff establishes 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 might end up with relatively high cost for their additional data consumption. That unexpected high cost makes this tariff unattractive from a customer perspective since it adds a pay-per-use element to the already uncertain and unpredictable demand for data consumption.

Finally, data caps are the dominant volume-based tariff scheme in mobile Internet access, but are becoming 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 [24]. Figure 2.1 illustrates the different volume based tariff schemes.

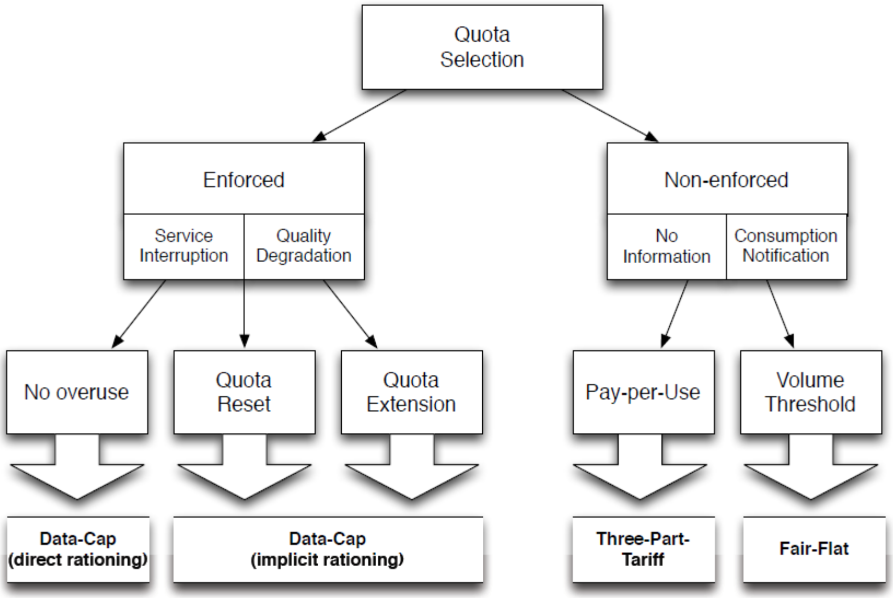


Figure 2.1. Volume based tariff schemes [24].

2.1.2. Quality of Service - QoS

Quality of Service (QoS) 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 [25], 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 [26], 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 QoS regulation manual [27] 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 [27].

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 and customer care.

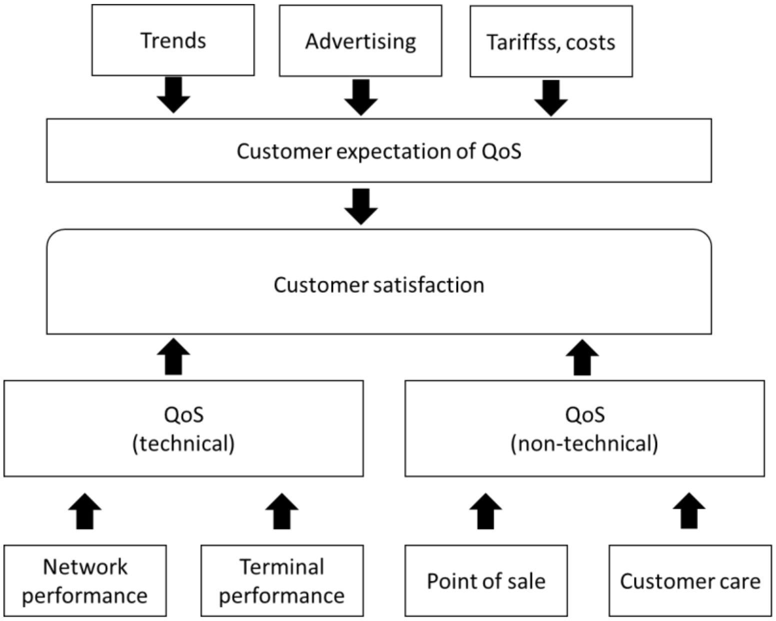


Figure 2.2. QoS technical and non-technical point of view [27].

2.1.3. Network flow monitoring

Network monitoring approaches have been proposed and developed throughout the years, each of them serving a different purpose. They can generally be classified into two categories: active and passive. Active approaches, such as the ones implemented in tools like Ping and Traceroute, inject traffic into a network to perform different types of measurements. Passive approaches observe existing traffic as it passes by a measurement point and therefore observe traffic generated by users. One passive monitoring approach is packet capture. This method generally provides most insight into the network traffic, as complete packets can be captured and further analyzed. However, in high-speed networks with line rates of up to 100 Gbps, packet capture requires expensive hardware and substantial infrastructure for storage and analysis. Another passive network monitoring approach that is more scalable for use in high-speed networks is flow exportation. In this setting, packets are aggregated into flows and exported for storage and analysis. A flow is defined in [28] as “a set of IP packets passing an observation point in the network during a certain time interval, such that all packets belonging to a particular flow have a set of common properties.” These common properties usually include packet header fields, such as source and destination IP addresses and port numbers, packet contents, and meta-information. Initial works on flow export date back to the nineties and became the basis for modern protocols, such as NetFlow and IP Flow Information eXport (IPFIX) [29].

In addition to their suitability for use in high-speed networks, flow export protocols and technologies provide several other advantages compared to regular packet capture. First, they are widely deployed, mainly due to their integration into high end packet forwarding devices, such as routers, switches and firewalls. As such, no additional capturing devices are needed, which makes flow monitoring less costly than regular packet capture. Second, flow export is well understood, since it is widely used for security analysis, capacity planning, accounting, and profiling, among others. It is also frequently used to comply with data retention laws. Third, significant data reduction can be achieved since packets are aggregated after they have been captured. Fourth, flow export is usually less privacy-sensitive than packet export, since traditionally only packet headers are considered. However, since researchers, vendors and standardization organizations are working on the inclusion of application information in flow data, the advantage of performing flow export in terms of privacy is fading [30].

The architecture of typical network flow monitoring setups consists of four stages, each of which are described as follows [30].

- **Packet Observation:** in this first stage, packets are captured and preprocessed from an Observation Point. Observation Points can be line cards or interfaces of packet forwarding devices.
- **Flow Metering & Export:** the second stage consists of both a metering and an exporting Process. Within the metering process, packets are aggregated into flows. After a flow is considered to be terminated, its flow record is exported and stored in a data structure (database, csv file, etc). Flow records may include both characteristic properties of a flow (e.g., IP addresses and port numbers) and statistical properties (e.g., packet sizes, inter-arrival times, among others).
- **Data Collection:** in the third stage, the main tasks are the reception, storage and preprocessing of flow data generated in the previous stage. Common preprocessing operations include aggregation, filtering, data compression, and summary generation.
- **Data Analysis:** This final stage usually consists of a semi-automatic analysis of the gathered data supported on data science frameworks and ML techniques. The implementation of common analysis functions depends on the research purpose. These functions include correlation and aggregation, traffic profiling, classification, characterization and anomaly detection, among many others.

2.1.4. Traffic Classification

Internet traffic classification has been a 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 [31]. Among them the following can be found: port number matching; Deep Packet Inspection (DPI); and classification in the dark. This approaches 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 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 that are usually allowed by firewalls. Thereby, nowadays port numbers as a classification mechanism is considered obsolete [32].

Traffic Classification based on DPI

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. Due to its effectiveness, classification systems based on DPI are especially significant for accounting solutions and charging mechanisms where accuracy is crucial. However, the main drawback of DPI techniques is their inability to be used when the traffic is encrypted. Since, in these cases, the contents of the packets is inaccessible, 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 [32].

Traffic Classification in the Dark

The inspection of IP packets content, 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. However, 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 ML algorithms [32]. Table 2.1 illustrates a brief comparison of the different traffic classification approaches.

2.1.5. Incremental Learning

Academics and practitioners alike believe that Incremental Learning (IL) is a fundamental step towards artificial intelligence. IL is the ability of a model to learn continually from a stream of data. In ML context, it aims to smoothly update the prediction model to account for different tasks and data distributions while still being able to re-use and retain previous knowledge over time. The idea is to mimic humans' ability to continually acquire, fine-tune, and transfer knowledge and skills throughout their lifespan [8]. Having in mind that new data is being constantly generated and changing, the need for solutions capable of adapting to this changes is becoming more relevant. Therefore, an IL algorithm can be defined as one that meets the following criteria [33]:

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. - It cannot identify specific applications.

Table 2.1. Comparison of the traffic classification approaches [32].

- **Adaptiveness:** having in mind that most real-life situations and phenomena cannot be limited to a specific number of unchanging variables that can be processed deterministically, it is imperative to keep adapting. The IL algorithm must guarantee adaptation capabilities through a fast, efficient, and flexible learning process. Maintaining knowledge acquisition with every new data input and learning new class labels or features when needed.
- **Scalability:** the input data must be processed only once to ensure that the algorithm can scale in terms of intelligence through the processing of more and more data while maintaining computational memory fixed or on sustainable terms. Storing large amounts of data with high-dimensionality would be impossible to maintain while aiming to process for a long time scale. This way, the IL algorithm is more similar to an actual brain that filters partial data and retains only the most essential information.
- **Efficiency:** as the process is continuous and the amount of data keeps increasing, the algorithm should not start from scratch every time. This way, there is no need for large amounts of computational resources to handle data.

A more in depth explanation and comparison between IL and traditional ML will be presented on chapter 5.

2.2. Related work

This section presents an analysis of the most important related work 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. The current state of knowledge will be presented after implementing two systematic mappings of academic documents, based on the methodology proposed in [34], to provide an overview of the research area and determine the amount and type of works. First, a systematic mapping related to the area of user profiling will be presented. Then, a systematic mapping related to service degradation and incremental learning will be explained. Finally, the identified shortcomings obtained from all the related work is exposed.

2.2.1. User profiling related works

Having in mind that one of the main contributions of this thesis is the reference model proposed to support network operators in the characterization of the OTT consumption behavior of users, the main objective of the first systematic mapping is to determine if there are similar or identical proposals related to the study of the OTT consumption behavior of users. Therefore, the systematic mapping was established to analyze recent research proposals between 2015 and 2020, using 9 keywords, and 11 search queries applied to 4 scientific databases (Scopus, ScienceDirect, IEEE Xplore, Google Scholar).

From the results obtained with the best search queries a total of 521 papers were obtained. Then, by analyzing the title and abstract of the papers a total of 108 papers were left in the mapping. From these papers, 80 unique papers were identified and 28 repeated papers were obtained due to applying the same search queries in the different scientific databases. Subsequently, by classifying the papers in the different research types defined in [34] and also by defining 6 different research contexts that consider the aim of the research proposals, Figure 2.3 illustrates the map of related work obtained after applying the systematic mapping methodology.

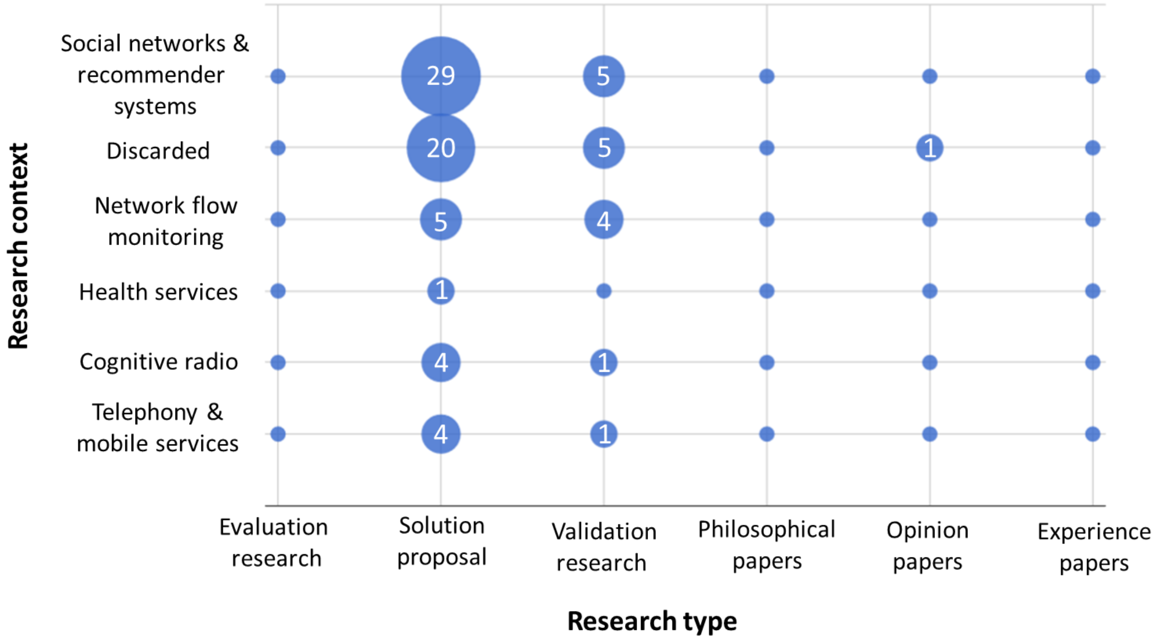


Figure 2.3. Systematic map – User profiling.

As mentioned before, based on the focus and aims of the works, a total of 6 research contexts were defined which are briefly described as follows: *Telephony & mobile services* hold papers that aim at creating models for user profiling based on their behavior when using traditional voice calls, applications, load on network resources and their mobility within mobile networks [35]–[39]. *Cognitive radio* holds papers that focus on studying the users' behavior in order to efficiently manage the limited resources of the spectrum when assigning frequencies for radio transmission [40]–[44]. *Health services* holds a paper that proposes a data driven approach to characterize users in online health services [45]. *Network flow monitoring* holds papers that mainly focus on security issues inside Internet networks. The aim is focused on finding ways for studying users' behavior in order to verify their identities and identify anomaly situations in their activities and mitigate security attacks on the network [46]–[54]. *Discarded* holds the papers that were out of the scope of the objectives of the systematic mapping. The works within this category includes tourism, transport and e-commerce related papers. *Social networks & recommender systems* hold the majority of papers and their focus is aimed at characterizing the way users behave in social networks and obtain more precise recommender systems in different contexts [55]–[88]. From these classifications the following conclusions can be stated:

- The most representative research context when searching for user profiling related work is social networks and recommender systems. The research proposed on this context are usually applied to areas like: security, marketing, content recommendation, fake news identification, among others.

- The most common research type for user profiling is solution proposal i.e., papers where a solution for a problem is proposed. This solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation.
- Since the topic of user characterization can be applied to several contexts, most of the discarded works are from areas unrelated to telecommunications and Internet networks. These areas include: tourism, public transport and e-commerce.
- There are no works found through this systematic mapping that propose a similar or identical approach that aims at proposing a reference model for the characterization of users' OTT consumption behavior. In fact, the study of OTT applications is quite limited with the exception of social networks like Facebook and Twitter.

Even though their approach is not applied to the same objective, some works classified within *network flow monitoring* and *telephony & mobile services* can present similar and interesting approaches in the topic of user profiling. Those works are highlighted as follows: One of the objectives of network security is to control the use of shared resources among users. In this regard, knowing the actual identity of network users is quite valuable to the intermediate nodes. With this in mind, Vinupaul et.al, [46] tries to establish network flow analysis as a viable method of user identification in such cases. They propose a test for ML supervised learning models that uses flow features to identify users within a given set. Based on their analysis of flow features, the concept of flow-bundle-level features which can be derived from the packet-level and flow-level features is introduced to characterize and identify users. A total of 6 flow-bundle-level user related features is defined. Four different ML models are validated using a dataset of flow records that holds a total of 65 users. After the training and testing of the models the best accuracy is of 83% when identifying users presented by the random forest algorithm.

Considering the ever-increasing capabilities of mobile networks, the amount of devices connected to the Internet and the fact that traffic classification has a long tradition in networking around modeling a network workload for several performance evaluations, Hess et.al, [36] defines a profiling model that characterizes the user behavior as well as its temporal dynamics from two perspectives: the network load the users generate, and their mobility patterns. The study is based on a 3G dataset captured between December 2011 and January 2012 (40 days) provided by Orange Labs in France, with millions of users. The model is evaluated with two unsupervised clustering algorithms (XMeans and Expectation Maximization). The main conclusions obtained in the profiling around both mobility and network load are: the spatial locations during an hour show peak times having more mobility, with around 1.3 cells. During off-peak times the number of cells is around 1.1, and over the day around 80% of users visit 6 cells. Knowing these regularities in mobility helps to estimate predictability of where in the network the load will be high. Also, 22% of users do not revisit a site and around 30% revisit the same sites between 6-10 days. Finally, on holidays users show a higher number of upload/download rates.

In [37], Rajashekar et.al, study to what extent phone behaviors can be associated with a single user. To this end, the research explores the use of a multilayer perceptron configured to act as an autoencoder, prior to the application of a Self-Organizing Map (SOM). Such a configuration operates under a one-class learning constraint with the objective of providing a unique characterization of user behavior by using the publicly available LiveLab iPhone usage traces from Rice University. The study used the application, cell tower and website usage logs to generate user behavior profile and isolate a single user for a specific device based on the behavior. In [47] Bakhshandeh et.al, state that the study of traffic is a critical source of information for network management and forensics highlighting the fact that identifying the user based on their behavior and not by the IP address is a vital task since the dynamic assignation of addresses can easily avoid security mechanism. Therefore, the authors propose a method for efficiently identifying users of a network based on their behavior using the Netflow traffic (which does not contain payloads). Through this method they are able to extract a set of features from the network flows and use a random forest model to classify users. This model achieved a precision of 94% in the detection of users and the results show that this method can be effectively used by forensic scientists as they do not need to examine the whole traffic to identify and characterize the users.

All the works previously highlighted consider a user profiling approach for several purposes, however, none of them consider the proposal of a reference model that can be replicated on any network and obtain an overview of the users’ OTT consumption behavior. In fact, in most of the cases, the model proposed by other researches is a ML model capable of doing a specific task (classification or prediction) but it does not offer guidelines on how to perform the user profiling nor can be implemented on a context different from the one considered on the work.

In the next subsection another systematic mapping will be presented in order to obtain perspective on the other main topics related to this research project.

2.2.2. Service Degradation, OTT services, traffic classification and incremental learning related work

Moving on with the other topics that are relevant to this project, it was necessary to perform an additional literature review were the following topics were considered: Service degradation, with the objective of identifying how this resource control mechanism is managed in the networks by Internet Service Providers (ISP); OTT services, in order to know how this topic has been developed in the research field; Traffic classification, focusing on identifying which techniques are used for this process and specially in identifying how incremental learning has been implemented within the service degradation and network traffic classification scope. Furthermore, it was also important to look for dataset descriptions related to the consumption of OTT services. Figure 2.4 illustrates the systematic map with 39 papers that were obtained after applying 8 search queries for works that related service degradation in OTT services, traffic classification and incremental learning. Considering the areas of application of each work a total of 6 research contexts were defined as observed in the vertical axis.

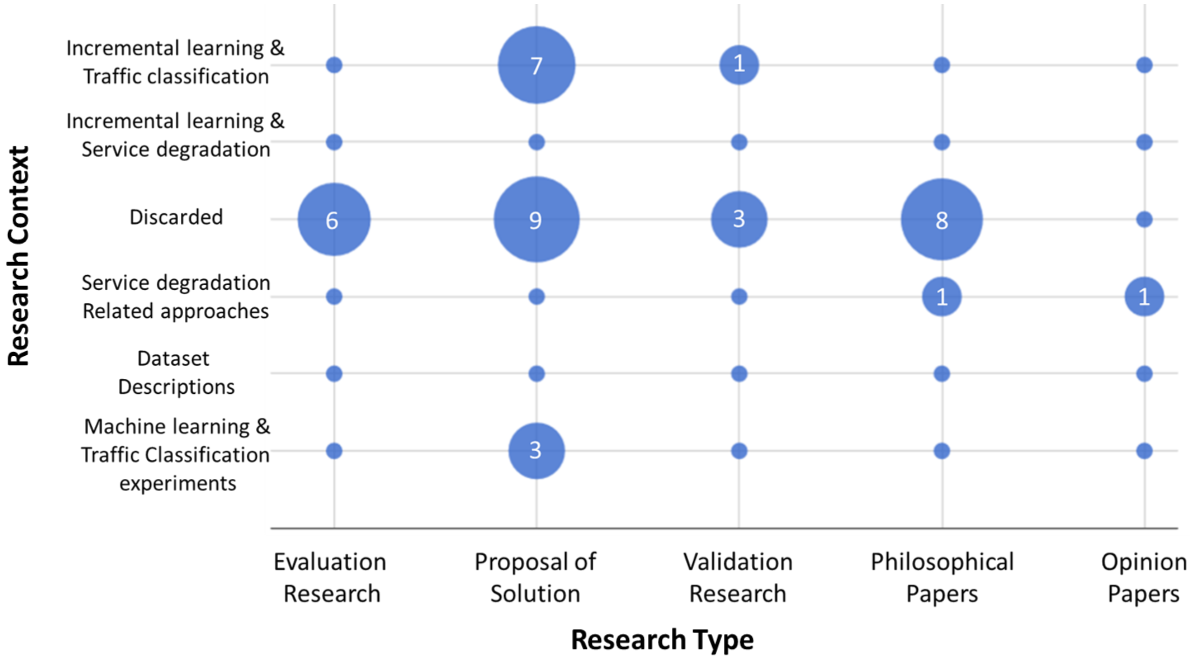


Figure 2.4. Second systematic map.

From the map the following annotations can be stated:

- There are 8 papers that apply incremental learning to network traffic classification without considering the study of user consumption behavior.
- There are no works that consider applying incremental learning in service degradation.
- Most of discarded papers are related to traffic mobility and object and people recognition.
- There are no papers that describe and share datasets related to the consumption of OTT services and user behavior.

Since no papers that involved service degradation, OTT services and incremental learning were found an additional literature review was performed considering these topics individually to obtain a clear insight on both topics. On the other hand, several research proposals involving traffic classification and IL were identified. With this in mind, the most relevant papers will be highlighted as follows.

Each thematic core had 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. *Resource control techniques*, 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 [13] 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 [89], 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 [24] 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 [90] 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., [91]. 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., [14] 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.

Proceeding with the thematic cores, we will continue with the most relevant works related with 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* (Voice over IP), 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 [10] 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 [92] 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 M*edia* 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., [93] 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 *VoIP* scope Zhu et.al., [94] 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 [95] 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 [96] 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.

Finally, proceeding with the most relevant works identified in the systematic mapping that related traffic classification and incremental learning, the following papers can be highlighted: In [97], G. Sun et.al, present a preliminary proposal of an incremental Support Vector Machine (SVM) method that is applied to address two issues of current SVM's: the inability to support continuous learning and the fact that this kind of algorithm has high requirements on both memory and CPU; experimental results show that the incremental Support Vector Machine method decreases the training time, while still sustains the high accuracy of traffic classification. In [98], considering that, in order to classify network traffic in today's dynamic environment, data stream mining algorithms have been introduced to overcome the shortcoming of conventional data mining algorithms, H. R. Loo et.al, presented an online classification method which is aimed for online network traffic classification, by applying an incremental k-means where the classification model can learn from unlabeled and labeled data becoming an incremental semi-supervised learning approach. In [99], having in mind that classification accuracy of supervised approaches is significantly affected if the size of the training set is small, and that a model built using a static training set will not be able to adapt to the non-static nature of Internet traffic, Divakaran et al. developed the concept of ‘‘self-learning’’ to deal with these two challenges; specifically, this paper designs and develops a new classifier called Self-Learning Intelligent Classifier (SLIC);SLIC starts with a small number of training instances, self-learns and rebuilds the classification model dynamically, with the aim of achieving high accuracy in classifying non-static traffic flows.

After analyzing the previous related works, it is important to mention that to the best of our knowledge, there have not been papers proposed by other authors that consider a similar approach to the one proposed on this project. In consequence, the next subsection will focus on the shortcomings identified after the review of all the papers that were found.

2.2.3. Shortcomings

The previous subsections illustrate the different trends and focus of the most relevant related works structured in thematic groups. From the works related to user profiling it is clear that none of them consider the proposal of a reference model that can be replicated on any network to obtain an overview of the users' OTT consumption behavior. From the service degradation related work 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 topic 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 and incremental learning indeed considered the inclusion of this new ML paradigm, however, it is only applied in the traditional traffic classification. Therefore, the application of IL to the study of users' consumption behavior and how this kind of models can be useful for network operators in an area such as the definition of service degradation policies is not considered.

Finally, Table 2.2 presents the different shortcomings identified in some of the highlighted works and subsequently a general summary of the identified shortcomings is presented.

- Most of the works related to user profiling focus on social networks, recommender systems and the identification of users to avoid security issues. Therefore, none of them propose a reference model that serves as a guideline to support network operators in the study of users' OTT consumption behavior.

- 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 consider the implementation of both traditional ML (supervised and unsupervised learning) and incremental learning in the context of traffic classification, none of those works has considered to perform an analysis in the user’s OTT consumption behavior nor the development of a dynamic classification model for the application of personalized service degradation policies.

- Considering that most of the works that are related to traffic classification reuse the KDD-Cup 99 dataset, the Cambridge dataset or the University of Brescia dataset, it is possible to conclude that during the construction and exploration of the state of the art an original dataset focused on the information about the consumption of OTT applications generated by users inside a network was not found.

User profiling related works		
Related work	**Contributions**	**Shortcomings**
[46]	- It proposes a test for ML supervised learning models that uses flow features to identify users within a given set of packet and flow features. - It presents the concept of flow-bundle-level features which can be derived from the packet and flow features to characterize and identify users.	- The main focus of the paper is the establishment of the identity of the user. Therefore, the consumption behavior is not considered. - Since the resulting model is the trained algorithm from a supervised learning approach, it does not offer a conceptual approach nor guidelines in how to characterize users' OTT consumption behavior.
[36]	- This work defines a profiling model that characterizes the user behavior as well as its temporal dynamics from two perspectives: the network load the users generate, and their mobility patterns. - The model is evaluated with two unsupervised clustering algorithms (XMeans and Expectation Maximization).	- Even though this work characterizes the user behavior in two perspectives, none of them consider the consumption of OTT applications. - The proposed model does not offer conceptual guidelines that support network operators in a decision making process when managing a network.
[47]	- This paper proposes a method for efficiently identifying users of a network based on their behavior using the Netflow traffic - A random forest model trained with the extracted set of features from the network flows which is implemented to classify and identify users.	- As in a previous case this work focuses on the importance of verifying the identity of the users instead of analyzing their consumption behavior. - The proposed model is the result of a ML train/test process. Therefore, it does not offer any conceptual base, reference model or guidelines about how to characterize the user.
Service degradation related works		
Related work	**Contributions**	**Shortcomings**
[13]	- Describes how monthly bandwidth caps affect households' Internet use in South Africa.	- Does not propose any considerations related to the service degradation.
[89]	- 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.
[24]	- 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.
[91]	- Presents a tool with the aim of being a proxy service for HTTP, extending the life time of users' data plans in mobile networks. - The tool is capable of reducing the size of the packages that are exchanged between servers and user equipment aiming at avoiding the need of users to acquire a new data plan.	- Aims at avoiding the service degradation, hence does not consider a course of action after the users exceed their consumption limit.

OTT Services related works		
Related work	**Contributions**	**Shortcomings**
[93]	**-** 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.
[92]	- 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 and incremental learning related works		
Related work	**Contributions**	**Shortcomings**
[97]	- Considers the implementation of incremental learning focused on Support Vector Machines (SVMs) in traditional traffic classification and evaluates its performance. - The training and evaluation is performed with the Cambridge dataset.	- It only focuses in the traditional traffic classification without considering a user consumption characterization nor a dynamic model applied to a personalization of service degradation. Furthermore, it implements the traditional Cambridge dataset without proposing a new one.
[98]	- This paper compares their incremental k means proposal against five different updatable algorithms used in MOA a tool developed by the Waikato university. - It implements the Cambridge dataset as well as a dataset proposed by the University of Brescia where the aimed classes are the type of applications (web, mail, P2P, Skype, among others).	- Although it implements incremental learning to create a dynamic model it is aimed at traditional traffic classification without considering further studies that characterize the user's consumption behavior for a dynamic application of personalized service degradation policies.
[99]	- This paper propose their own incremental learning approach to resolve the problems of static models generated by the traditional machine learning approach when applied in traffic classification.	- Their approach only focuses on the traditional traffic classification using types of applications as the objective class (HTTP, DNS, BitTorrent, among others). Hence, as in previous cases, a deeper study about the characterization of user's consumption behavior and a dynamic application of personalized service degradation policies is out of the scope of this work.

Table 2.2. Identified shortcomings.

Summary

This chapter presented the concepts related with this research project such as data caps or service degradation, Quality of Service (QoS), Traffic Classification, network flow monitoring and incremental learning. Furthermore, a study of the related works was performed dividing them into two analyses. First, a systematic mapping of the user profiling area with the aim of identifying some research works that proposed a reference model for characterizing users' behavior when consuming OTT applications. Second, a systematic mapping aiming to find works that related Service degradation, OTT services, Traffic Classification and incremental learning.

After this, the most important related works for each thematic core were highlighted and briefly explained in its contributions. Furthermore, all the works were separated on different categories having in mind the proposed contributions and the area of application of each work.

Finally, the different shortcomings that were identified are illustrated highlighting the following major conclusion:

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 supported by incremental learning models or the proposal of a reference model that provides guidelines and support to network operators in the study of users' OTT consumption behavior.

Chapter 3

Reference Model

This chapter presents a detailed description of the reference model that is defined to offer guidelines to researchers in the field of network flow monitoring, and network operators to support their decision making process when dealing with the definition and personalization of service degradation policies. This model defines a series of steps that can be replicated to study and characterize users' OTT consumption behavior and obtained an IL model capable of classifying each user based on his/her consumption behavior. First, the actors that are involved in the model are presented. Then the workflow and components are described for each step.

Figure 3.1 illustrates the proposed reference model with all its different actors, components and processes.

3.1. Actors

Since the reference model has different steps that have to be carried out, there are some roles and responsibilities that have to be considered in order to develop the needed activities. In consequence, this section presents the definition of four different actors that have to be considered when following the structure proposed in the current reference model.

3.1.1. Network users – end users

As it can be expected, the first actor that will be involved in the activities related to the reference model is the people that are connected to the network while using different devices. Since there are several definitions for network/end user, some of them will be provided as follows to offer some context. One of those definitions is the one proposed in the Dictionary of computer and Internet terms [100]: “an end user is the person ultimately intended to use a product, as opposed to people involved in developing or marketing it.”

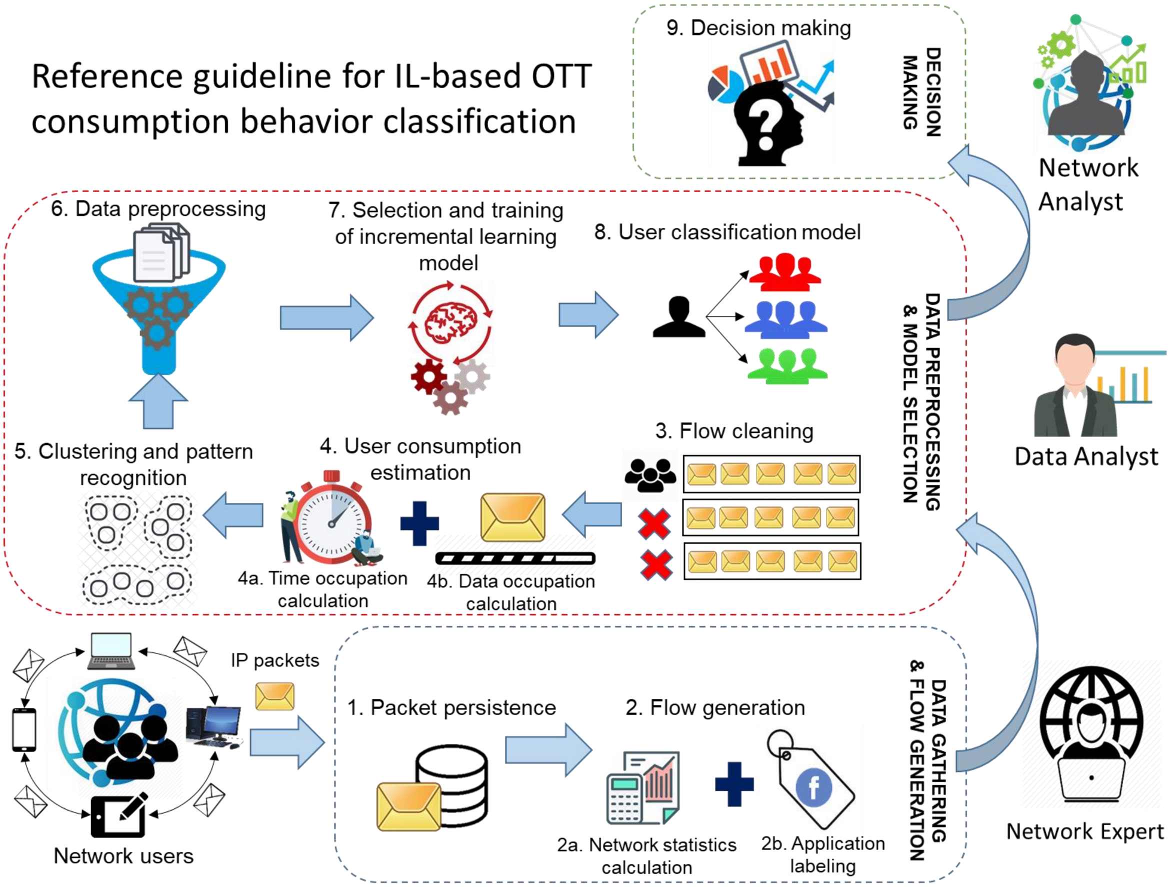

Reference guideline for IL-based OTT consumption behavior classification
9. Decision making
DECISION MAKING
Network Analyst
6. Data preprocessing
7. Selection and training of incremental learning model
8. User classification model
DATA PREPROCESSING & MODEL SELECTION
Data Analyst
5. Clustering and pattern recognition
4. User consumption estimation
3. Flow cleaning
4a. Time occupation calculation
4b. Data occupation calculation
IP packets
1. Packet persistence
2. Flow generation
DATA GATHERING & FLOW GENERATION
Network users
2a. Network statistics calculation
2b. Application labeling
Network Expert

Figure 3.1. Reference model for the characterization of users’ OTT consumption behavior.

Another definition is provided in [101], [102] : “A network user is a person who utilizes a computer or network service. Users of computer systems and software products generally lack the technical expertise required to fully understand how they work.”

With the previous definitions in mind, and to provide our own definition of network/end user, we can state that: a network user is the entity on which the reference model is focused on. This actor is represented by any person that has access to a device capable of connecting to the network (e.g., smartphone, laptop, or tablet). Their main activity is the generation of network traffic through the consumption of OTT applications or any other activity that can be performed through an Internet connection without requiring any technical or specific knowledge about how the OTT applications or the network works in the background.

3.1.2. Network expert – Network consultant

The second role that can be seen on the first level of the reference model (starting from the bottom of the figure) is the network expert/consultant. A formal definition of the skills and responsibilities of the current role is provided based on [103]:

“The network consultant is an experienced and educated professional who certifies network functionality and performance. They are responsible for designing, setting up and maintaining computer networks at either an organization or client location. Consultants meet with the network engineers to discuss networking requirements.”

A network consultant researches the network’s performance and security, and after careful analysis, suggest changes and equipment investments for a better functionality along with cost-benefit considerations. It is a broad title that includes a wide variety of work and responsibilities. In the telecommunications scope, a network expert/consultant might fulfill the skills of a network architect, system administrator, and security specialist, among many others.

In the case of the proposed reference model, the network expert/consultant represents the person or staff who holds all the knowledge about the technical activities related to the network architecture and devices. This actor is capable of executing activities related to the configuration of network devices, gathering and summarizing network information, and configuring environments for experiments within the network. Therefore, the main activity of this role in the context of the reference model is the configuration of network equipment in order to gather network data related to the end users’ OTT consumption behavior. This activity also includes storing and preprocessing of the gathered information.

3.1.3. Data analyst

The third actor is the data analyst. The tasks assigned to this role are illustrated on the second level of the reference model. Data analysis is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and it is used in different business and science domains [104].

Based on [105]: "A Data Analyst interprets data and turns it into information which can offer ways to improve a business, thus affecting business decisions. Data Analysts gather information from various sources and interpret patterns and trends – as such a Data Analyst job description should highlight the analytical nature of the role. Once data has been gathered and interpreted, the Data Analyst will report back what has been found in a comprehensive study to the relevant colleagues."

For the context of the reference model, the data analyst is in charge of developing all the activities related to data analysis and ML models. Those activities include data cleaning and preprocessing, data transformation, clustering and pattern recognition, and training, selection, and deployment of ML models. Based on the collected data, the main purpose of this actor is to generate the IL model that classifies users based on the changes of their consumption behavior and then support the decision making process performed by the network analyst which is described as follows.

3.1.4. Network analyst – Network manager – Network administrator

The fourth and final actor involved in the reference model is commonly known as the network analyst, network manager or network administrator. The role of the network administrator can vary significantly depending on an organization's size, location, and socio-economic considerations. To offer a general overview, a network administrator: "is the person designated in an organization whose responsibility includes maintaining computer infrastructures with emphasis on networking. Responsibilities may vary between organizations, but the key areas of focus are on-site servers, software-network interactions as well as network integrity/resilience." [106].

Nowadays is not uncommon to outsource the role of network administration. Therefore, a company dedicated to information systems can provide this asset to several clients at the same time in a remote manner.

In the context of the reference model, the network analyst is the person or group of people in charge of the decision-making process related to network maintenance, network policies application, resource allocation, and business strategies. This actor can leverage the valuable information provided by the data analyst to make well-based decisions regarding new strategies that can benefit the Internet Service Provider (ISP) either in economic or operational terms. Specifically, the network analyst receives the relevant information obtained through the study and classification of the users based on their OTT consumption behavior. Then, based on this information, makes the decisions that allow to define personalized service degradation policies in a way that the quality and network performance maintains its functionality while considering the users most used OTT applications when the service degradation has to be applied.

It is important to remark that in some cases, depending on the available staff, the activities performed by the network analyst and network expert (subsection 3.1.2) can be handled by the same person or group of people.

3.2. Components and Workflow

Now, since there is a common understanding of the different actors that are involved with the reference model, in this section a description on the components and workflow that is proposed to be followed is provided. The description is offered in a conceptual manner without giving technical or technological details. Such details will be provided on Chapter 4 were the model is applied on an experimental scenario. As it is illustrated on Figure 3.1, there are 3 components or levels and each of them are composed of different tasks that each actor has the responsibility to accomplish. First a general description of the components will be provided. Then, the workflow is described following step by step.

3.2.1. Data Gathering and Flow Generation

This component comprises all the processes that must be performed by the network expert since this actor holds all the knowledge related to network equipment and architecture. These processes (packet persistence and flow generation) involve all the activities related to configuring network devices, capturing and storing of IP packets, packet aggregation for flow generation, and application labeling.

These processes are based on the initial steps of a traditional flow monitoring architecture as illustrated in Figure 3.2.

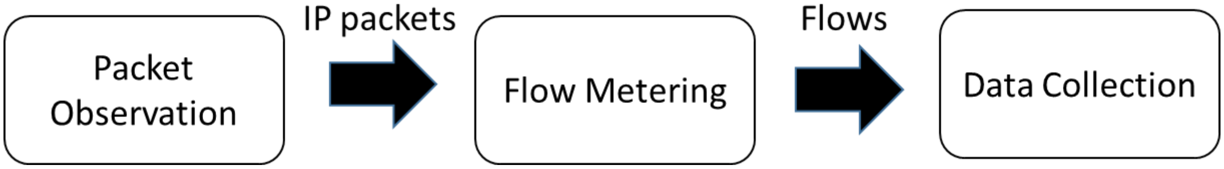


Figure 3.2. Traditional flow monitoring architecture [30].

The first stage is Packet Observation, in which packets are captured from an Observation Point and pre-processed. Observation Points can be line cards or network interfaces of packet forwarding devices. The second stage is Flow Metering, where packets are aggregated into flows. Then, after a flow is considered to have terminated, the flow record is exported, i.e., the record is placed in storing file format so it can be analyzed or transformed in a further process. Flow records can be imagined as records or rows in a typical database. This process of storing terminated flow records is known as Data Collection. At this point the gathered flows can be used for further analysis depending on the purpose of the research.

Based on this architecture, the processes that are related to this component will be explained as follows.

3.2.1.1 Packet Persistence

Similar to the first stage of Figure 3.2, Packet persistence (step 1 in Figure 3.1) is the process of capturing packets from the line and pre-processing them for further use. It is usually achieved by having a network device generating a copy of the packets sent through the network. Then, the packets are usually sent to storage. Packets are the fundamental elements that describe the communications established by network users. Usually, these packets are stored in packet capture (PCAP) files. Depending on the amount of traffic, these files might need a large storing capacity. Therefore, in networks where massive traffic volumes are generated, it is recommended to have an infrastructure capable of storing it. A generic architecture of the Packet Persistence stage is shown in Figure 3.3. Before any packet pre-processing can be performed, packets must be read from the network. This step, is typically carried out by a Network Interface Card (NIC). Before packets are stored in on-card reception buffers and later moved to the receiving host's memory, they have to pass several checks when they enter the card, such as checksum error checks.

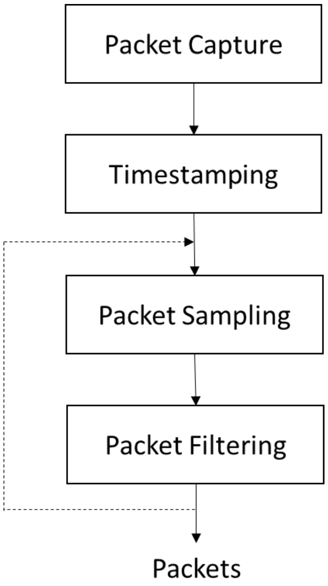


Figure 3.3. Packet persistence architecture [30].

The second step is Timestamping. Accurate packet timestamps are essential for many processing functions and analysis applications. When packets from different observation points have to be merged into a single dataset, usually they are ordered based on their timestamps. Both packet capture and Timestamping are performed for all packets under any condition. All subsequent steps shown in Figure 3.3, are optional. The first of them is packet truncation, which selects only those bytes of a packet that fit into a preconfigured snapshot length. This reduces the amount of data received and processed by a capture application, and therefore also the number of computation cycles, bus bandwidth and memory used to process the network traffic.

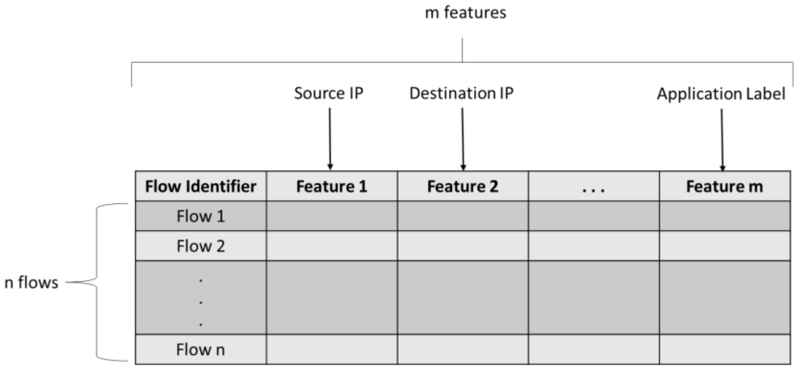
The last steps are packet sampling and filtering [107]. Capture applications may define sampling and filtering rules so that only certain packets are selected for measurement. The motivation for sampling is to select a packet subset, while still being able to estimate properties of the full packet stream. The motivation for filtering is to remove all packets that are not of interest. As a last remark, it is necessary to establish how much network traffic will be captured (hours, days, months, etc.). This should be specified from the beginning of the study so there is enough quantity of information and it reflects what wants to be analyzed. Once all the packets of interest are captured we can proceed to the flow generation process.

3.2.1.2 Flow Generation

A representation of the communications generated by users' network devices is obtained via packets being aggregated into network flows. To carry out Flow Generation (step 2 on Figure 3.1), two tasks are required to be carried out to achieve this (steps 2a and 2b):

- *Network statistics calculation:* once all the packets that belong to the same network flow are aggregated, a set of statistics can be calculated to obtain a statistical representation of the communication. The obtained statistics depend on the objectives of the measurement. The statistics can be on the forward direction (source to destination), backward direction (destination to source), or bidirectional. An example of common network statistics includes the IP addresses and ports, packet sizes, number of packets, interarrival times, flow durations, among many others.
- *Application labeling:* to obtain an overview of network users' consumption behavior, it is necessary to know the applications they are consuming. With this in mind, all the network flows are labeled with the respective application name that is being consumed. The most common approach is Deep Packet Inspection (DPI). Through this approach, the payload of network packets is inspected to obtain the application information.

Once the network flows are generated, a common way to store them is through CSV files, which do not require a large amount of storing capacity and can be leveraged by most ML software. On the other hand, an approach based on databases or any other storage mechanism that enables further processing can also be considered. Figure 3.4 illustrates an example of the structure of the network flows dataset obtained after this step.



Flow Identifier	Feature 1	Feature 2	. . .	Feature m
Flow 1				
Flow 2				
. . .				
Flow n				

Figure 3.4. Network flows dataset structure.

3.2.2. Data Preprocessing and Model Selection

This component comprises all the processes related to the data analyst. All the preprocessing that needs to be performed on the data delivered by the network expert is carried out on this component. The preprocessing steps include: flow cleaning (step 3), user consumption estimation (step 4), clustering and pattern recognition (step 5), and data cleaning (step 6). Furthermore, once the dataset is ready to be implemented, the data analyst is in charge of performing the IL algorithms comparison and evaluation to obtain the model that is capable of maintaining its performance consistency and knowledge retention over time (steps 7 and 8).

The workflow of the 6 steps that are performed by the data analyst on this component will be explained as follows.

3.2.2.1 Flow Cleaning

A lot of communications happen simultaneously inside a network. Such communications can be generated by user devices (smartphones, tablets, laptops, etc.) or network devices (routers, switches, etc.). Hence, it is possible that the flows delivered by the network expert still have information that is not useful for the study of users’ OTT consumption behavior. Since all of this communications are captured when performing the packet capture sessions, in step 3, it is crucial that the data analyst performs a flow cleaning process where all this unnecessary information is removed.

Therefore, all the flows that do not belong to communications generated by user devices should be removed from the data. This process usually requires a collaboration between the data analyst and the network expert. By leveraging their knowledge, the network experts can assess the data to ensure that the flows to be removed are not information with a high value.

Usually, the network expert knows the network domains where user devices are commonly assigned within the network. This way, the data analyst has the criteria to identify which flows should be removed. Depending on the objectives of the study, after this step the amount of flows can be considerably reduced. For example, if the objective is only to study the behavior of a single user, the flows of interest will be the ones generated by the IP address that corresponds to that network device.

Finally, the time window for the analysis should be defined. Depending on how much network traffic has been captured, the data analyst can decide to analyze the behavior of the users per specific hours, per day, per week, per month, etc. It all depends on the study objectives and on how many days' worth of network flows the network expert was able to capture.

3.2.2.2 User consumption estimation

Now, after the data analyst is certain that the remaining network flows belong to communications generated by the users of interest, it is possible to proceed with the estimation of user consumption since all the necessary information should be available in the network flows.

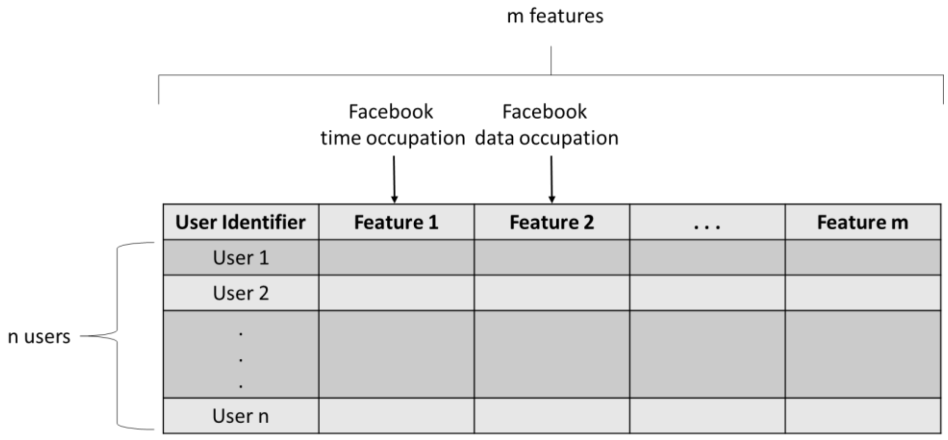
Since up to this point all the network flows are labeled with their respective OTT application (e.g., Facebook, YouTube, WhatsApp, etc.), it is possible to calculate different statistics using the network flows of each application as a starting point. It is important to estimate the consumption per user so each of them can be handled individually. The easiest way to identify users without invading their privacy is through the IP address assigned to his/her device. However, if there is access to another feature that enables such identification (e.g., IMSI in mobile networks or MAC addresses) it can be used as well.

Depending on the objectives of the study, the consumption estimation can include more or less information. For example, if the total quantity of packets sent in the forward direction (source to destination) is an interesting feature for the consumption analysis, then the data analyst can calculate this feature by adding all the number of packets sent in the forward direction of each flow per application (obtaining: total number of packets sent in the forward direction for YouTube, total number of packets sent in the forward direction for Facebook, etc.). Another example that can be considered is the total number of flows a user generated for each application. This can be calculated by counting the number of flows that had been labeled with each application. At the end it all depends on the information available in the network flows and the objectives established for the consumption analysis.

Since the objective is to support the definition of personalized service degradation policies, we need to determine which OTT applications are most commonly consumed by the users. The features that reflect such behavior from the users and allows us to determine such applications are the total amount of time the user spent on each application and the total amount of data exchanged through the network. Both features are illustrated in steps 4a and 4b on Figure 3.1. Both tasks are described as follows:

- *Time occupation calculation:* the amount of time spent by each user on the consumption of OTT applications offers essential insights for the network operator. Knowing this aspect enables the identification of their preferred applications and the high demand for network resources. Therefore, the amount of time that the user spent consuming each application must be calculated. This can be carried out by taking each flow duration and calculating the average flow duration for each application.
- *Data occupation calculation:* another aspect that offers essential insights on users' behavior is the amount of data (bytes) exchanged for each application. As in the previous case, this aspect allows us to observe if the user demands significant network resources and if a different data plan or special considerations might be more adequate to his/her needs. Therefore, the data occupation of each user must be calculated for every OTT application identified in the analysis. This can be done by taking the total number of bytes exchanged on each the network flow and then calculating an average for each application.

Finally, after this step, a new dataset holding a summarization of the consumption behavior of each user is obtained. On this dataset each instance represents a specific user and each column offers information related to a feature of interest on a specific OTT application. Figure 3.5 illustrates an example of the structure of the dataset that is obtained as output of this step.



User Identifier	Feature 1	Feature 2	. . .	Feature m
User 1				
User 2				
. . .				
User n				

Figure 3.5. User consumption dataset structure.

3.2.2.3 Clustering and pattern recognition

After obtaining a representation of the users' OTT consumption behavior on the previous step, the next question to answer is: how to classify the users based on their consumption behavior?

For this situation, unsupervised learning techniques, and specifically clustering algorithms are the most common and adequate mechanisms to identify patterns within the data and establish a classification scheme. Data clustering is an essential step in the arrangement of a correct and throughout data model. To fulfill an analysis, the volume of information should be sorted out according to the commonalities. Since there is no algorithm that can be applied for every case, the recommended solution is that the data analyst tests different clustering approaches, compares the results and chooses the one that delivers the best clustering scheme for user classification. Based on [108], [109], the most common clustering approaches will be briefly described as follows:

- *Connectivity based algorithms:* clustering based on the computation of distances between the data points of the whole dataset is called connectivity-based or hierarchical. Depending on the "direction" of the algorithm, it can unite or, inversely, divide the array of information (known as agglomerative and divisive algorithms respectively). The most popular and reasonable type is the agglomerative algorithms, where the data points are the input of the algorithm, and subsequently are united into larger and larger clusters, until the limit is reached. After applying one of the connectivity-based algorithms, a dendrogram of data is obtained. It presents the structure of the information rather than its distinct separation on clusters. Such structure may present benefits or harmful situations depending on the dataset. In some cases, the complexity of the algorithm may turn out to be excessive or simply inapplicable for datasets with little to no hierarchy. Also, if several iterations are applied the algorithm might show poor performance.
- *Centroid-based* algorithms: is the most frequently used clustering approach thanks to its comparative simplicity. The number of clusters (K) is chosen randomly, which is probably the greatest "weakness" of the method. The process of calculation consists of multiple steps. Firstly, the number of the clusters the dataset should be divided into is chosen. Based on this, the centroids of the clusters are situated as far as possible from each other to increase the accuracy. Secondly, the algorithm finds distances between each data point and every centroid. The smallest distance determines to which cluster the data point is

assigned. After that, the center of the cluster is recalculated according to the means of all data points' coordinates. Then, the first step of the algorithm is repeated with a new center of the recomputed cluster. Such iterations continue until either the centroid does not move or the maximum number of iterations are reached.

- *Gaussian mixture models*: these algorithms are supported on probability. On this approach, the relation probability that each data point belongs to each specified cluster is calculated. These algorithms take the assumption that the data points follow a Gaussian distribution. The most common algorithm of this approach is the Expectation Maximization (EM) algorithm [110].
- *Density based algorithms:* on this approach the main objective is to divide the dataset into clusters based on the ε parameter that determines the number of data points needed to establish a "neighborhood". If the object is located within the circle (sphere) of the ε radius, it belongs to the cluster. The most common algorithm of this approach is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [111]. This algorithm checks every data point, classifies it to its respective cluster or as noise, until finally the whole dataset is processed. The clusters determined with DBSCAN can have arbitrary shapes, thereby are extremely accurate. Besides, the algorithm determines the number of clusters automatically. However, DBSCAN has a drawback. If the dataset consists of variable density clusters or if the placement of data points is too close, the method shows poor results since the ε parameter can't be estimated easily.

Summing up, there is no such thing as a better clustering algorithm. Some of them are just more suitable for the particular dataset structures. Therefore, it is recommended to test several approaches and compare results. Once this step is finished, an additional feature will be added to the dataset. This feature will become the target variable for the IL algorithms, specifying the classes or groups the users can be classified into.

3.2.2.4 Data preprocessing

Up to this point we already have a dataset holding a representation of the users' OTT consumption behavior with their respective class labels. However, it is of common knowledge that poor data quality in data mining, ML and data science projects will negatively impact the quality of results of analyses and that it will therefore impact on the decisions made on the basis of these results. Hence, before moving forward to algorithm training and testing, it is important to analyze the quality of the data and identify possible problems among the gathered data. Based on [112] the most common problems that can be found in a dataset are described:

- *Outliers:* these are observations which deviate so much from other observations as to arouse suspicions that it was generated by a different mechanism. Outlier detection is used extensively in many applications. Current application areas of outlier detection include detection of credit card frauds, detecting fraudulent applications or potentially problematic customers in loan application processing, intrusion detection in computer networks, among many others.
- *Noise:* defined as irrelevant or meaningless data in the instances. For a given domain-specific dataset, attributes that contain a significant amount of noise can have a detrimental impact on the success of a knowledge discovery initiative, e.g., reducing the predictive ability of a classifier in a supervised learning task.
- *Inconsistency:* refers to a lack of harmony between different parts or elements; instances that are self-contradictory, or lacking in agreement when it is expected. This problem is also known as mislabeled data or class noise. e.g., in supervised learning tasks, two instances have the same values, but have different labels or the label values do not correspond itself.
- *Incompleteness:* it is widely recognized that datasets are affected by missing values. Most of the time this happens due to several reasons like: sensor faults, a lack of response in scientific experiments, faulty measurements and data transfer problems in digital systems.
- *Timeliness*: has been defined as the degree to which data represent reality from the required point in time. When the state of the world changes faster than our ability to discover these changes and update the data repositories accordingly, the confidence on the validity of data decays with time e.g., people move, get married, and even die without filling out all necessary forms to record these events in each system where their data is stored.
- *Amount of data*: the amount of data available for model building has a direct impact in the performance of the algorithms. Small and imbalanced datasets build inaccurate models. Therefore, it is important to analyze the total amount of data and if the classes within the dataset have the same amount of instances (balanced dataset).

After analyzing if the dataset presents any of the common problems around data quality and solving such issues, we can move forward into the selection and comparison of IL algorithms.

3.2.2.5 Selection and training of incremental learning model

Moving on, in step 7, a selection and training of an IL model is performed. By leveraging the structured information obtained with the dataset up to this point, a set of IL algorithms must be tested to identify which model can maintain their retention, consistency, and comprehension of the data over time.

Evaluating the performance of IL algorithms presents specific problems and must take into account aspects not present in conventional evaluation of learning. Papers such as [113], [114] describe measures of performance that are aimed to address each of the significant characteristics of IL algorithms. These measures and approaches are described as follows:

- *Generality:* This is the kind of concept which can be described by the representation mechanism and is learnable by the algorithm. Through this approach, the objective is to observe if is the representation that the algorithm obtains from the data (a hyperplane, a decision tree, a voting scheme, a set of rules) is accurate for the task at hand (classification or regression) and obtains good results based on this representation.
- *Accuracy*: This is the classification accuracy, i.e., the traditional statistics that are analyzed in ML algorithms. This includes the success or error rate that the algorithm obtains when performing the classification and are usually represented through several values (e.g., accuracy, precision, Kappa statistic, etc.).
- *Learning Rate*: This is the speed at which the classification accuracy increases during training. It is a more useful indicator of the performance of the learning algorithm than the accuracy for finite-sized training sets. The objective is to observe if the algorithm increases, maintains or decreases its performance when receiving knowledge from new data either at warm-up phase (training) or during the testing phase.
- *Incorporation Costs*: These are the costs incurred while updating the concept descriptions of the algorithm with a single training instance. They include classification costs in terms of accuracy and performance. Furthermore, every organization should consider economical and functional costs when the algorithm is updated since an algorithm that takes too much time in training or updating can present a negative impact on the activities of the organization.

- *Storage Requirement*: This includes the costs in terms of computational requirements for the implementation of the IL algorithm. Common questions that should be considered for this are: How much data does the algorithm maintain in memory to maintain and update its concept representation (decision tree, hyperplane, rules set, etc.)? – Which computational requirements does the algorithm need to be deployed? – How many disk storage capacity is needed?

The most common approach when evaluating performance in ML algorithms is the accuracy approach. However, it is up to the data analyst which approach to implement as well as which algorithms to test. At the end of this process the data analyst will have an adequate IL algorithm to deploy for the user classification.

3.2.2.6 User classification model

Finally, at the end of the second component (step 8) an IL algorithm has been selected trained and evaluated in terms of performance. This algorithm should be capable of classifying the users’ consumption behavior while maintaining its performance consistency over time. This IL model will offer complete and timely information to the network analyst about how the users behave in terms of OTT applications and support the decision making process.

3.2.3. Decision Making

Once all the processes have been successfully completed, on the last component (step 9), the network analyst can take appropriate actions regarding the *decision-making* process.

With the users' information provided by the classification model, the network analyst has the freedom to develop new strategies to handle aspects like network policies, resource allocation, data plans and infrastructure deployment, among many others.

As has been mentioned before, within this project, the purpose of this process is supporting the network analyst in the definition of personalized service degradation policies. This specific use case will be described on a further section of this document after applying the model to an experimental scenario and analyzing the obtained results (chapter 6).

Summary

The major conclusion and contribution of this chapter is the proposal of the reference model with all its components defined from a conceptual perspective.

Specifically, this chapter presented a detailed description of the reference model that is defined to offer guidelines to researchers in the field of network flow monitoring, and network operators, to support their decision making process when dealing with the definition and personalization of service degradation policies. This model defines a series of steps that can be replicated to study and characterize users' OTT consumption behavior and obtain an IL model capable of classifying each user based on his/her consumption behavior.

First, the actors that are involved in the model were presented. The actors included four important roles:

- Network user
- Network expert
- Data analyst
- Network analyst.

Then the three components (data gathering and flow generation, data preprocessing and model selection, decision making) and each of their purposes are described.

Subsequently, all the 9 processes that should be carried out on each component are presented along with important details and observations that should be considered when applying the reference model. In certain processes several decisions depend on the aim and perspective of the actor responsible.

Chapter 4

Data Gathering and user consumption estimation

Up to this point, the fundamental concepts, state of the art and the reference model proposal for studying users' OTT consumption behavior has been described. In this chapter, an initial approach that allowed the creation of a set of synthetic data generators to obtain new datasets aiming at obtaining an IL model capable of classifying users' consumption behavior is provided. However, since the results obtained with those generators were not as expected due to the fact that the generators were unable to create data that truly reflected a real consumption behavior from users, this chapter also describes the application of the reference model to an experimental scenario aimed at studying the consumption behavior of the people within the Universidad del Cauca network. The application of the reference model allowed the construction of a new dataset that will be implemented later for the comparison of IL algorithms.

4.1. Synthetic data generators

With the aim of obtaining a representative dataset and compare the performance of traditional and incremental ML algorithms when classifying users' OTT consumption behavior, as a first attempt, a set of synthetic data generators were created. Those generators were based on the dataset obtained on the masters' thesis that preceded this project. With this in mind, this section will present a brief description of the dataset and how these generators were created.

4.1.1 Dataset description

The dataset used as a base for the generators is a result described in [7] and is available for download in [18]. This dataset contains 1,581 instances and 131 attributes on a single CSV file. Each instance represents a user’s consumption profile which holds summarized information about the consumption behavior of the user, related to the 29 OTT applications identified in the different IP flows captured in an experiment scenario during 2017.

The OTT applications that the users interacted with during the experiment and were stored in the dataset included: Amazon, Apple store, Apple Icloud, Apple ITunes, Deezer, Dropbox, EasyTaxi, Ebay, Facebook, Gmail, Google, Google Maps, Browsing (HTTP, HTTP_Connect, HTTP_Download, HTTP_Proxy), Instagram, LastFM, Microsoft One Drive (MS_One_Drive), Facebook Messenger (MSN), Netflix, Skype, Spotify, Teamspeak, Teamviewer, Twitch, Twitter, Waze, WhatsApp, Wikipedia, Yahoo and YouTube. Each application has 4 different types of attributes (quantity of generated flows, mean duration of the flows, average size of the packets exchanged on the flows and the mean bytes per second on the flows). These attributes summarize the interaction that the user had with the respective OTT application in terms of consumption. Furthermore, the dataset contains the user’s IP address in network and decimal format which are used as user identifiers. Finally, the User Group attribute represents the objective class in which a user is classified considering his/her OTT consumption behavior. There are 643 users for the high consumption profile, 475 users for the medium consumption profile and 463 users for the low consumption profile. Table 4.1 describes the features stored on this dataset.

4.1.2 Generators development

The idea of creating synthetic data generators came up after noticing that the number of instances stored on the dataset (1,581 instances) were considerably low for the IL tests. Additionally, the process needed to be carried out in order to generate more instances from raw data of IP flows required an important effort in terms of time.

Attribute name	Attribute description
Source.Decimal	This attribute holds the user’s IP address in decimal format and it is mainly used as a user identifier.
Source.IP	This attribute holds the user’s IP address in network format (e.g., 192.168.14.35) and as in the previous case its main function is to work as a user identifier.
Application-Name.Flows	This group of attributes hold the information about the quantity of IP flows that a user generated toward an OTT application. As was mentioned before each application has a group of 4 attributes that describe the interaction of the user with a specific OTT application (an example for this case would be Netflix.Flows or Facebook.Flows).
Application-Name.Flow.Duration.Mean	This group of attributes hold the information related to the mean duration (time) of the flows generated by the user towards a specific OTT application, measured in seconds. Examples of how this attributes are stored in the dataset are: Amazon.Flow.Duration.Mean or Instagram.Flow.Duration.Mean
Application-Name.AVG.Packet.Size	This group of attributes hold the average size of the IP packets that were exchanged in all the flows generated by the user towards a specific OTT application, measured in bytes. It is important to notice that this size is focused on the packet’s header only. Examples of how this attribute are presented on the dataset are: Google_Maps.AVG.Packet.Size or Spotify.AVG.Packet.Size
Application-Name.Flow.Bytes.Per.Sec	This group of attributes hold the mean number of bytes per second that were exchanged in the flows generated by the user towards a specific OTT application. Examples of this kind of attributes in the dataset are: Deezer.Flow.Bytes.Per.Sec or Skype.Flow.Bytes.Per.Sec.
User.Group	This group of attribute represents the objective class of the dataset i.e., the different groups that the users are classified in according to their OTT consumption behavior. Those groups are: High consumption (643 instances), Medium consumption (463 instances) and Low consumption (475 instances).

Table 4.1. Dataset structure for the synthetic data generators.

Therefore, the development of a synthetic data generator of users’ OTT consumption profiles based on the statistical distribution obtained on the current dataset was considered to be the most adequate option. With this in mind, by assuming that all the attributes in the dataset were mutually independent and by leveraging the features of the R programming language, specifically the ‘‘fitdist’’ library [115] that allows to fit univariate theoretical distributions (normal distribution, beta distribution, gamma distribution, etc.) to non-censored data using different estimation methods, it was possible to infer a statistical distribution for each attribute in order to generate new data that presented the same statistical behavior observed on the original dataset. Figure 4.1 illustrates the process that had to be carried out to create the synthetic data generators.

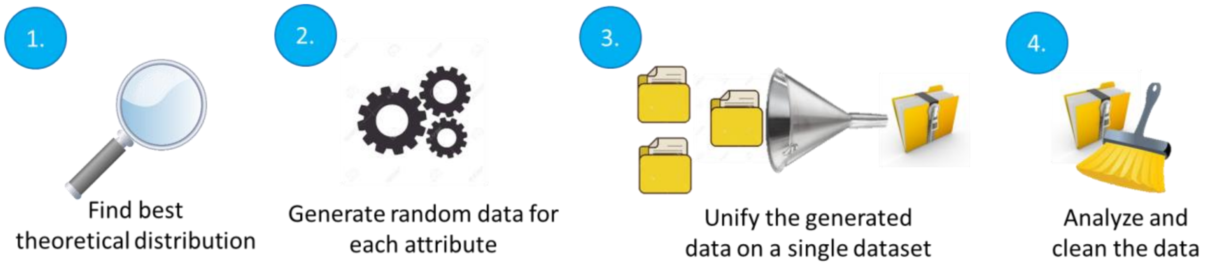

1.
Find best theoretical distribution
2.
Generate random data for each attribute
3.
Unify the generated data on a single dataset
4.
Analyze and clean the data

Figure 4.1. Synthetic data generators – step by step.

The first step in order to perform the statistical estimation was to separate all the instances from each objective class (high, medium and low consumption) since the statistical estimation had to analyze the distribution of all the attributes on each type of consumption profile individually. After separating the instances of each class it was decided that the estimation should be performed on 130 attributes (removing the objective class). Each attribute distribution was analyzed through a Cullen and Frey graph where the kurtosis and the square of the skewness [116] of the data distribution are compared to the same measures of different theoretical distributions (normal, gamma distribution, etc.) in order to determine which distribution fits best to the behavior of the data. As an example of the process, Figure 4.2 illustrates the Cullen and Frey graph obtained for the attribute Twitter.Flows of the instances classified as high consumption users. This process was carried out for all the 29 OTT applications stored on the dataset.

On Figure 4.2, the blue dot illustrates the intersection of the square of the skewness and the kurtosis obtained from the attribute data, and the different lines and areas represent a theoretical distribution that could fit the data. For this specific case, the data can be fit into a beta or gamma distribution. Therefore, in order to decide which of these two distributions is the better fit, a one-sample Kolmogorov-Smirnov test [117], [118] is performed obtaining two values: The first one is the maximum distance between the empirical CDF (Cumulative Distribution Function) obtained from the data and the theoretical CDF from the beta and gamma distributions in this specific case. The second one is the p-value that allows to accept the null hypothesis (the data shows a distribution similar to a specific theoretical distribution) if it is higher than 0.05. As can be observed on Table 4.2, by comparing the results obtained with the Kolmogorov-Smirnov test, the distribution that fits best to the data from the Twitter.Flows attribute is the beta distribution since the maximum distance between the CDF's is closer to zero and the p-value is higher than 0.05 and is higher than the p-value from the gamma distribution.

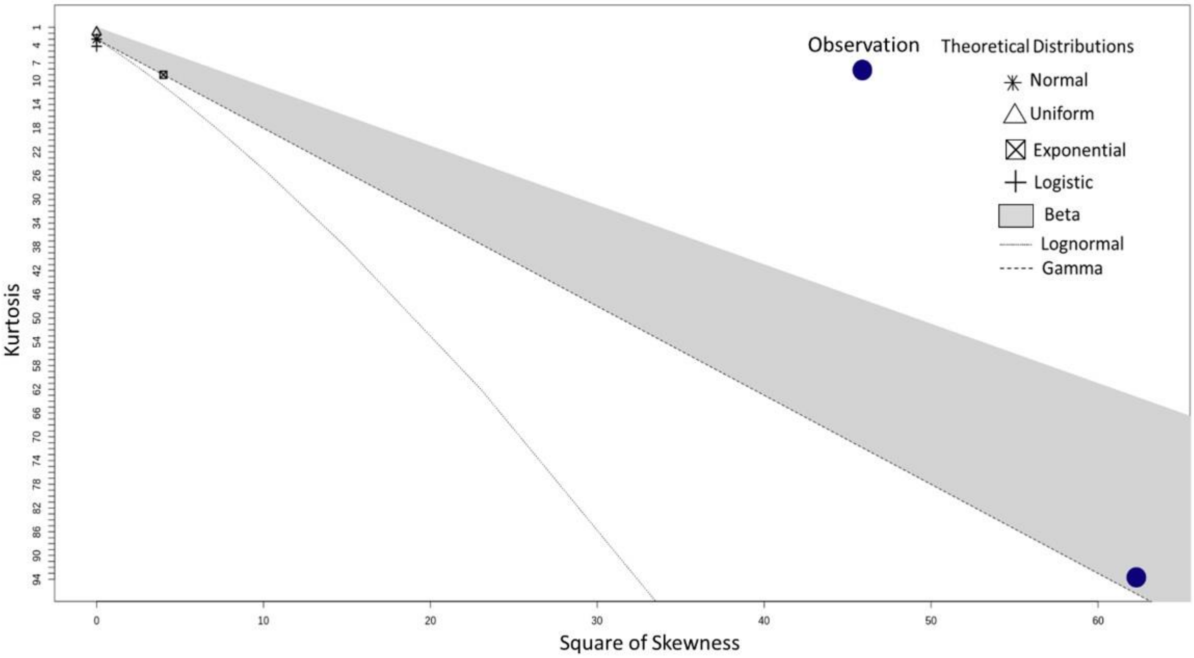


Figure 4.2. Cullen and Frey graph – Twitter.Flows attribute – High consumption profile.

Theoretical Distributions	Maximum distance between CDF's	P-value
Beta Distribution	0.0039431	0.4185
Gamma Distribution	0.0051021	0.148

Table 4.2. Kolmogorov-Smirnov test - Twitter.Flows attribute – High Consumption profile.

Once the best theoretical distribution is identified, a comparison between the empirical and theoretical CDF's is performed, fitting the theoretical statistical distribution to the attribute data distribution using maximum likelihood estimation. Such comparison is illustrated in Figure 4.3 through 4 plots: the empirical and theoretical density, the

Quantile-Quantile plot (Q-Q plot), the empirical and theoretical CDF's and the Probability-Probability plot (P-P plot). The red line represents the behavior of the CDF from the beta distribution. It can be observed that this distribution is a good fit since the data exhibits a similar behavior. Afterwards, the variables that describe the distribution (α and β for this case since it is a beta distribution) are calculated and the synthetic data are generated for the attribute using the R programming language. This process is repeated for all the other attributes from the dataset and for the three classes (390 attributes in total) creating 3 data generators, one per consumption profile (high, medium and low consumption).

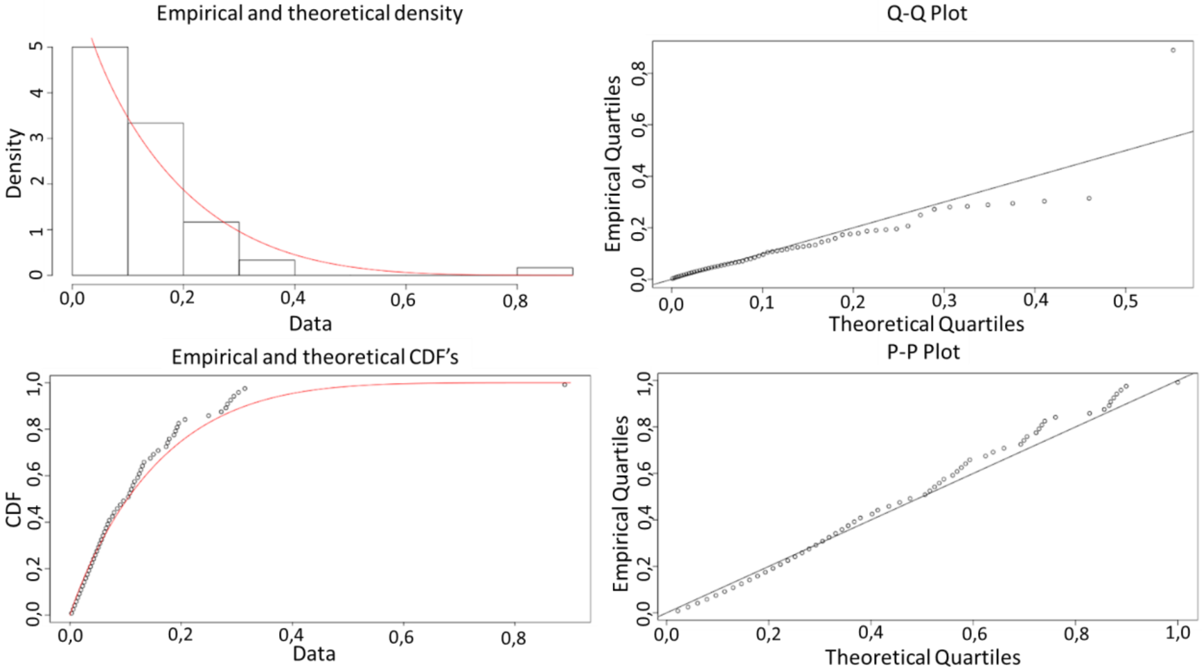


Figure 4.3. CDF comparison - Twitter.Flows attribute – High Consumption profile.

Once the three data generators were finished, a synthetic dataset holding 150,000 instances (50,000 for each class – high, medium and low consumption) was created and the instances were shuffled in order to proceed with the performance comparison of the traditional batch learning algorithms and the IL algorithms. The result of that comparison will be presented in detail on Chapter 5.

It is important to mention that the synthetic data generators were created assuming statistical independence among the dataset features. This means that the data generation process from each feature does not consider the others. It had to be performed like this since it was not possible to create a maximum likelihood estimation library that considered the correlation between 2 or more features. Therefore, there might be some instances after the generation process that do not reflect the behavior of data captured from actual users on a real network. Due to this consideration, it was necessary to apply the reference model on an experimental scenario and create a new dataset with real data of users' OTT consumption behavior. This will be explained in Section 4.2.

4.2. Users dataset generation and preprocessing

This section presents the application of the first six steps of the reference model, that are included in the Data Gathering and Flow Generation component and the Data Preprocessing and Model Selection component. All of this aims at obtaining a dataset that summarizes the actual behavior of users' OTT consumption in order to create an IL model that supports the decision making process of a network manager when defining personalized service degradation policies. Figure 4.4 presents a deployment model where the software tools and technologies implemented on each step are depicted. The application of each step of the reference model and the obtained results will be described as follows.

4.2.1. Packet persistence

As explained on section 3.2.1.1, this step aims at capturing and storing IP packets that hold the information of the communications generated by the users within an Internet network. To perform the capture sessions, a server was configured with Wireshark within the network core of the Faculty of Electronic and Telecommunications Engineering of Universidad del Cauca to capture the IP packets and store them on PCAP files. Specifically, the capture sessions had a duration of 30 minutes each, and were performed during mornings and afternoons between April and June of 2019. At the end of the capture sessions around 3 TB (TeraBytes) worth of PCAP files were stored to be analyzed and transformed into network flows.

4.2.2. Flow generation

Once all the packet capture sessions were finished, it was necessary to develop and application capable of aggregating such packets into flows. There are several existent applications that can convert IP packets into flows while obtaining flow statistics;

however, none of them were capable of identifying the application that was being used on the flow. For this reason, we developed our own application, named Flow Labeler, to do both steps: network statistics calculation and application labeling.

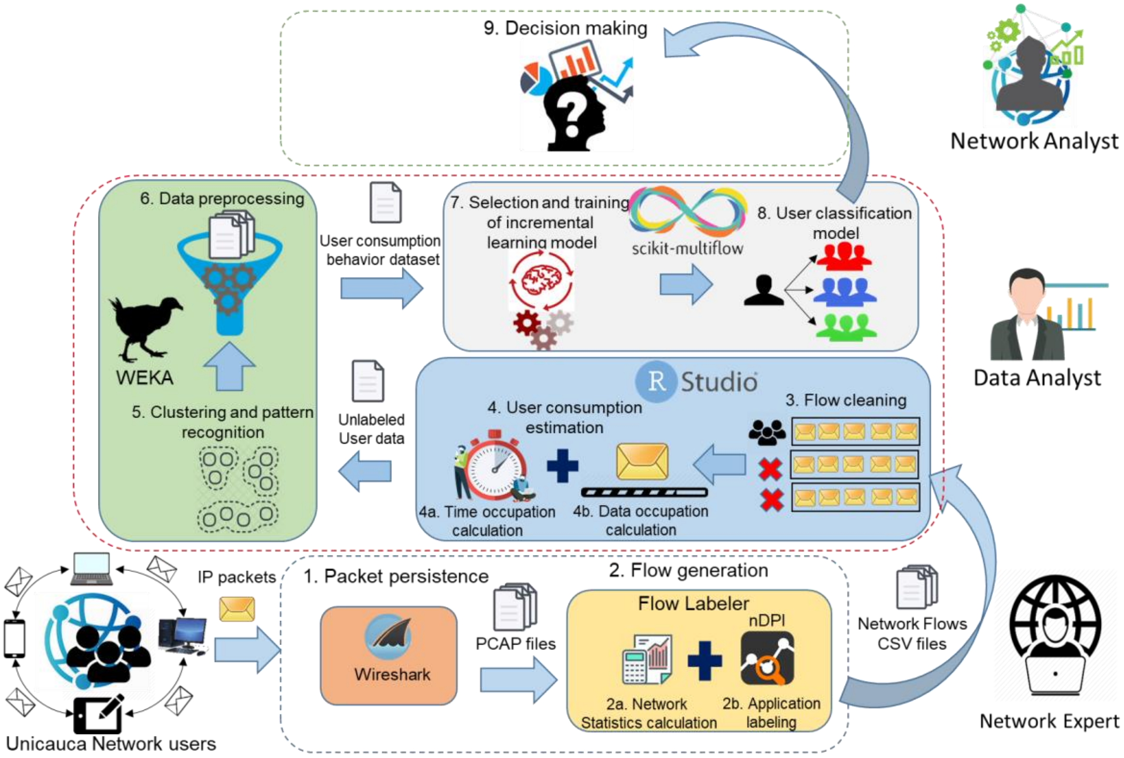


Figure 4.4. Deployment model.

This application is based on FlowRecorder [119] and NFStream [120], integrates their functionalities and will be described as follows.

4.2.2.1 Flow Labeler

Flow Labeler [121] was developed with Python 3.6.8 and is capable of aggregating IP packets into flow records either from PCAP files or in a live capture mode. At the end of the PCAP file processing or live packet capture session and by using nDPI library [122], Flow Labeler generates a CSV (Comma Separated Value) file with the flow records containing bidirectional statistics and the application label. Table 4.3 illustrates the 50 attributes that are calculated per flow and Figure 4.5 depicts the flow diagram of the processing that Flow Labeler performs on each IP packet.

Feature name	Feature description	Direction		
		FWD	BWD	Bidirectional
flow_key	Flow identifier through a hash algorithm	N.A.	N.A	✓
src_ip_numeric	Source IP in decimal format	N.A.	N.A	✓
src_ip	Source IP in network format	N.A.	N.A	✓
src_port	Source port number	N.A.	N.A	✓
dst_ip	Destination IP in network format	N.A.	N.A	✓
dst_port	Destination port number	N.A.	N.A	✓
proto	Transport protocol number according to IANA (e.g., 1 for ICMP, 6 for TCP, 17 for UDP)	N.A.	N.A	✓
pktTotalCount	Total number of packets in both directions	✓	✓	✓
octetTotalCount	Total of bytes exchanged focusing on the IP payload only	✓	✓	✓
min_ps	Minimum packet size on the flow	✓	✓	✓
max_ps	Maximum packet size on the flow	✓	✓	✓
avg_ps	Average packet size on the flow	✓	✓	✓
std_dev_ps	Packet size standard deviation	✓	✓	✓
flowStart	Flow start time in seconds using UNIX time format	✓	✓	✓
flowEnd	Flow end time in seconds using UNIX time format	✓	✓	✓
flowDuration	Total flow duration in seconds using UNIX time format	✓	✓	✓
min_piat	Minimum packet interarrival time on the flow	✓	✓	✓
max_piat	Maximum packet interarrival time on the flow	✓	✓	✓
avg_piat	Average packet interarrival time on the flow	✓	✓	✓
std_dev_piat	Packet interarrival time standard deviation	✓	✓	✓
flowEndReason	The reason why the flow was expired and sent to the final array that will be converted to CSV file: 0 inactive timeout expired - 1 active timeout expired - 2 forced expiration due to end of pcap file or live captured stopped - 3 FIN flag detected on both directions - 4 RST flag detected - 5 FIN Flag detected on one direction only and timer expired	N.A.	N.A	✓
category	Category of the communication as delivered by nDPI	N.A.	N.A.	✓
application_protocol	Application protocol for the flow (e.g., TLS, HTTP, DNS, etc.) detected by nDPI	N.A.	N.A.	✓
web_service	Web service detected by nDPI (e.g., Facebook, WhatsApp, Google, etc.)	N.A.	N.A.	✓

Table 4.3. Attributes generated by Flow Labeler – 50 attributes.

As observed on Figure 4.5, Flow Labeler receives an IP packet (either from a PCAP file or directly from the network interface card) and by looking to its network identifiers (Source and destination IP addresses and ports) checks if the packet belongs to a new flow or not. If the packet belongs to a new flow, a flow record is created on an in-memory array and then Flow Labeler continues with the next packet.

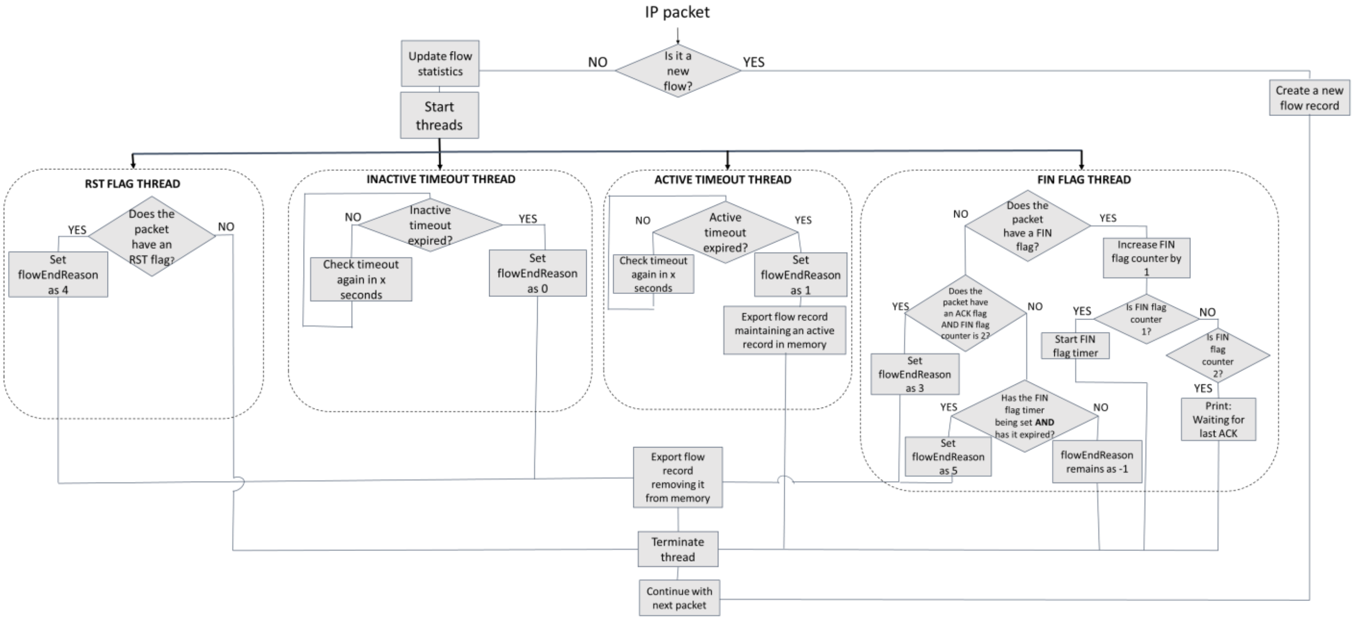

IP packet
Is it a new flow?
NO
YES
Update flow statistics
Start threads
Create a new flow record
RST FLAG THREAD
Does the packet have an RST flag?
YES
NO
Set flowEndReason as 4
INACTIVE TIMEOUT THREAD
Inactive timeout expired?
NO
YES
Check timeout again in x seconds
Set flowEndReason as 0
ACTIVE TIMEOUT THREAD
Active timeout expired?
NO
YES
Check timeout again in x seconds
Set flowEndReason as 1
Export flow record maintaining an active record in memory
FIN FLAG THREAD
Does the packet have a FIN flag?
NO
YES
Increase FIN flag counter by 1
Does the packet have an ACK flag AND FIN flag counter is 2?
YES
NO
Set flowEndReason as 3
Is FIN flag counter 1?
YES
NO
Start FIN flag timer
Is FIN flag counter 2?
YES
Print: Waiting for last ACK
Has the FIN flag timer being set AND has it expired?
YES
NO
Set flowEndReason as 5
flowEndReason remains as -1
Export flow record removing it from memory
Terminate thread
Continue with next packet

Figure 4.5. Flow Diagram - Flow Labeler.

If the packet belongs to an existent flow record, Flow Labeler updates all the flow statistics and then starts four threads in parallel. The first one is the *RST Flag thread*; it checks if the packet has the RST (Reset) flag set, if it does the “flowEndReason” attribute is set as 4 and the flow is exported from the in-memory array to the final CSV file. If the packet does not have the RST flag, then the thread is terminated and the flow record remains active within the in-memory array. The second thread is the *Inactive timeout thread*. Once started, this thread checks periodically if the flow has been inactive for a period of time specified by the user, i.e., it checks if no packets have been aggregated to the flow during a certain amount of time (usually between 15 seconds and 5 minutes). If this condition is met, the flow is exported from the in-memory array to the CSV file while setting the “flowEndReason” attribute as 0. If the timer has not expired yet the thread remains active while checking the expiration once again after another amount of time also specified by the user (e.g., 5 seconds).

The third thread is the *Active timeout thread*. Contrary to the previous case, this thread checks periodically if the flow has been active for a long period of time. When the timer expires means that the flow has been active over a time specified by the user (usually between 120 seconds and 30 minutes), then the current flow statistics are exported to the CSV file while maintaining an active flow record on the in-memory array and the “flowEndReason” is set to 1. If the active timeout has not expired, then the timer expiration is checked again after another period of time specified by the user.

Finally, the fourth thread is the *FIN flag thread*. This thread checks if the current packet has the FIN (finish) flag set. This flag terminates a flow communication. Since both the source and the destination hosts need to send a packet with a FIN flag to terminate the communication, if Flow Labeler detects a FIN flag for the first time it sets a counter to 1 and starts a timer that once expired exports the flow to the CSV file while setting the “flowEndReason” attribute to 5. This timer is created in case there was an error in the capture session and the second FIN flag was never received. On the other hand, if the FIN flag timer has not expired and the second FIN flag arrives, the counter is set to 2 and then Flow Labeler waits for a packet holding an ACK flag (acknowledged) since this flag is the last step to terminate the communication. When both the ACK flag is detected and the FIN flag counter is 2 the flow is exported to the CSV file and the “flowEndReason” is set to 3.

All this process is repeated for all the packets that are either stored on a PCAP file or captured during a live-capture session. Once the process is terminated, all the flows that still remain within the in-memory array are flushed to the CSV file while setting the “flowEndReason” attribute as 2. After all the captured PCAP files were processed a dataset holding a total of 2,704,839 instances, 50 attributes and 141 application labels was obtained which can be downloaded on [123].

4.2.3. Flow cleaning - user consumption estimation – clustering and pattern recognition – data preprocessing

Moving forward with the steps proposed on the reference model, we performed a flow cleaning process, while leveraging the tools offered by RStudio framework, on the dataset obtained through Flow Labeler. First, a filtering process was applied to get only flows belonging to a specific IP address range of user devices. Following the guidelines given by the network experts from Universidad del Cauca, all flows that were generated between the IP addresses 192.168.121.0 and 192.168.129.255 were considered as communications generated by user devices while removing all the flows outside of this range (flows with applications like DNS,Windows Update, UbuntuOne, etc).

Then, a selection process was applied to reduce the number of labeled flows among these range of IP addresses. The original dataset contained flows belonging to 141 application labels, however, not all of those labels were from OTT applications. Since our focus is the study of users' OTT consumption behavior the aim was to leave only the flows related to this type of applications. After this filtering, flows related to 56 OTT applications were identified and preserved. These applications include, among many others, Amazon, Deezer, Dropbox, Facebook, Gmail, Google, LinkedIn, Netflix, Spotify, Twitter, WhatsApp, and YouTube.

After these two filtering processes, the next step was to perform the user consumption estimation. For this we considered two approaches for the attributes calculations. On the first approach, we considered in the calculations for the summary of users' behavior all the numeric bidirectional statistics obtained with Flow Labeler to be included in the dataset. This approach considered several flow-based statistics like: the average packet size, average minimum and average maximum packet size, as well as the average minimum and average maximum interarrival times of the flows among many others, all of them per OTT application. On the second approach, based on the domain knowledge and recommendations from network experts, the consumption of an OTT service can be measured using the amount of time a user spent consuming the application, and the amount of information (bytes) exchanged through the network. Therefore, we considered only the set of the attributes that were directly related to users' consumption behavior: time occupation and data occupation per OTT application obtaining a dataset with 114 attributes. Table 4.4 shows the attributes structure of the dataset obtaining a total of 618 attributes and 716 instances (attributes related to the domain knowledge are marked with an asterisk).

After all the calculations were obtained, it was necessary to perform a clustering and pattern recognition process since the data remained unlabeled and to determine a way to classify the users according to their consumption behavior. This was performed on both approaches: the complete dataset with 618 attributes and the subset based on the knowledge domain with 114 attributes. On both cases the first step was to determine the ideal number of clusters or groups that the users could be classified on. This number was obtained via a combination of the average silhouette method and the elbow method (external and internal cluster measures, respectively) [124], [125]. Figure 4.6 and Figure 4.7 presents the results from both methods for the dataset with 618 attributes, and Figure 4.8 and 4.9 depicts the results for the dataset with 114 attributes.

In the elbow method the ideal number of clusters is where the total within sum of squares minimizes while forming an "elbow". On the other hand, in the silhouette method ideal number of clusters is where the average silhouette width is closer to 1.

Group of attributes name	Description
src_ip_numeric - 1 attribute	This attribute holds the users' IP in decimal format. It is mainly used as a user identifier
ApplicationName_time_occupation - 56 attributes*	These attributes hold all the information related to the amount of time the user spent using the application measured in seconds.
ApplicationName_data_occupation - 56 attributes*	These attributes hold all the information related to the number of bytes the user exchanged while using each application.
ApplicationName_mean_packetsNumber - 56 attributes	These attributes hold all the information related to the mean number of packets sent on the flows generated by the user on each application.
ApplicationName_mean_minimum_PacketSize - 56 attributes	These attributes show the average minimum packet size that the user flows exhibit on each OTT application.
ApplicationName_mean_maximum_PacketSize - 56 attributes	These attributes show the average maximum packet size that the user flows exhibit on each OTT application.
ApplicationName_mean_avg_PacketSize - 56 attributes	These attributes hold the average packet size that the user flows exhibit on each OTT application.
ApplicationName_mean_std_dev_PacketSize - 56 attributes	These attributes hold the average standard deviation of packet size that the user flows exhibit on each OTT application.
ApplicationName_mean_min_piat - 56 attributes	These attributes hold the average minimum packet interarrival time that the user flows exhibit on each OTT application.
ApplicationName_mean_max_piat - 56 attributes	These attributes hold the average maximum packet interarrival time that the user flows exhibit on each OTT application.
ApplicationName_mean_piat - 56 attributes	These attributes hold the average packet interarrival time that the user flows exhibit on each OTT application.
ApplicationName_mean_std_dev_piat - 56 attributes	These attributes hold the average standard deviation of the packet interarrival time that the user flows exhibit on each OTT application.
User group - 1 attribute	This attribute represents the target class of the dataset i.e., the users that are classified in three different groups (high, medium and low consumption).

Table 4.4. User consumption dataset structure – 618 attributes.

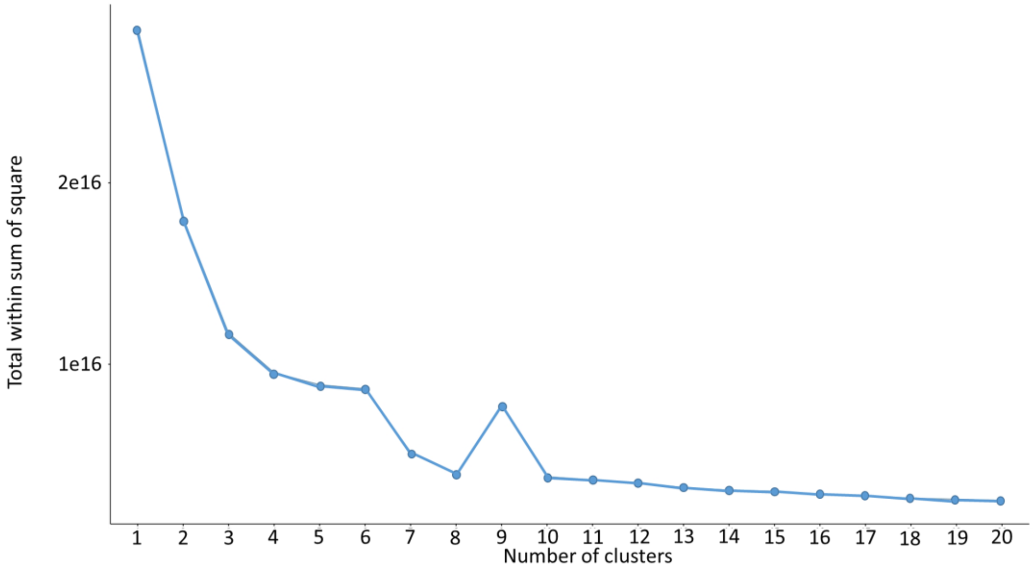


Figure 4.6. Elbow method results – 618 attributes

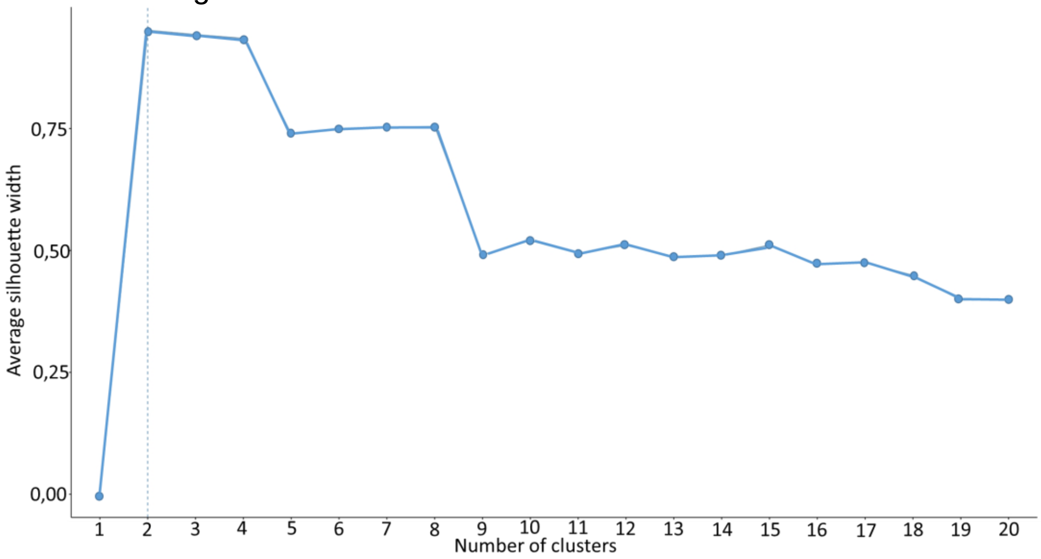


Figure 4.7. Silhouette method results – 618 attributes

With this in mind, it can be observed that in all four figures the result for both elbow and silhouette method is almost identical having an ideal number of clusters on either 4 or 7 clusters. Now, since we have determined the number of clusters we can proceed with the actual clustering process. Following the description presented by the reference model four clustering approaches were tested for both datasets. Specifically: Connectivity based algorithms, Gaussian mixture models, density based clustering and centroid based clustering. The obtained results will be analyzed subsequently.

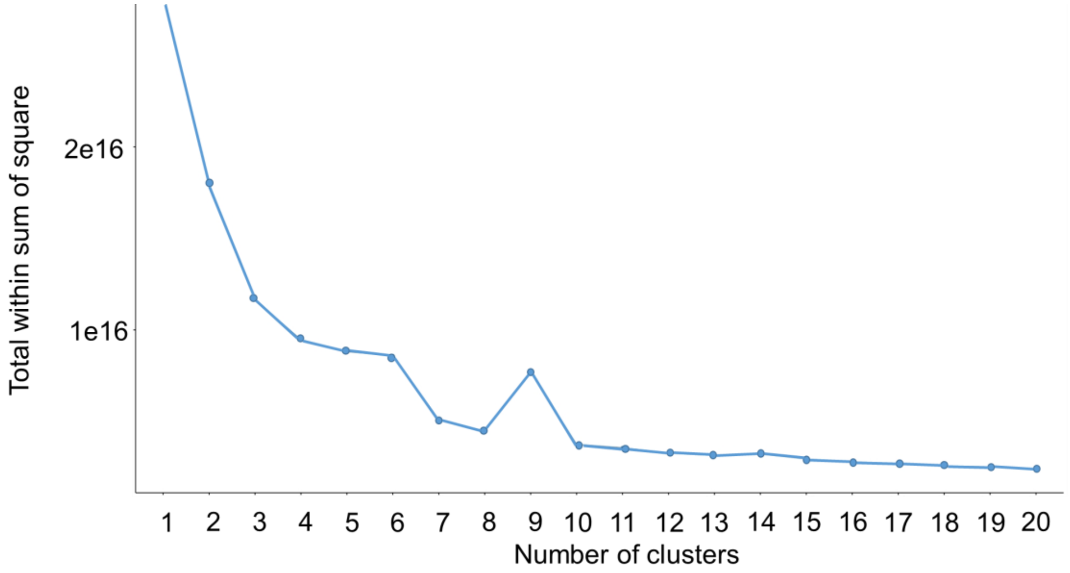


Figure 4.8. Elbow method results – 114 attributes

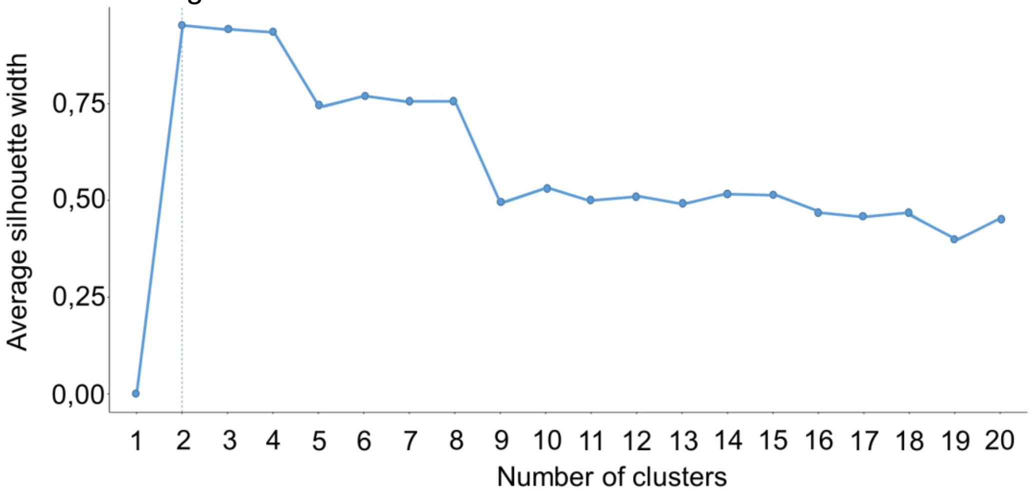


Figure 4.9. Silhouette method results – 114 attributes

- **Connectivity based algorithms:** For this approach two algorithms were tested Cobweb and hierarchical clusterer.
    - **Cobweb:** This algorithm incrementally organizes observations into a classification tree [126]. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic value. Therefore, this algorithm automatically determines the number of clusters. For the complete dataset it detected 829 clusters while for the dataset with 114 attributes it determined 789 clusters. Most likely, this is provoked by the fact that Cobweb is based on the assumption that probability distributions

on separate attributes are statistically independent of one another. Therefore, without considering the correlations between attributes this algorithm is unable of clustering the users' consumption behavior.

  - **<u>Hierarchical clusterer:</u>** This algorithm implements a classic agglomerative approach to obtain the clusters (bottom-up). The tests were performed for both datasets and for both ideal number of clusters (4 and 7). For the test with 7 clusters, in both datasets, the algorithm divided the instances in 7 groups, however, from a total of 716 instances, 710 instances were assigned to one cluster while the rest was assigned to the other six groups, one instance per cluster. For the test with 4 clusters similar results were obtained on both datasets. One cluster was assigned with 713 instances while the other 3 instances were assigned to the other 3 clusters. This allowed us to conclude that this algorithm was unable to identify differences and separate the users' OTT consumption behavior into groups.

- **Gaussian mixture models:**
  - **<u>Expectation Maximization:</u>** For this approach the EM algorithm [110] was tested for both datasets with 4 and 7 clusters. For both cases the instances were distributed correctly among the 4 and 7 clusters. However, the log-likelihood, the parameter that measures the goodness of fit of a statistical model to a set of data calculated really low values. For the complete dataset the log-likelihood was -2597.93 for 4 clusters and -2530.353 for 7 clusters. For the smaller dataset it was -871.35 for 4 clusters and -834.24 for 7 clusters. The ideal case is that the log likelihood is as closer to 0 as possible and in both cases the results of the value is really low generating a bad fit of the statistical model and hence a bad clustering. This could be happening since the EM algorithm assumes that the attributes follow a Gaussian distribution and assumes independence between the attributes which is not the case for both datasets.

- **Density based clustering:**
  - **<u>DBSCAN:</u>** the DBSCAN algorithm tries to perform a clustering by establishing core, border and noise points based on the density and closeness of the data points [111]. It automatically determines the amount of clusters and also generates a noise clusters that is filled with irrelevant data points for the clustering process. However, this algorithm has a drawback. If the dataset consists of data points too close to each other the algorithm shows poor results. This was the case for both datasets. With the complete dataset, the algorithm obtained two clusters. One

cluster with 671 instances and a noise cluster with 45 instances. A similar situation was obtained for the smaller dataset obtaining 661 instances and a noise cluster with 55 instances concluding that DBSCAN is unable to obtain an appropriate clustering from both datasets.

- **Centroid based clustering:**

  K Means: Finally, the K Means clustering was tested for the centroid based approach. On this algorithm the idea is to find the smallest WSS (Within cluster Sum of Squared errors). Both datasets were tested with 7 and 4 clusters obtaining a good division of the instances. For the complete dataset, with 7 clusters the obtained WSS was 3467.8; and with 4 clusters the WSS was 3746.39. For the smaller dataset, with 7 clusters the obtained WSS was 198.94; and with 4 clusters, the obtained WSS was 213.456.

  This results showed that K Means was able to cluster both datasets and that the smaller dataset obtains a better clustering that the complete dataset. Furthermore, the difference between both WSS on the smaller dataset is insignificant. For this reason, the K Means clustering with 4 groups applied to the smaller dataset was selected as the dataset that will be implemented on the selection and comparison of IL algorithms.

Specifically, K Means clustering divided the samples into four classes. After analyzing the centroids, the distributions and calculating the total time and data occupation, each cluster was labeled as illustrated in Table 4.5 having in mind that data occupation is the most important feature. For this reason, the group of users that occupied more amount of data were labeled as higher consumption users.

Cluster	Total time occupation	Total data occupation	Number of users	Label
Cluster 1	136,534 hours	3,135 Mb	50 users	Medium consumption
Cluster 2	1275,462 hours	11,2 Mb	1 user	Very high consumption
Cluster 3	18,46 hours	2,242 Mb	582 users	Low consumption
Cluster 4	36,545 hours	3,482 Mb	83 users	High consumption

Table 4.5. Clusters labeling.

By implementing a correlation study between the 113 attributes and the target class, the most correlated attributes were determined in AppleIcloud data occupation with a correlation of 0.53251 and Gmail time occupation with a correlation of 0.41435. Although the applications are different and that several examples can be observed through different correlation plots, we focused on the mostly correlated to the target class for visualization purposes. Therefore, in Figure 4.10, a plot of these two attributes is illustrated to observe the cluster distribution. In general, it can be observed that the low consumption users maintain a low rate on both data and time occupations. Medium consumption users spend more time consuming while maintaining a low rate of data occupation. High consumption users present higher data occupations in shorter periods. Also, there is only one very high consumption user that exceeds all the other clusters. This user exhibits the highest data and time occupations. However, considering the clusters' behavior, this case could be treated as an anomaly on the data. Additionally, when analyzing the amount of users per cluster it is evident that there is a class imbalance.

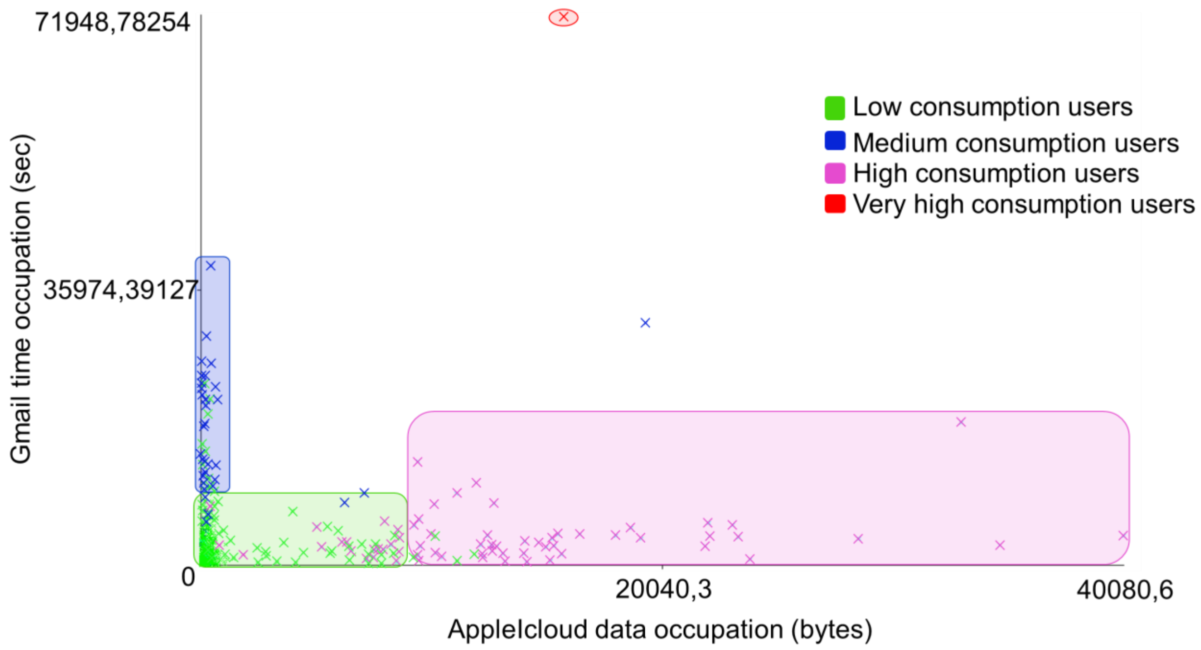


Figure 4.10. Clusters visualization.

Considering the previous statements, we performed the data preprocessing step to achieve two goals: (i) to remove anomalies and (ii) to improve the class balance. To achieve this, we applied the Inter Quartile Range (IQR) [127] approach that can detect and remove the anomalies in the dataset. To improve the class balance, we used the

SMOTE algorithm [128]. In consequence, the dataset was left with three clusters and the total number of instances (users) increased to 1249 after balancing the classes. Therefore, the distribution of users per cluster was left as follows:

- 510 low consumption users,
- 333 medium consumption users, and
- 406 high consumption users.

After all the first six steps of the Data Preprocessing and Model Selection component were completed, a dataset holding all the information of users' OTT consumption behavior was obtained. This dataset will be implemented to compare the performance behavior of IL algorithms and obtain a classification model capable of supporting the decision making process of network managers. This comparison will be presented on Chapter 5.

Summary

The major conclusion and contributions of this chapter are the datasets obtained with both the synthetic data generator and the initial processes of the reference model.

Specifically, this chapter presented a detailed description of all the processes carried out and the obtained results after applying the first two components of the proposed reference model. First on the data gathering and flow generation component a set PCAP files were gathered through capture sessions carried out through the configuration of a server on the network core of the Faculty of Electronics and Telecommunications Engineering of the Universidad del Cauca. Then all the captured packets were aggregated into flows through the development and implementation of an application named as Flow Labeler.

After all the flows were obtained, a cleaning process was carried out leaving only the flows that belonged to user devices. Subsequently, the user consumption estimation was carried out focusing on 56 OTT applications and obtaining two datasets: one dataset with a set of flow based statistics focused on the bidirectional attributes of the flows. On the other hand, by following the recommendation and knowledge domain of the network experts, the second dataset focused only on the time and data occupation of the users since these are the most related attributes to the consumption of OTT applications.

Once all the data related to the users' consumption was calculated, a clustering process was carried out aiming at obtaining a set of groups were users could be classified based on their consumption behavior. Several clustering approaches were applied to both datasets obtaining the best results with the KMeans algorithm (centroid based clustering) and the dataset focused on time and data occupation. Then, after all the analysis of the clusters and the step of data preprocessing, a dataset holding three clusters was obtained. The users were classified in low, medium and high consumption users. This dataset will be implemented to compare the performance of IL algorithms and obtain a classification model capable of supporting the decision making process of the network manager.

Chapter 5

Model comparison and selection

Up to this point, a dataset and synthetic data generators had been obtained in order to perform a comparison between algorithms. In this chapter, a conceptual overview between incremental learning and the traditional batch learning will be provided. Then, a brief definition of all the ML algorithms implemented in the performance comparison is presented. Finally, two performance comparisons are presented. The first comparison focuses on analyzing the performance of traditional batch learning algorithms against IL algorithms while using a dataset obtained with the synthetic data generators. Then, the second comparison analyzes the performance behavior of IL algorithms while using the dataset obtained with the reference model focusing on observing if these algorithms are capable of adapting to changes in the consumption behavior that the users may present over time.

5.1. The power of incremental learning

Current data are heterogeneous, generated at high speeds, and present in immense volumes. Their processing requires smart and sophisticated approaches. Consequently, the applicability of models suitable for real-time streaming data analysis that can maintain adaptiveness, efficiency, and scalability has recently been a focus of several research work [97]–[99], [129]. To better understand the advantages of such models, we first briefly discuss traditional ML.

5.1.1 Batch Learning

Traditional ML, also referred to as batch learning, is an application of Artificial Intelligence (AI) that is aimed at providing systems the ability to automatically learn a task from experience without being explicitly programmed for it. As presented in the CRISP-DM methodology, the ML lifecycle consists of repeatable processes and contextualizes the challenges faced by a data science team. In general, the ML lifecycle can be divided into four stages that present common challenges to be addressed [130]. The data science team must address these challenges to ensure the high quality of the models produced in the cycle. The description of this lifecycle is presented as follows.

- **Project design** focuses on selecting the appropriate methods to address the task at hand. The biggest challenge comes down to determining whether the problem indeed requires an ML solution. The activities used to help to determine this include output determination, target data identification, and model selection.
- **Data preparation** focuses on obtaining relevant data samples to fulfill the task. The typical activities helping to accomplish this task include data gathering, feature engineering, data cleaning, and data labeling.
- **Model fitting** is devoted to determining the best model in terms of performance for the task at hand. The activities considered to achieve this include model training and model evaluation. Depending on the definition of the desired output, model training can be performed in an unsupervised or supervised fashion. During a model evaluation, new and unseen data is provided to the algorithm to analyze if the algorithm has sufficient knowledge to perform the task correctly; this process is known as traditional batch learning.
- **Inference and deployment** phase deals with preparing the model to be leveraged for making viable predictions. Typical tasks associated with this phase include how the model will be deployed and how the results will be presented.

Traditional batch learning has been implemented widely and successfully in several areas like e-commerce, bioinformatics, robotics, and weather prediction. Its main advantages include minimum human interaction (i.e., automation), massive, multi-dimensional and heterogeneous data management, and unveiling hidden relationships among the data and correlations between features. However, it also presents some disadvantages. Firstly, depending on the algorithm and the size of the dataset, the training phase might take a considerable amount of time and demand high computational resources. Secondly, when training with a small dataset with no representative number of samples, the algorithm might present a poor performance.

Lastly, once an algorithm has been trained and implemented, it cannot acquire new knowledge from new samples. In consequence, if there is a change in the statistical properties of the data (i.e., a concept drift), a new model must be generated. IL seems to be a suitable alternative to overcome these limitations [131].

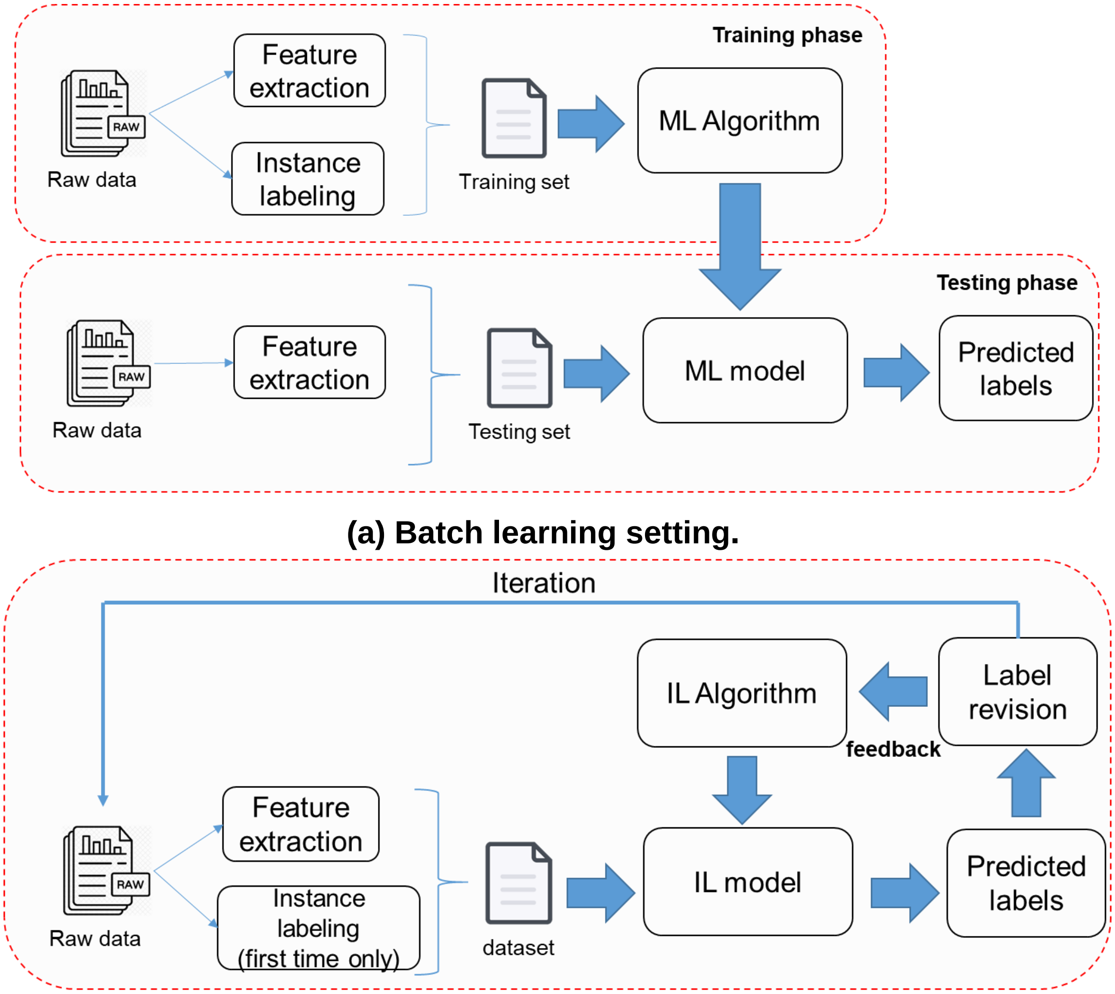
5.1.2 Incremental Learning

IL is built on the premise of continuous learning and adaptation, enabling the autonomous incremental development of complex skills and knowledge. In ML context, it aims to smoothly update the prediction model to account for different tasks and data distributions while still being able to re-use and retain knowledge over time. Figure 5.1 presents a comparison between a traditional batch learning and IL setting in model generation [132].

The main difference between both settings is the fact that IL provides a continuous iterative process. In contrast, the traditional batch setting is divided into two phases, training and testing, that must be repeated every time a model update is necessary. IL can build its prediction capabilities on top of what was previously learned, amending previous errors and shortcomings while efficiently adapting to new environmental conditions as new data becomes available. This way, the systematical re-trainings can be avoided, saving time and computational resources while maintaining the model's performance. Therefore, IL is preferred whenever systems need to act autonomously, such as in interactive scenarios where data can be continuously provided [8], [9].

As mentioned before, an IL algorithm can be defined as one that meets the following criteria [33]:

- **Adaptiveness:** having in mind that most real-life situations and phenomena cannot be limited to a specific number of unchanging variables that can be processed deterministically, it is imperative to keep adapting. The IL algorithm must guarantee adaptation capabilities through a fast, efficient, and flexible learning process. Maintaining knowledge acquisition with every new data input and learning new class labels or features when needed.



(a) Batch learning setting.

(b) Incremental learning setting.

Figure 5.1. Comparison of learning settings.

- **Scalability:** the input data must be processed only once to ensure that the algorithm can scale in terms of intelligence through the processing of more and more data while maintaining computational memory fixed or on sustainable terms. Storing large amounts of data with high-dimensionality would be impossible to maintain while aiming to process for a long time scale. This way, the IL algorithm is more similar to an actual brain that filters partial data and retains only the most essential information.
- **Efficiency:** as the process is continuous and the amount of data keeps increasing, the algorithm should not start from scratch every time. This way, there is no need for large amounts of computational resources to handle data.

IL appears to be a suitable alternative to replace traditional ML approaches, offering a solution to address their limitations. IL can keep learning over time and maintain steady performance for the task. A typical application area of IL is big data processing. In this field, there exist an increasing interest in low memory consumption models that do not require substantial computational resources for storing large datasets for the training phase. In the robotics area, the data arrives as a stream of signals with high possibilities of changes in their structure making IL an ideal candidate solution to handle those changes. Finally, image processing becomes an ideal scenario for IL since the data is often gathered in a streaming fashion [8].

It is important to emphasize that IL also has some drawbacks such as i) it is not completely clear yet how to evaluate IL techniques. ii) it is not clear how to behave after the capability of the model is saturated, neither iii) how to selectively forget previous knowledge (stability-plasticity dilemma) [8], [133], [134]. Also, as it is a relatively new paradigm, there is only a limited number of works that are mainly aimed at evaluating its efficacy in clustering. Undoubtedly, there is room for further research and improvement. However, IL is expected to have a decisive role in the future, and it is becoming more and more popular by outperforming traditional batch learning approaches.

Now, since there is a common understanding between the concepts and learning settings of both approaches, we will proceed with the presentation of a general overview of the algorithms that are considered in the performance comparisons.

5.1.3 Algorithms overview

This section presents a brief definition of the algorithms implemented in all the experimental scenarios. First the definition of the algorithms belonging to the traditional batch learning are presented. Subsequently all the IL algorithms are defined.

5.1.3.1 Batch learning algorithms

Remembering that this project is the subsequent step of a previous masters' thesis [16], the batch learning algorithms selected for the current experimental scenarios were the three algorithms that exhibited the best performance on the masters' thesis tests. These algorithms are presented as follows:

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 [135]. 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 ML meta-algorithm formulated by Yoav Freund and Robert Schapire [136], which 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 [135].

Support Vector Machines – SVM

In ML, 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 [137]. 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 the Sequential Minimal Optimization (SMO) algorithm was implemented [138].

K Nearest Neighbor - KNN

In pattern recognition, the KNN algorithm is a non-parametric method used for classification and regression [139]. In both cases, the input consists of the K closest training examples in the feature space. The output depends on whether KNN is used for classification or regression:

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

5.1.3.2 Incremental learning algorithms

The most commonly used IL algorithms include decision trees, rule-based systems, NB, KNN, SVM, and neural networks [33], [140]–[144]. Further common alternatives represent ensemble methods such as Oza Bagging (OB), Adaptive Random Forest (ARF), Leverage Bagging (LB), and Learn++. We evaluated the Python scikit-multiflow [145] implementation of these algorithms. The main characteristics of the algorithms are shown in Table 3 as per [146]. Furthermore, their brief description is as follows.

Feature/algorithm	*Decision trees (VFDT, ARF)*	*SVM*	*Neural Networks (MLP)*	*Lazy methods (KNN)*	*Ensemble methods (OB, LB, Learn++)*	*Naïve Bayes*
Type of input parameters	Numeric					
Parameters considered for the classification	Selection of the most discriminant attribute	The zone of influence in the hyperplane border	The rule combination of classifiers	The number of nearest neighbors	Classification is determined by a weighted voting system	Estimated probabilities for each target class
Type of classification	Strict					
Advantages	Simplicity in comprehension and interpretation	Strong mathematical foundations	Resistance to noise in the data	Versatile algorithm for classification and regression	Allows the composition of classifiers improving their individual performance	Fast and simple classification based on a statistical approach
Disadvantages	Depending on the number of attributes its representation can be difficult	Large calculation time of hyperplanes	The absence of a systematic method for its implementation	Prediction might be slow if the number of neighbors is large	Depending on the composition the prediction time might be large.	Naïve assumption of correlation independence between input features

Table 5.1. General overview of IL algorithms [146].

Hoeffding Tree

The Hoeffding Tree (HT) or Very Fast Decision Tree (VFDT) [147] is an incremental decision tree induction algorithm that exploits the fact that a small sample of data can often be enough to choose an optimal splitting attribute for the construction of the decision tree. This idea is supported mathematically by the Hoeffding bound, which gives a level of confidence that allows choosing the best attribute to split the tree [148].

Incremental Naïve Bayes

The incremental adaptation of the NB algorithm is a classifier that is known for its simplicity and low computational cost. This algorithm performs the classification process while applying the Bayes theorem, where a strong (naive) assumption is made (all the features in the input data are independent of each other).

Incremental K Nearest Neighbor

The KNN algorithm, in an incremental setting, works by keeping track of a fixed number of training samples of the last window of observed samples. Then, whenever new input data is received, the algorithm searches within these stored samples and finds the closest neighbors using a selected distance metric (e.g., Euclidean distance). To store the samples, while maintaining low search times, scikit-multiflow uses a structure called a K Dimensional Tree.

Oza Bagging

OB [149] is an incremental ensemble learning method that improves the traditional Bagging from the batch setting. It simulates the training phase by taking each arriving sample to train the base estimator over K times. This sample is drawn by a binomial distribution -- a method that can be considered as "a good drawing with replacement" substitution from the one implemented in the traditional batch learning.

Leverage Bagging

LB [150] is based on the OB algorithm. It tries to obtain better results by modifying the Poisson distribution parameters obtained from the binomial distribution when assuming an infinite input data stream. This Poisson distribution is used to perform the drawing with replacement to reduce the number of zero values in the distribution's mass probability function. This is achieved by increasing the value of λ from 1 to 6.

Adaptive Random Forest

ARF [151] is an adaptation of the traditional random forest algorithm applied to the incremental learning scope. ARF generates multiple decision trees and decides how to classify the input data through a weighted voting system. Within the voting system, the individual tree that has the best performance (in terms of accuracy or the Kappa statistic) has a more substantial weight in the votes, i.e., higher priority in the decision.

Learn++

Learn++ [152] is an ensemble method inspired by the Adaptive Boosting (AdaBoost) algorithm, originally developed to improve the classification performance of weak classifiers. In essence, Learn++ generates an ensemble of weak classifiers, each trained using a different distribution of training samples. The outputs of these classifiers are then combined using a majority-voting scheme to obtain the final classification.

MultiLayer Perceptron - MLP

Finally, the MLP algorithm [153], [154], is a kind of neural network that has three or more layers. It is used to classify data that cannot be separated linearly. It is a type of artificial neural network that is fully connected. Every single node in a layer is connected to each node in the following layer and it uses a nonlinear activation function (mainly hyperbolic tangent or logistic function).

5.2. Batch learning vs incremental learning

With the aim of comparing the performance of traditional and incremental ML algorithms in order to conclude if there was a real difference in the performance of both approaches when applied to the same dataset, a set of three test scenarios where defined for comparing 3 traditional ML algorithms and 9 IL algorithms. All of the algorithms were tested through the dataset obtained with the synthetic data generators, which holds information about the users' OTT consumption behavior implemented in the masters' thesis that was used to test the batch learning algorithms. The experimental scenarios that were defined will be explained as follows.

5.2.1 Experimental scenarios

This subsection describes the three test scenarios that were defined in order to compare the performance of traditional ML algorithms with IL algorithms. This considers that users can change their consumption behavior over time, that the market that involves the OTT applications is highly volatile and that, although the KDN paradigm [2] proposed the introduction of AI techniques such as machine learning to network management, it does not consider the need of machine learning models that maintain their usefulness over time. Figure 5.2 illustrates the different test scenarios. In order to understand the experimental process, it is important to mention the following remarks

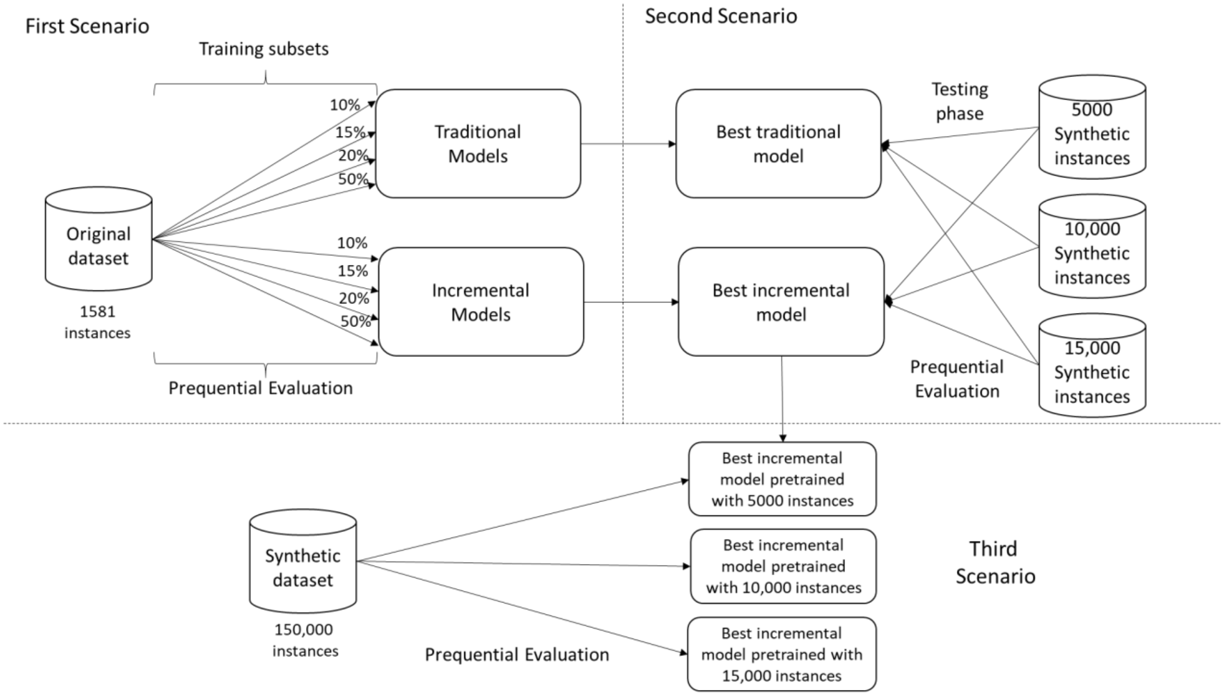


Figure 5.2. Experimental scenarios Batch learning vs Incremental Learning.

- The traditional models were trained and tested using a percentage split [155] configuration on the first scenario. Therefore, different subsets of the original dataset were used to train the models and the remaining subsets were used to test their performance (the training and testing datasets are different).
- From the second scenario onwards, only the performance of the best traditional model obtained from the first scenario was tested (i.e., no additional training phases were performed). Hence, the model received datasets apart from the one used in the training phase (the synthetic dataset), in order to be tested.
- The incremental models were tested using a prequential evaluation [156] or interleaved test-then-train configuration. Such evaluation is an alternative to the traditional holdout evaluation, inherited from batch setting problems. This method consists of using each sample to test the model, which means to make

predictions, and then the same sample is used to train the model. This way the model is always tested on samples that it hasn’t seen yet.

The tests scenarios will be described as follows:

5.2.1.1 First test scenario

On this scenario, the aim was to identify if the IL models were capable of obtaining a similar or better performance with a fewer quantity of instances on the training phase, by comparing them with the traditional models using the original dataset with 1,581 instances. The training phase of the models was carried out using different sizes of the dataset. Specifically, 10%, 15%, 20% and 50%. The batch learning algorithms implemented were: Adaboost with J48 decision tree as base classifier, KNN with 30 neighbors (such number of neighbors was identified using a cross validation approach) and the SMO (Sequential Minimal Optimization) algorithm. All the tests performed for the batch learning algorithms were performed using Weka. On the other hand, the IL algorithms that were trained and tested were: Hoeffding tree, Naive Bayes, Adaptive Random Forest (ARF), Leverage Bagging using KNN with 8 neighbors as base classifier, Leverage Bagging using ARF as base classifier, Perceptron mask (neural network), Oza Bagging using KNN with 8 neighbors as base classifier, Oza bagging using a Hoeffding tree as base classifier, and KNN with 5 neighbors. The tests using the IL algorithms were performed using Python’s library Scikit-multiflow.

5.2.1.2 Second test scenario

On this scenario, the aim was to compare the performance behavior of the best models of each approach, obtained from the first test scenario, with a subset of instances taken from the synthetic dataset. Such analysis was carried out by observing two aspects: first of all, if the best incremental model was able to maintain a good performance while adapting to the new data. Second, if the best traditional model was able to exhibit a good performance against the new data using only the knowledge obtained from the training of the first scenario. To perform these tests three subsets of 5,000, 10,000 and 15,000 instances were taken from the synthetic dataset and used individually to evaluate both models.

5.2.1.3 Third test scenario

On this last scenario, the aim was to observe how the best incremental learning model, obtained from the first scenario, would behave in terms of performance when receiving all of the instances from the synthetic dataset (150,000 instances) while having a ‘‘warm-up’’ or pre-train with the different subsets from the second scenario. All of this in order to observe if the incremental model reaches a maximum point in terms of performance or when receiving a high number of instances its performance begins to be affected in a negative way.

5.2.2 Results analysis

This section presents the results and analysis obtained on each of the test scenarios focusing on comparing the precision or accuracy, recall, kappa statistic and confusion matrix in order to determine which model achieves better performance.

5.2.2.1 Results - first scenario

As was mentioned before, on this scenario the training was performed with different subsets taken from the original dataset, and a total of 12 algorithms were tested (3 batch learning algorithms and 9 incremental algorithms). Figure 5.3 and Figure 5.4 present the obtained results for the traditional and incremental models (Figures (a) and (c) present the results for the Batch learning algorithms while figures (b) and (d) present the IL algorithms) with a training phase using 10%, 15%, 20% and 50% of the original dataset. In general, all three batch learning algorithms exhibit a good performance, with SMO being the best by a small difference.

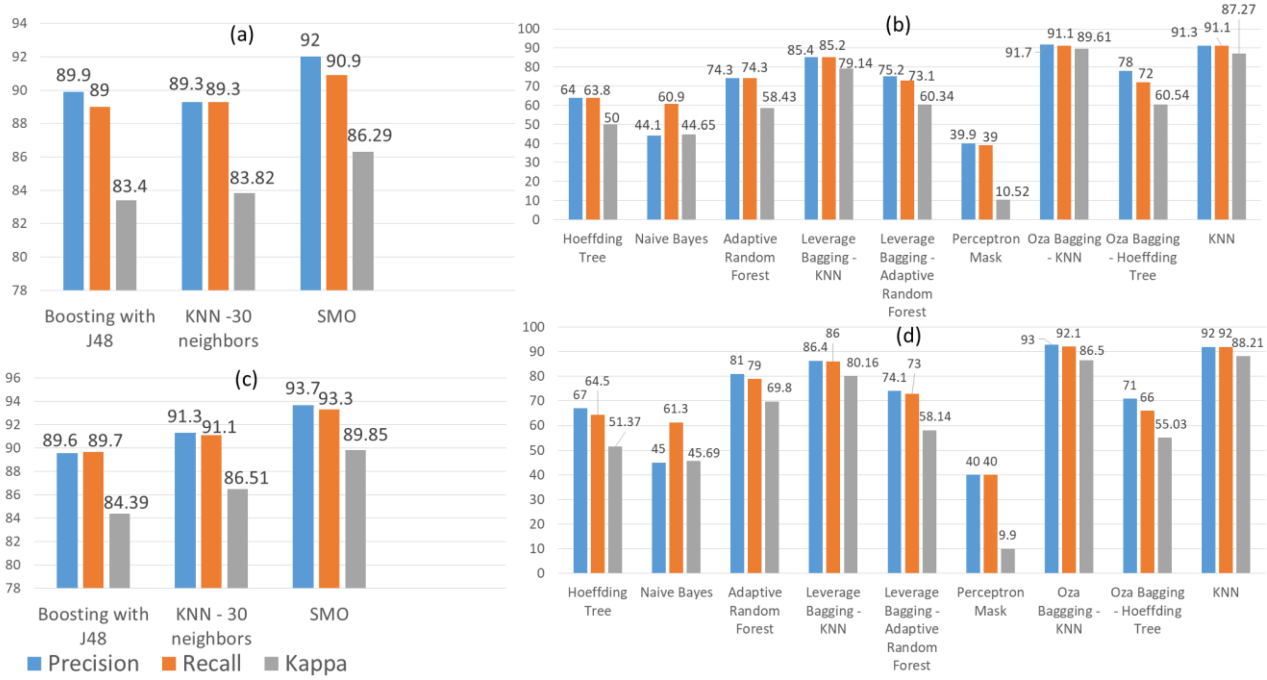


Figure 5.3. Results – training with 10% and 15%.

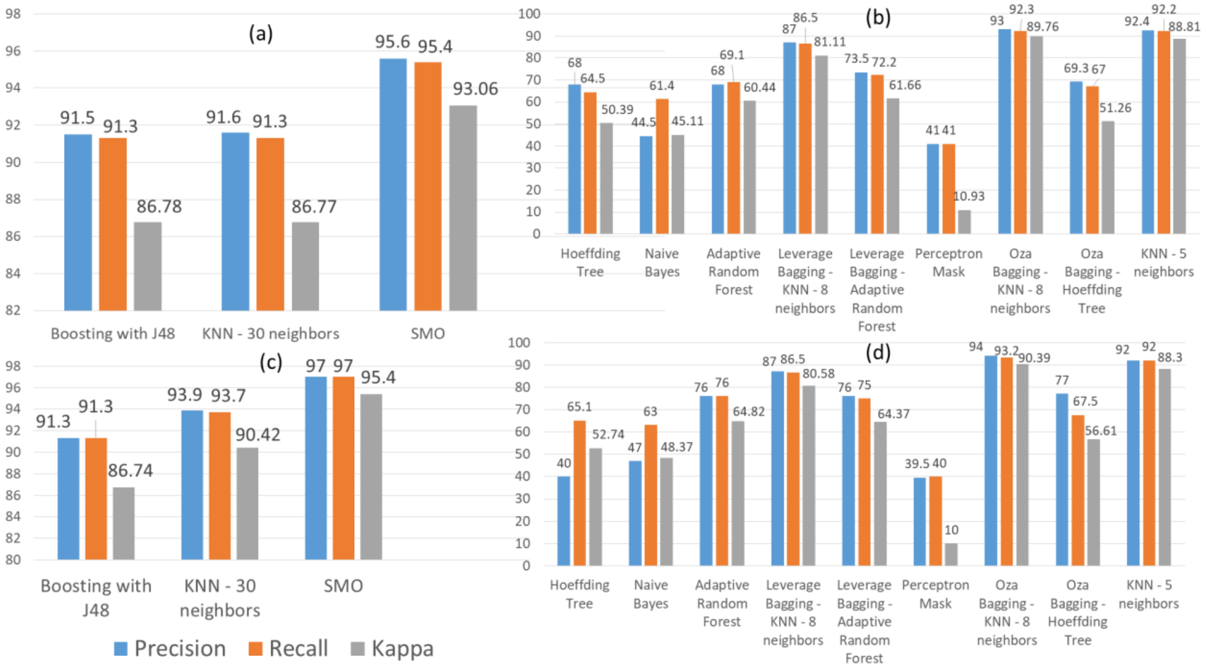


Figure 5.4. Results – training with 20% and 50%.

After analyzing all the results, it can be observed that all the Batch learning algorithms (Boosting with J48, KNN with 30 neighbors and SMO) exhibit a good overall performance, highlighting the fact that SMO is the best with all the training subsets. On the other hand, the incremental algorithms present a more volatile behavior, where, unexpectedly, the Perceptron Mask shows the worst behavior on all cases. The KNN algorithm and the composition of Oza Bagging with KNN are the best incremental classifiers on all cases. Furthermore, it is important to mention that for this scenario, where the training and testing is done using the same dataset, as expected, the Batch learning models keep improving their performance when the training set has more instances, with the SMO algorithm having the best overall performance when the training set holds 50% of the instances. However, the Oza Bagging with KNN and the KNN algorithm also exhibit a really similar performance to the SMO algorithm on all the cases. Having this in mind, the model selected as the best classifier from the incremental learning approach on this scenario is the composition of Oza Bagging with KNN trained with the subset holding 10% instances from the original dataset, since the difference with the other cases is not significant in terms of performance and this case required less computational resources in order to be tested.

For the time being, with the results observed on this first scenario it can be concluded that, while using the same dataset for training and testing, there is no major difference in terms of performance between some algorithms from both approaches. However, the incremental models hold an important advantage for a network manager considering that their training and testing is performed in a more efficient way than the batch learning models, requiring less time and computational resources.

5.2.2.1 Results - second scenario

As was presented in a previous section, this scenario aims at comparing the performance behavior of the best algorithms from each learning approach obtained on the first scenario while using subsets from the generated synthetic dataset. Figure 5.5 illustrates the obtained results. Table 5.2 and Table 5.3 shows the confusion matrix for both algorithms in the case tested with 15,000 instances. The following conclusions are stated.

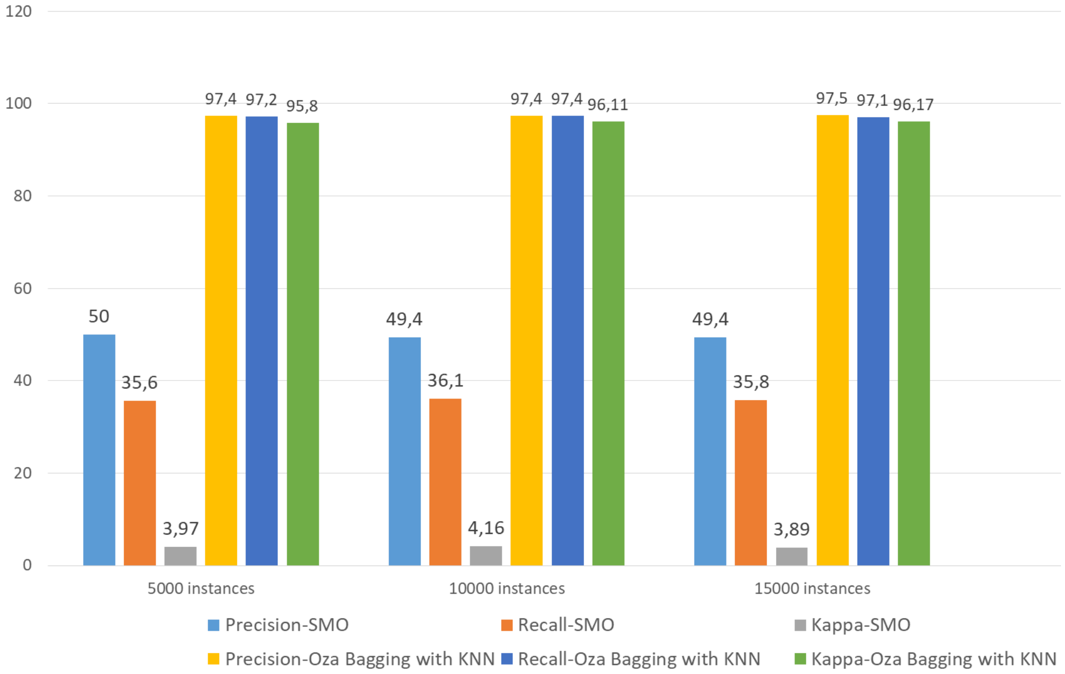


Figure 5.5. Results – Second scenario.

- The incremental model (Oza Bagging with KNN) not only was able to maintain a good performance with new (synthetic) data, but showed a better performance with an important difference in all cases (more than 40% in all of the performance metrics). This is because the model is capable of analyzing the incoming data streams, keep learning on new data and adapt its knowledge.

SMO 50% - 15000 inst			
-	High Consumption	Medium Consumption	Low Consumption
High Consumption	4932	2	0
Medium Consumption	5037	24	0
Low Consumption	260	4338	407

Table 5.2. Confusion matrix - SMO trained with 50% - 15000 instances.

Oza Bagging - KNN - 8 neighbors - 15000 inst			
-	High Consumption	Medium Consumption	Low Consumption
High Consumption	4672	261	1
Medium Consumption	36	4944	81
Low Consumption	3	1	5001

Table 5.3. Confusion matrix - Oza Bagging with KNN trained with 10% - 15000 instances.

- The batch learning model (SMO) is able to identify only two classes out of the three that exists on the dataset. Such behavior can be happening considering that the differences between the users classified on the medium and high consumption profiles are very subtle and the knowledge acquired by the traditional model in the training phase is insufficient for the classification task. Therefore, since it is not learning anymore, it is incapable of separating these instances and ends up gathering them on a single group. This can be observed in the confusion matrix (Table 5.2) where most of the medium consumption instances are being classified as high consumption and most of the instances from the low consumption class are being classified as medium consumption.
- The batch learning model (SMO) was not able to exhibit a good overall performance when facing the new (synthetic) data. This is because the new data has meaningful changes and the knowledge obtained by this model in the training phases is not sufficient to properly process them. The synthetic data was generated using the same statistical distribution as in the real data used in the training phase, assuming statistical independence; however, this independence is not entirely true, and therefore this assumption introduced

changes in the dataset that the traditional model was not able to handle with its current knowledge.

The previous statements allow us to conclude that, when we consider the volatility of the Internet and OTT applications, the IL approach is a suitable option when dealing with possible changes that users may present in their OTT consumption behavior over time i.e., since the attributes that define the user consumption profile (Number of generated flows, time of consumption and required network resources) can present multiple changes on a time period, the IL approach represents an important advantage for network managers, since it overcomes the weakness that a batch learning model presents due to their incapability of adapting to new data without a new training process.

5.2.2.1 Results - third scenario

As previously stated, the aim for this scenario was to observe if the best incremental model reaches a maximum point in terms of performance or if it is affected in a negative manner, when receiving a high number of instances obtained from the synthetic dataset (150,000 instances). This is considered since a usual inconvenience of the adaptability of incremental learning models is that they might “forget” important information about the data when adapting to incoming data streams, decreasing their performance in the classification. Figure 5.6 and Figure 5.7 illustrates the obtained results for this scenario.

By analyzing the obtained results two conclusions are clear: first, the number of instances used as “warm-up” for the model are irrelevant since the performance is almost identical in all cases. Second, the incremental model maintains a good overall performance after being tested with 150,000 synthetic instances without showing any negative decline in the accuracy or kappa statistic.

From this scenarios, it is possible to conclude that the IL approach surpasses the overall performance of batch learning models when the training and testing datasets are different. On the other hand, both approaches show a similar performance when the training and testing sets are the same. Also, the composition of Oza Bagging with KNN seems to be a suitable IL approach when dealing with the possible changes that users may present in their OTT consumption behavior over time, however, this has to be tested once again since the independence assumption in the dataset created with the synthetic data generators does not reflect completely a real consumption behavior presented by users. This will be presented in the following section.

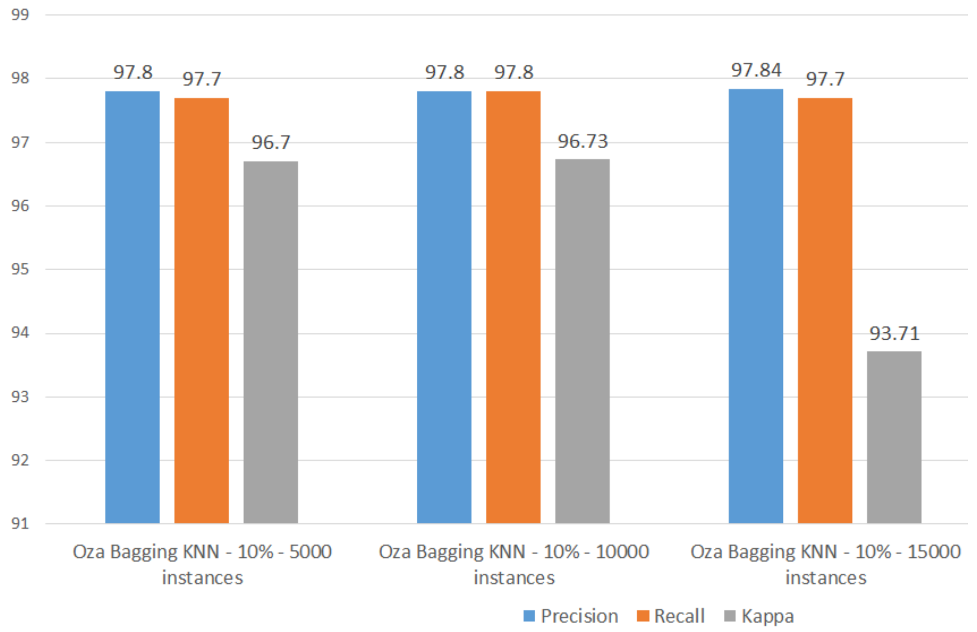
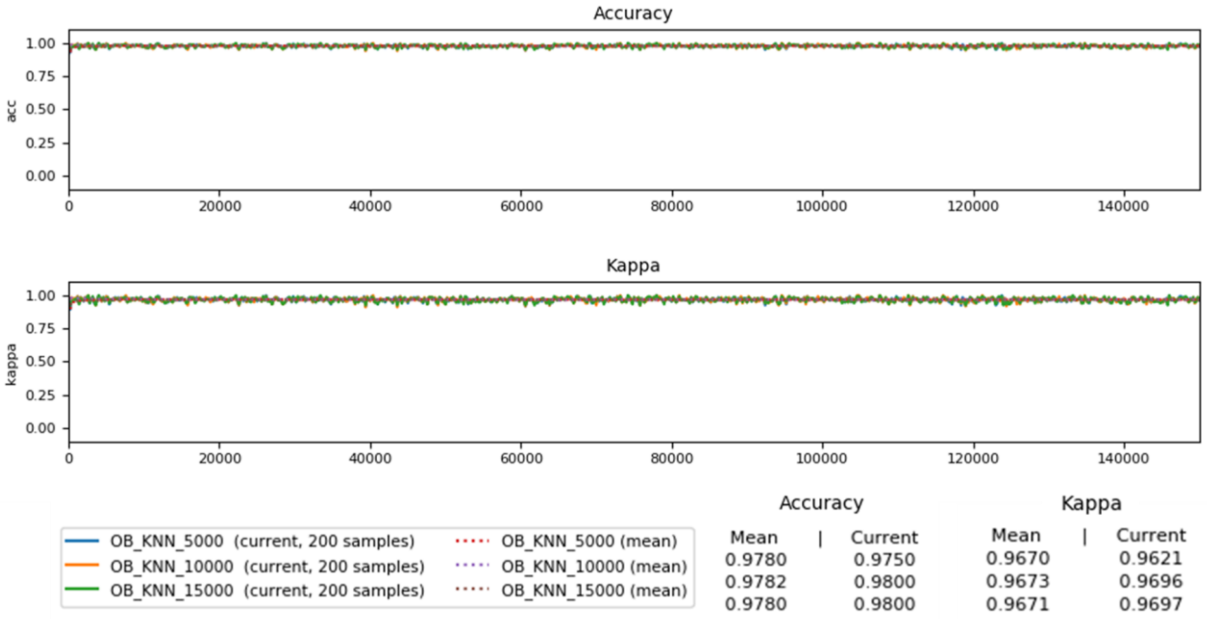


Figure 5.6. Results – Third scenario.



Accuracy		Kappa	
Mean	Current	Mean	Current
0.9780	0.9750	0.9670	0.9621
0.9782	0.9800	0.9673	0.9696
0.9780	0.9800	0.9671	0.9697

Figure 5.7. Accuracy and Kappa evolution – Third scenario.

5.3. Incremental learning comparison

Due to the fact that the synthetic data does not reflect completely the real behavior of users when consuming OTT applications, it was still necessary to compare the performance behavior from IL algorithms when faced with real data. For this reason, the dataset obtained after applying the reference model, as described on Chapter 4, was implemented in a set of tests aim at observing which IL algorithm could classify users behavior while maintaining a consistent performance behavior and support the decision making process of a network manager.

Having in mind the different evaluation approaches that can be considered for IL [114], similar to the previous case, we focused on analyzing the precision or accuracy and learning rate of the algorithms when performing classification. These measurements include precision, recall, and the Kappa statistic [157], [158]. For the comparison, two experimental scenarios were defined and will be described as follows.

5.3.1 First scenario

In this scenario the aim was to identify the best IL algorithms in terms of performance. As in the previous comparison, Figure 5.8 illustrates the set of prequential evaluations that were carried out for each algorithm changing the warm-up data sample size between 10% (125 instances), 15% (187 instances), 20% (250 instances), and 50% (625 instances) of the original dataset. The algorithms selected for the comparison were Naïve Bayes, KNN, HT, ARF, LB-KNN, LB-ARF, OB-KNN, OB-HT, Learn++, and MLP.

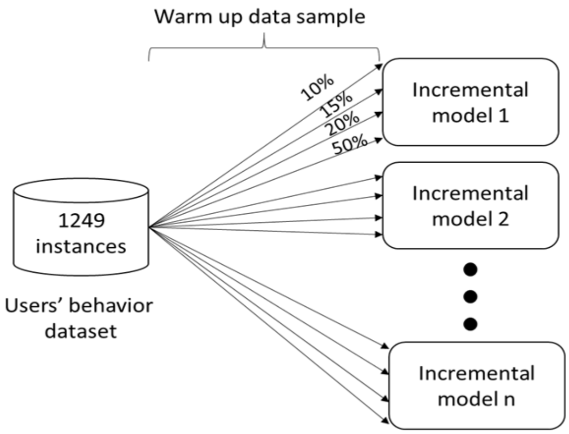


Figure 5.8. IL comparison – First Scenario.

For the ensemble methods (OB, LB, Learn++, and ARF), ten estimators were selected [159]. Figures 5.9, 5.10, 5.11 and 5.12 depict the obtained results for the algorithms tested in terms of precision, recall, and the Kappa statistic for all warm-up sizes.

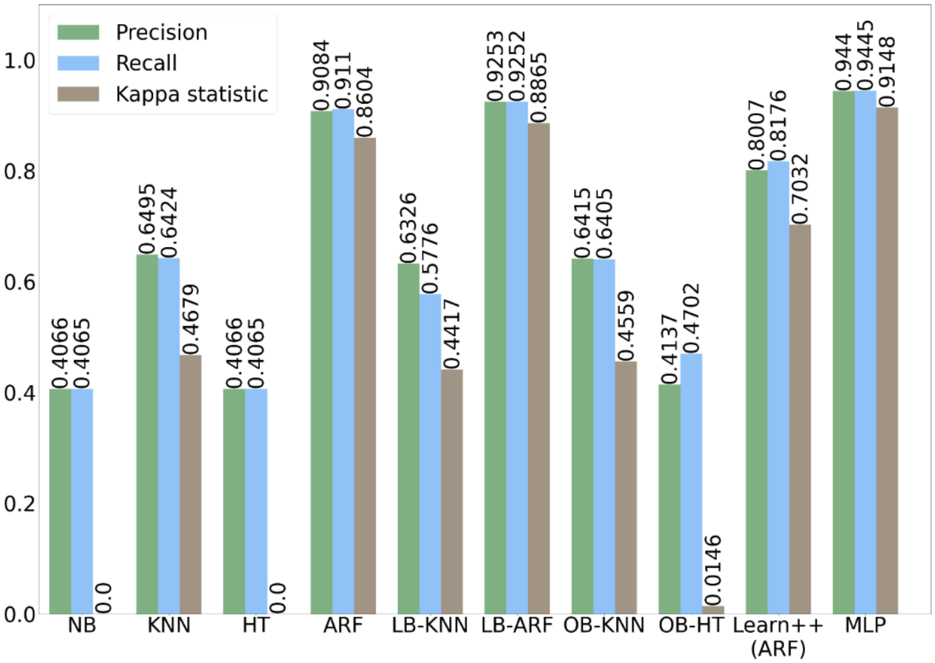


Figure 5.9. Results - Warm-up with 10%.

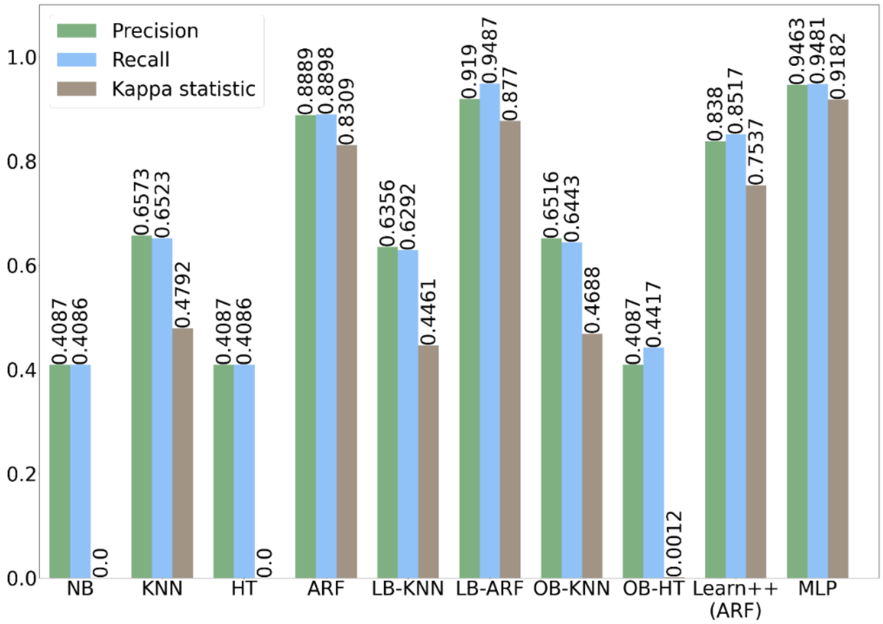


Figure 5.10. Results - Warm-up with 15%.

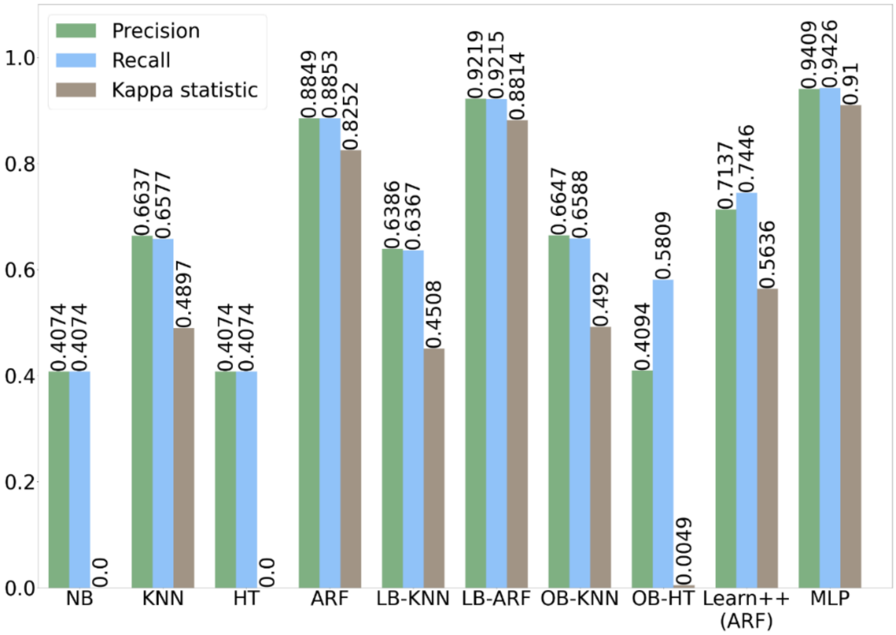


Figure 5.11. Results - Warm-up with 15%.

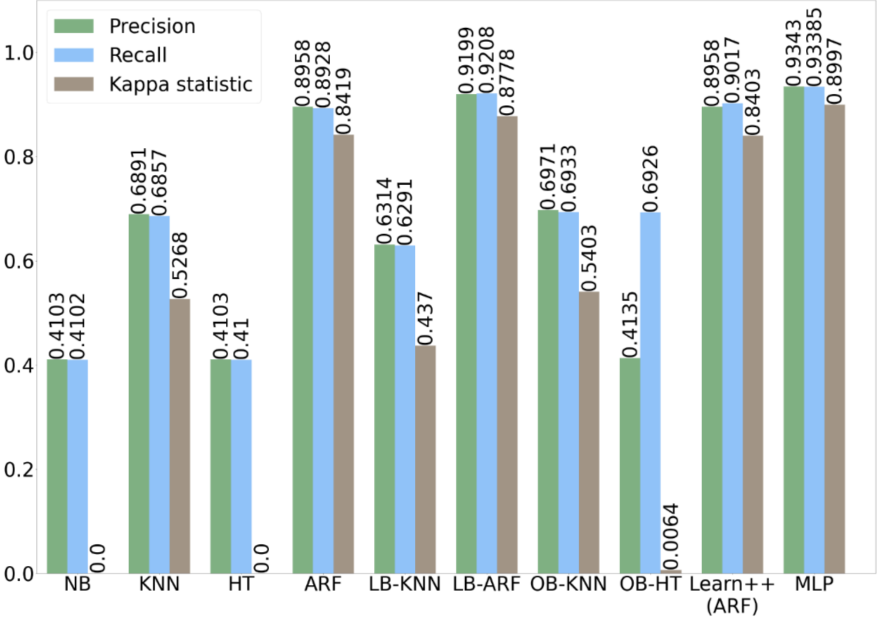


Figure 5.12. Results - Warm-up with 15%.

From the results, several conclusions can be derived. In general, in all tests, we observed similar performance behaviors. We noticed that the NB, HT, KNN algorithm and the ensemble methods that used them as base estimators in the composition, achieved the worst performance. For NB, such performance could be provoked due to its (naive) assumption, where statistical independence between the attributes of the input data is assumed. Since the time and data consumption of the same OTT applications are related, assuming all the attributes are independent could lead to an increase in the error rate throughout the classification process.

For the classification process, the KNN algorithm maintains a small window of previously seen instances (previous perception data) and looks among them for the nearest neighbors to the current input. In consequence, the data within the stored window likely does not provide a correct classification for the input. It can also be the case for the ensemble methods that use KNN as the base estimator (LB and OB). The ensemble methods are composed of 10 KNN estimators, while each estimator stores a data window. Therefore, the performance of the ensemble method becomes compromised if such a window does not provide a correct classification on each estimator, selecting a wrong classification for the input into the voting process.

Also, the HT algorithm and the OB with the HT as a base estimator exhibited poor performance. This algorithm builds a decision tree while exploiting the following assumptions:

i) A small sample can often be enough to choose an optimal splitting attribute.

ii) The distribution that generates the data does not change over time.

These assumptions make the algorithm efficient in terms of computational resources. However, in some instances, they can be a disadvantage as once a node is created in the tree, it cannot be changed anymore [160]. Therefore, where the users present different consumption behaviors, and the tree has a warm-up with a small sample of the dataset, the performance might not achieve the best results if the nodes built during the warm-up are unable to correctly classify the subsequent data inputs.

On the other hand, we observed that four algorithms (ARF, LB-ARF, Learn++ with ARF, and MLP) achieved an excellent overall performance. The excellent performance of ARF is likely due to the multiple decision trees that this approach generates, and the weighted voting system used for classification. Furthermore, this algorithm also has a warning- and drift-detection method. The warning-detection method tries to detect when possible concept drift occurs in the target class. For the other method, if drift is detected, the algorithm starts training another decision tree besides building the main ensemble method composition. Once the drift-detection method confirms that there is a concept drift in one of the individual decision trees, it is replaced with the tree trained in the background to keep the classification accuracy. It is also important to mention that the ARF algorithm was used as a base estimator for Learn++ and LB to observe if the excellent performance obtained by this ARF individually could be further improved by combining it with ensemble methods. Such improvement was achieved with LB, however, not with Learn++.

Finally, MLP also achieved an excellent performance, especially after performing several experimental setups to help find the optimal network structure and an activation function. We varied the number of hidden layers, the number of nodes, and the activation function as per [161]. The best results were obtained with a 5-layer network (including the input and output layers). It means we could ensure a fast warm-up stage and low computational resource use. The number of nodes on each layer was set between 1 and 113. The best results were obtained with 113 nodes for the first hidden layer and 40 nodes for the two subsequent layers (113x40x40). As an activation function, we considered ReLU, hyperbolic tangent, sigmoid, and identity functions. The best results were achieved using the hyperbolic tangent function. Finally, we selected Adam as the optimization algorithm that was shown to be an efficient extension to stochastic gradient descent [162].

In conclusion, on this scenario, the best results are achieved using the ARF, LB-ARF, and MLP algorithms.

5.3.2 Second scenario

Considering the results on the previous scenario, ARF, LB-ARF and MLP algorithms appeared to be suitable for user OTT application consumption behavior classification. However, the transfer of these approaches into real-world applications also depends on how well they maintain their accuracy when changes in users’ consumption behavior occurs. Specifically, we assessed how well the candidate algorithms maintain their classification performance and how their learning rates are affected when a change in users’ OTT consumption behavior occurs. In doing so, the algorithms were first pre-trained using the entire dataset. Then, we investigated their performance maintenance through a prequential evaluation by systematically modifying 60 user behaviors. This was to simulate a real-world implementation use case where user behavior changes are very likely to happen over time. The changes that we introduced are as follows:

- 20 Low Consumption users had their behavior changed (10 users to Medium Consumption and 10 to High Consumption),
- 20 Medium consumption behaviors were changed (10 users to Low Consumption and 10 to High Consumption), and
- 20 High Consumption users had their behaviors changed (10 users to Low Consumption and 10 users to Medium Consumption).

Figure 5.13 depicts the setting for this evaluation. The obtained results are shown in Figure 5.14, while the precision evolution of each algorithm is depicted in Figure 5.15.

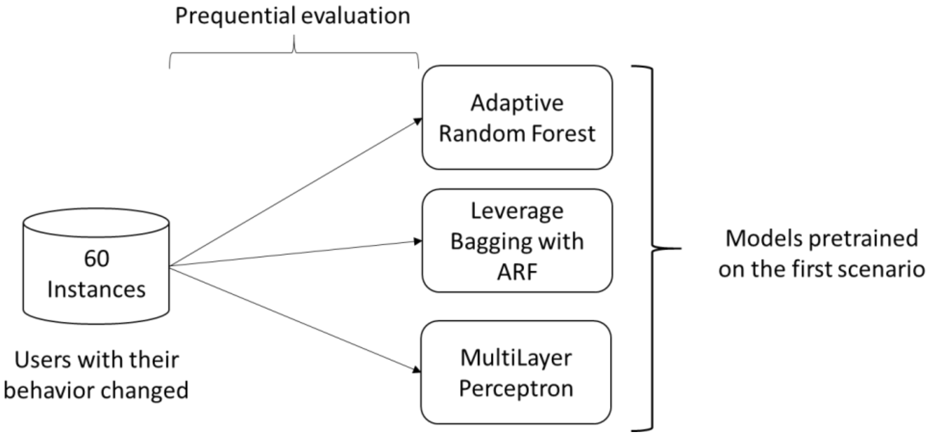

Prequential evaluation
60 Instances
Users with their behavior changed
Adaptive Random Forest
Leverage Bagging with ARF
MultiLayer Perceptron
Models pretrained on the first scenario

Figure 5.13. Classification performance maintenance setting.

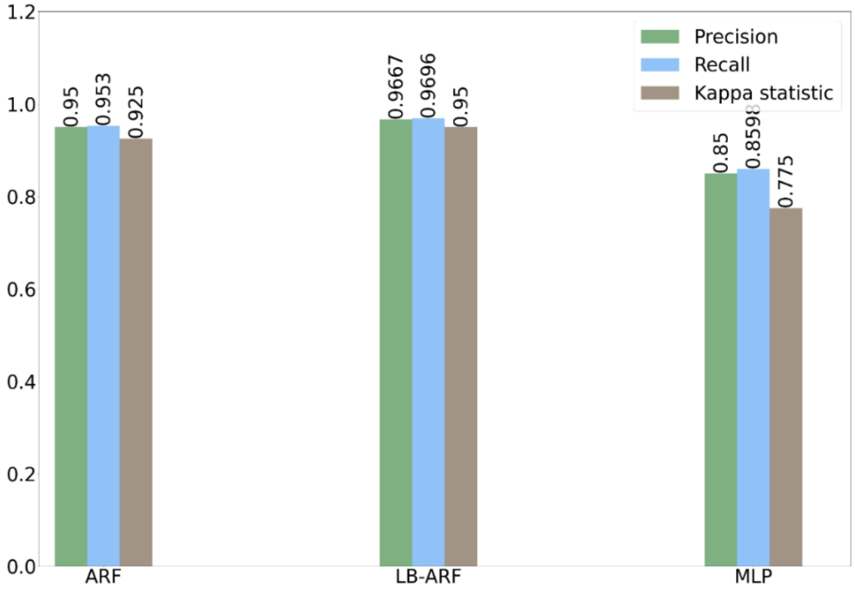


Figure 5.14. Precision, recall, and Kappa results maintenance with changing samples.

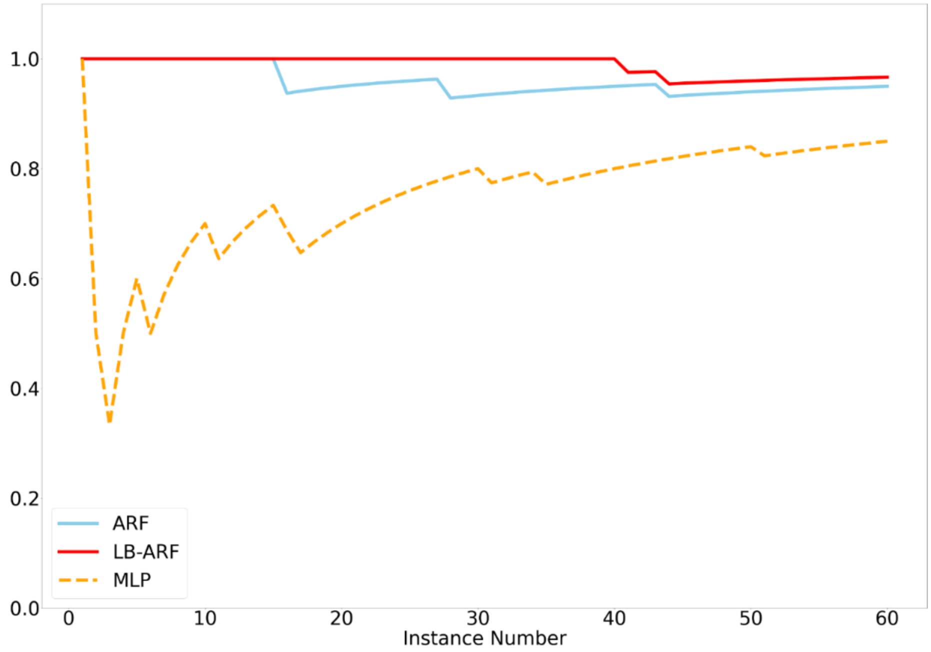


Figure 5.15. Precision evolution.

From Figure 5.14 and Figure 5.15, we can observe that all the three algorithms presented an excellent overall performance. However, as Figure 5.15 shows, MLP is affected considerably by the changes introduced in the users' behavior at the beginning of the data stream. While the results eventually improve, the adaptation to the changes is slow. On the other hand, both ARF and LB with ARF can maintain their performance throughout the entire experiment.

Considering the above mentioned, now we can conclude that both ARF and the composition between LB and ARF are IL algorithms suitable for performing user OTT application consumption behavior classification since on this case the tests are based on data from a real scenario. More importantly, these algorithms are capable of providing useful information, even in the event of dynamic changes on users’ behavior. Network administrators could make good use of such an approach as IL can make network traffic-related managerial tasks faster as well as more robust and resource-efficient.

With this in mind, Chapter 6 will discuss the decision making stage, as observed on the reference model, focusing on a proposal of personalized service degradation policies.

Summary

The major contribution of this chapter are the experimental results obtained through the performance comparison between the batch learning approach and the IL approach executed on the datasets previously built. All of the tests concluded in the fact that the IL approach is indeed capable of adapting to the changes that the user presents on his OTT consumption behavior while the batch learning approach is affected negatively on its performance when this changes occur.

Specifically, this chapter is divided in three sections. The first section focused on offering a conceptual comparison of IL and the traditional batch learning setting while discussing the advantages and weaknesses of both approaches. Then a brief overview of the algorithms that were implemented on the experimental scenarios is provided.

On the second section a performance comparison between 3 batch learning algorithms and 9 IL algorithms was carried out while using the dataset that was obtained through the synthetic data generators described on Chapter 4. The comparison is divided in three experimental scenarios. On the first scenario, the aim was to identify if the IL models were capable of obtaining a similar or better performance with a fewer quantity of instances on the training phase, by comparing them with the batch learning models. On the second scenario, the aim was to compare the performance behavior of the best models of each approach from the first test scenario, while observing two aspects: first, if the best IL model was able to maintain a good performance while adapting to the new data; second, if the best traditional model was able to exhibit a good performance against the new data using only the knowledge obtained from the training of the first scenario. On the third scenario, the aim was to observe if the best incremental learning model, obtained from the first scenario, would be affected negatively, in terms of performance, when receiving all of the instances from the synthetic dataset (150,000 instances) while having a pre-training with the different subsets from the second scenario. From this scenarios, it is possible to conclude that the IL approach surpasses the overall performance of batch learning models when the training and testing datasets are different. On the other hand, both approaches show a similar performance when the training and testing sets are the same.

The third section compared different IL algorithms with the consumption behavior data from a real scenario obtained after applying the reference model presented on Chapter 3. On a similar fashion, the comparison was divided in two scenarios. On the first scenario, 10 algorithms were compared by giving them different subsets of warm-up, obtaining 3 algorithms with an excellent performance (ARF, LB-ARF MLP). Then, these 3 algorithms were tested on the second scenario by analyzing how well each algorithm maintained their classification performance and how their learning rates were affected when a change in users' OTT consumption behavior was introduced in the data.

Chapter 6

Decision making

This chapter focuses on the last stage of the reference model proposed and explained on Chapter 3. The decision making task which is the responsibility of the network analyst. Having in mind that the aim of this project is to propose a recommendation of a set of personalized service degradation policies based on users' OTT consumption behavior and an IL model capable of dynamic classification of users, and also, in the fact that service degradation is majorly implemented on mobile networks [24], this chapter presents the proposal of personalized service degradation policies based on the Policy and Charging Control (PCC) architecture and the concept of a PCC rule of a 5G network presented on 3GPP Technical Specification 23.503 [163]. In order to provide some context, the first section of this chapter presents a brief description of the 5G core architecture. Then, a deeper explanation of the Policy Control Function and the definition of a PCC rule, the mechanism that enables the implementation of QoS policies inside the network is provided. Finally, the proposal of personalized service degradation policies aimed at the users' consumption behavior from the gathered dataset is presented.

6.1. 5G core architecture

We are witnessing floods of information, devices and services that are provided through the Internet. Furthermore, the proposal of 5G networks has the potential of a faster connectivity and a better quality in services to consumers, enterprises, and things (Devices, Machines, Sensors, etc.). Certainly, 5G looks very promising in terms of parameters like high bandwidth, high capacity, high speed, low/ultra-low latency, high-density coverage, high availability, low device energy consumption, high throughput, different treatments to different network slices, among many others [163], [164].

The 5G core network architecture implementation does not follow the network evolution trends followed by 2G, 3G, and 4G networks making it very different from its predecessors. In 5G, network management is envisioned to be software-driven, network functions and resources are virtualized at the edges and core. Unlike previous mobile generations, 5G implementation is based on several aspects like: Cloud-native applications, REST (REpresentational State Transfer) services-based integration, virtualized network functions, microservices, network slice-based approach of utilizing physical network resources, handling of advanced analytics and separation of control and user planes.

All these characteristics make 5G quite different and more effective than older generations. Though it is worth mentioning that some 4G operators were early adopters of virtualization of network elements and separating their EPC (Evolved Packet Core) User & Control planes for better service delivery. The core or Service Based Architecture (SBA) of 5G is illustrated in Figure 6.1

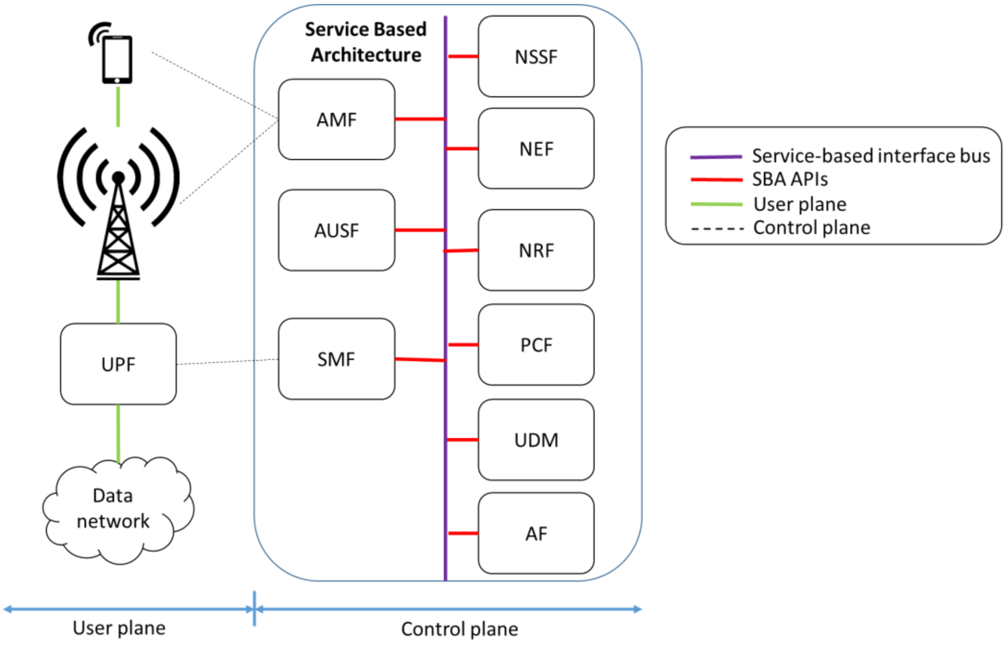


Figure 6.1. 5G core Service Based Architecture

In SBA, the approach is to adapt, evolve, expose and develop the network capabilities based on the new generation of services offered in the 5G architecture. As shown in Figure 6.1, network functions within the control plane are interconnected via the service message interface bus for exposing the network capabilities within and outside the core network. Network functions connected via the service bus in the Control Plane form the Service Based Architecture (SBA) in the 5G Core Network. In SBA, Network Functions (NF) capabilities are exposed via REST APIs and are based on the HTTP2.0 protocol. Interconnection between NFs can be based on the Request/Response model or Subscribe/Notify model for availing the different 5G Services.

A brief description of these NFs is provided as follows.

- **UPF (User Plane Function):** It performs operations like maintaining PDU Session (Packet Data Unit), packet routing & forwarding, packet inspection, policy enforcement for user plane, QoS handling, etc. When compared with 4G EPC, its functionalities resemble with elements like the SGW-U (Serving Gateway User Plane function) and the PGW-U (PDN Gateway User Plane function).
- **AMF (Access and Mobility Management Function):** It performs operations like mobility, registration and connection management, among others. It acts as a single entry point for the User Equipment (UE) connection. Based on the service requested by the consumer, AMF selects the respective SMF (Session Management Function) for managing the user session context. When compared with 4G EPC, its functionalities resemble the MME (Mobility Management Entity).
- **AUSF (Authentication Server Function):** It allows the AMF to authenticate the UE. When compared with 4G EPC, its functionalities resemble the HSS (Home Subscriber Server).
- **SMF (Session Management Function):** Performs operations like session management, IP address allocation & management for UE, User plane selection, QoS & Policy enforcement for the control plane. When compared with 4G EPC, its functionalities resemble with MME, SGW-C (Control Plane) and PGW-C (Control Plane).
- **NSSF (Network Slice Selection Function):** It maintains a list of network slice instances defined by the operator. AMF authorizes the use of network slices based on the subscription information stored in the UDM (Unified Data Management) or it can query the NSSF to authorize access to a network slice

based on the service requirements. NSSF redirects the traffic to an intended network slice.

- **NEF (Network Exposure Function):** It exposes services and resources over APIs within and outside the 5G Core. 5G services exposure by NEF is based on RESTful APIs. With the help of NEF, third party applications can also access the 5G services.
- **NRF (NF Repository Function):** It maintains the list of available network functions instances and their profiles. It also performs service registration and discovery so that different network functions can find each other via APIs. As an example, SMF which is registered to NRF; gets discoverable by AMF when UE tries to access a service type served by the SMF. Since network functions are connected via the service message bus, any authorized consumers can access the services offered through registered network functions.
- **PCF (Policy Control Function):** It supports the policy control framework, applying policy decisions and accessing subscription information to govern the network behavior. When compared with 4G EPC, its functionalities resemble with the PCRF (Policy and Charging Rules Function).
- **Unified Data Management (UDM):** It performs operations like user identification handling, subscription management, user authentication, access authorization for operations like Roaming. When compared with 4G EPC, its functionalities resemble with HSS.
- **AF (Application Function):** It performs operations like accessing the NEF for retrieving resources, interaction with the PCF for Policy Control, exposing services to end users, etc. When compared with 4G EPC, its functionalities resemble with the same component named AF.

After this brief overview of the 5G architecture, it is clear that the component of interest when defining policies is the PCF. Therefore, the next section will present a detailed explanation about how it works and the element that should be defined when proposing a QoS policy.

6.2. Policy Control Function

Policy and charging control play a very critical role in the 5G ecosystem. It provides transparency and control over the consumption of network resources during real-time service delivery. PCF governs the control plane functions via policy rules defined and user plane functions via policy enforcement. It works very closely with CHF (Charging

Function) for usage monitoring. Through PCF, operators can manage and govern network behavior. The following key aspects are supported by the PCF: QoS control, traffic routing, application detection, usage monitoring, interworking with IMS nodes, gating control, network slice enablement and roaming support. In order to understand its functionalities, the interaction and information exchanged between the PCF and its related components will be described as follows as shown in Figure 6.2.

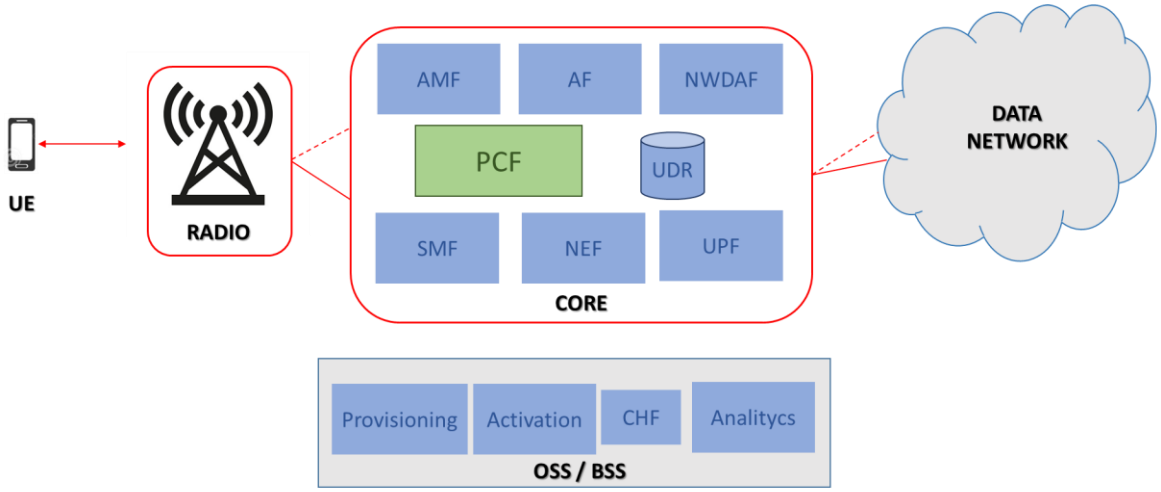


Figure 6.2. PCF related components.

- **PCF - AF Interface:** On this interface, application-level information is exchanged between AF and PCF. This includes information like: bandwidth requirements for QoS, identifying application service providers and applications, traffic routing based on applications access, identifying application traffic for charging and policy control.
- **PCF - UDR Interface:** On this communication, the PCF retrieves the policy, subscription and application specific data from the UDR (Unified Data Repository). Also, policy control related subscription and application data gets provisioned into the UDR. Furthermore, the UDR can also generate notifications based on the changes in the subscription information depending on the operator's pricing model.
- **PCF - NWDAF Interface:** The PCF shall be able to collect directly slice specific network status analytic information from the NWDAF (Network Data Analytics Function). The NWDAF provides network data analytics (i.e. load level information) to the PCF on a network slice level and the NWDAF is not required to be aware of the current subscribers using the slice. PCF shall be able to use that data in its policy decisions.

- **PCF - NEF Interface:** The NEF exposes network function services and resources to the external world. In terms of interaction with the PCF, it exposes the capabilities of network functions for supporting the policy and charging operations.
- **PCF - CHF Interface:** Through this interface, operators can manage and control subscriber spending as well as usage control. The CHF (CHarging Function) stores the policy counter information against the subscriber pricing plan and notifies the PCF whenever a subscriber breaches the policy thresholds based on the usage consumption. Once the PCF receives policy trigger information, it applies the policy decision by interacting with the SMF (which in turn informs the UPF for the policy enforcement).
- **PCF - AMF Interface:** The AMF acts as a single entry point for the UE connection. The PCF provides access and mobility management related policies for the AMF in order to trigger policy rules on the UE or user sessions.
- **PCF - SMF Interface:** The SMF receives control plane information from several sources. From network functions like the AMF and user plane information from the UPF. This information includes: subscription details, QoS and PDU Session level. Also, the SMF requests usage related information from the UPF. It triggers the PCF to enforce policy decisions once the policy trigger related to the session management is met. Similarly, PCF provisions the policy and charging control decision on the SMF.
- **PCF - SMF - UPF Interface:** The PCF and UPF do not communicate directly with each other. They exchange policy actions and enforcements through the SMF. The SMF provisions the policy and threshold rules to the UPF for usage control based on both the static and dynamic policy rules configured in PCF, pre-defined rules in SMF and/or credit control triggers received from CHF.

After analyzing each interface, it is clear that all of them provide vital information to the PCF to support and trigger the management and enforcement of policies. With this in mind, Figure 6.2 illustrates a sequence diagram with an example about a service session that occurs once a user equipment requires a service usage within a 5G network and then exceeds his/her consumption limit provoking the enforcement of a service degradation policy.

As it can be seen on Figure 6.3, at first the UE starts the communication (step 1) by accessing a service that the customer needs to use (e.g., video streaming, VoIP call, social networks, etc.). This request goes through the AMF that asks the PCF for user profile information in order to verify its identity (step 2). The PCF communicates with the UDR, where all the subscription and profile data is stored, and asks for the current user profile (step 3). If the user profile is indeed found in the UDR, it answers the PCF request with the profile information details (step 4). Then, based on this information the PCF prepares the policies that should be applied to this subscriber and triggers the usage monitoring that is executed between the SMF and UPF (steps 5 and 6). Subsequently, the SMF checks the credit information (data plan details, consumption quota to date, policy and charging counters) by communicating with the CHF and this component delivers the requested information about the current subscriber (steps 7 and 8). Once all this information is gathered and authorized, the policy rules and usage monitoring functions are activated by the SMF and UPF and the subscriber starts using the service requested (step 9).

The process up to this point is the normal procedure that occurs on a 5G network to enable a user the consumption of a service.

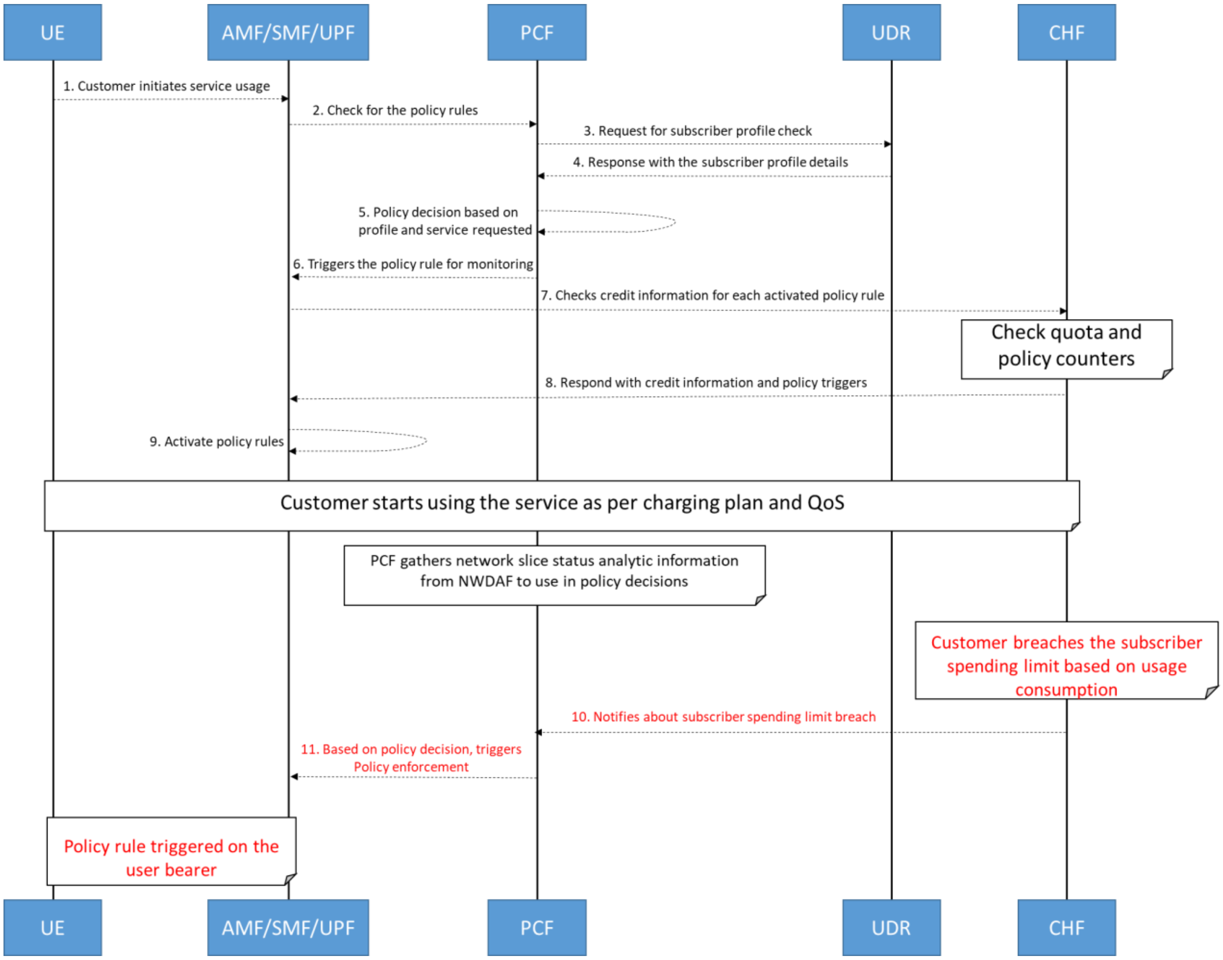

UE
AMF/SMF/UPF
PCF
UDR
CHF
1. Customer initiates service usage
2. Check for the policy rules
3. Request for subscriber profile check
4. Response with the subscriber profile details
5. Policy decision based on profile and service requested
6. Triggers the policy rule for monitoring
7. Checks credit information for each activated policy rule
Check quota and policy counters
8. Respond with credit information and policy triggers
9. Activate policy rules
Customer starts using the service as per charging plan and QoS
PCF gathers network slice status analytic information from NWDAF to use in policy decisions
Customer breaches the subscriber spending limit based on usage consumption
10. Notifies about subscriber spending limit breach
11. Based on policy decision, triggers Policy enforcement
Policy rule triggered on the user bearer
UE
AMF/SMF/UPF
PCF
UDR
CHF

Figure 6.3. Sequence diagram of a service session with service degradation on a 5G network.

While the user consumes the service, the PCF is watching over two aspects. First, the PCF watches over the current status of the network slice by communicating with the NWDAF. Second, the PCF along with the SMF and UPF are expecting a notification about the usage consumption of the subscriber from the CHF. When the user exceeds its consumption or spending limit, the CHF notifies the PCF and it triggers the policy enforcement. This is the moment where the service degradation policies get into action (steps 10 and 11). After the PCF notification, the SMF and the UPF enforces the policies and affect the connection of the subscriber in the current service consumption according to the definition of the policy. All of the steps related to the enforcement of service degradation policies are highlighted in red within the sequence diagram.

With the previous example, it is clear how the elements of the core architecture of a 5G network interact to enforce QoS policies upon the communications generated by the users. Now, it is necessary to define the tool that allows the definition of the policies that are contained in the PCF and its parameters, so we can proceed with the proposal of personalized service degradation policies based on the user behavior analysis obtained from the dataset generated with the reference model. The following section will define what is known as the PCC rule.

6.3. Policy and Charging Control rule

The Policy and Charging Control rule (PCC rule) or QoS rule comprises the information that is required to enable the user plane with detection, policy control and charging capabilities for a Service Data Flow (SDF). An SDF is an IP flow or aggregation of IP flows of user traffic classified by the type of the service in use. Different SDFs have different QoS class and hence an SDF serves as a unit by which QoS rules are applied in accordance with the policy and charging procedure in the network [165].

The purpose of the PCC rule is to detect a packet belonging to an SDF, identify the service the SDF contributes to, provide applicable charging parameters for the SDF, and provide policy control for the SDF. Every PCC rule is defined by the operator. There are two different types of PCC rules: dynamic rules and predefined rules. The dynamic PCC rules are provisioned by the PCF to the SMF, while the predefined PCC rules are configured into the SMF, as described in TS 23.503 [163], and only referenced by the PCF. Each dynamic rule, as defined by the PCF, is forwarded to the SMF over their communication interface to be enforced for each SDF. After enforcing the PCC rules, when IP packets arrive, the SMF detects the SDF that each packet belongs to, and applies a PCC rule to each packet according to their respective SDF.

After analyzing the structure of the QoS model and the PCC rules in the 5G architecture, we can state that 5G supports both QoS Flows that require Guaranteed flow Bit Rate (GBR QoS Flows) and QoS Flows that do not require Guaranteed flow Bit Rate (Non-GBR QoS Flows). Any QoS Flow is characterized by:

- A QoS profile provided by the SMF to the access network via the AMF.
- One or more PCC or QoS rules with QoS parameters which can be provided by the SMF to the UE via the AMF.

A PCC rule consists of the following groups of attributes: a policy rule name or rule identifier, SDF detection parameters, charging parameters and policy control parameters. Considering the objectives stated on this research project related to service degradation, the charging parameters are not within its scope. On the other hand, the policy control parameters are of vital importance in the definition of PCC rules related with service degradation. Based on the QoS profile defined for 5G, the parameters that must be taken into consideration when defining PCC rules include the parameters defined on Table 6.1:

Element	Definition
Policy rule name	This 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 a 5-tuple (Source IP address, Destination IP address, Source port number, Destination port number, and Protocol ID). The Protocol ID is checked with DPI.
5G QoS Identifier – 5QI	It's an integer that indicates different QoS performance characteristics of each IP packet. QCI values are standardized to reference specific QoS characteristics, and each QCI contains standardized performance characteristics (values), such as resource type (GBR or non-GBR), priority, 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}).
GBR & Non-GBR (Guaranteed Bit Rate)	For a QoS flow, having a GBR resource type means the bandwidth of the bearer is guaranteed, i.e., a GBR type QoS flow has a "guaranteed bit rate" associated as one of its QoS parameters. The 5QI of a GBR type QoS flow can range from 1 to 4 (and also includes 65, 66, 67 and 75). On the other hand, having a non-GBR

	resource type means that the QoS flow is a best effort type flow and its bandwidth is not guaranteed. The 5QI of a non-GBR type QoS flow can range from 5 to 9 (including 69, 70, 79, 80 and 81).
MBR (Maximum Bit Rate)	This parameter indicates the maximum bit rate allowed in the 5G network. Any packets arriving at the QoS flow exceeding the specified MBR will be discarded.
ARP (Allocation and Retention Priority)	An integer ranging from 1 to 15, with 1 being the highest level of priority. When a new QoS flow is needed in a 5G network with insufficient resources, the SMF decides, based on the ARP, whether to: remove the existing QoS flow and create a new one (e.g. removing an QoS flow with low priority ARP to create one with high priority ARP) or refuse to create a new one.
Gate status	The gate status indicates whether the SDF, detected by the SDF template, may pass (gate is open) or shall be discarded (gate is closed).

Table 6.1. 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.

6.4. Proposal of personalized service degradation policies

By analyzing the structure of a PCC rule and considering that in the technical specification the 5QI parameter defines the ARP and if a QoS flow is GBR (QCI 1 to 4) or non-GBR (QCI 5 to 9), there are only two possibilities for the definition of a service degradation policy: degrade the service by affecting the Maximum Bit Rate (MBR) associated to the SDF or block the service by modifying the Gate status parameter.

Considering that 56 OTT applications were identified in the dataset, it was necessary to analyze which applications were most commonly consumed on each group of users. With this in mind, the 20 most used applications were identified in terms of both time and data occupation. Figures 6.4, 6.5 and 6.6 illustrate the 20 applications most commonly used in terms of time occupation for low, medium and high consumption users respectively. Similarly, Figures 6.7, 6.8 and 6.9 illustrate the 20 most used applications in terms of data occupation for the three groups.

It is important to mention that a larger amount of time occupation does not mean more data occupation. From the figures the following remarks can be stated:

- The application that was used during the most quantity of time by the low consumption users was Google with a total of 10.26 hours (36942.51 seconds). While the applications that demanded the most amount of data occupation from highest to lowest were: HTTP (browsing), GoogleDocs, YouTube, Gmail, Google, GoogleDrive and AmazonVideo.
- For the medium consumption users, Google remained as the application where users spent most quantity of time with 77.19 hours (277919.25 seconds). On the other hand, in the data occupation it can be appreciated that the applications that required most network resources were: YouTube, GoogleDocs, Gmail, GoogleDrive, GoogleHangoutDuo, Spotify, Twitter and AmazonVideo.
- Finally, the high consumption users exhibit as well a large total amount of time spent on Google with 18.52 hours (66678.48 seconds). On data occupation, these users consumed the most the following applications: YouTube, GoogleDrive, GoogleHangoutDuo, Amazon, Facebook, Google, WhatsApp and HTTP.

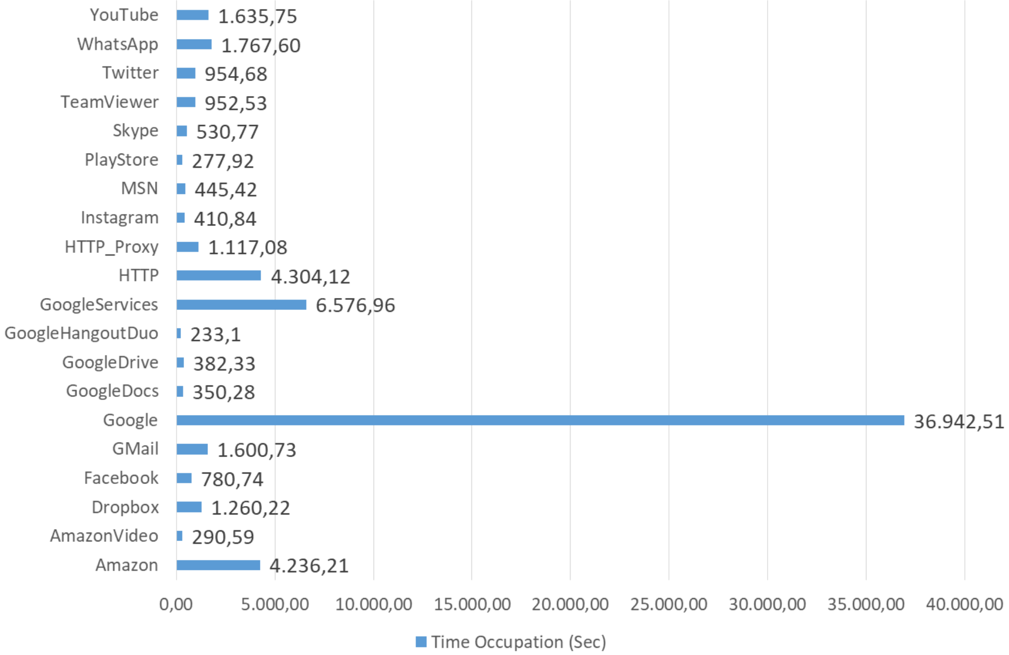


Figure 6.4. Time occupation – Low consumption users

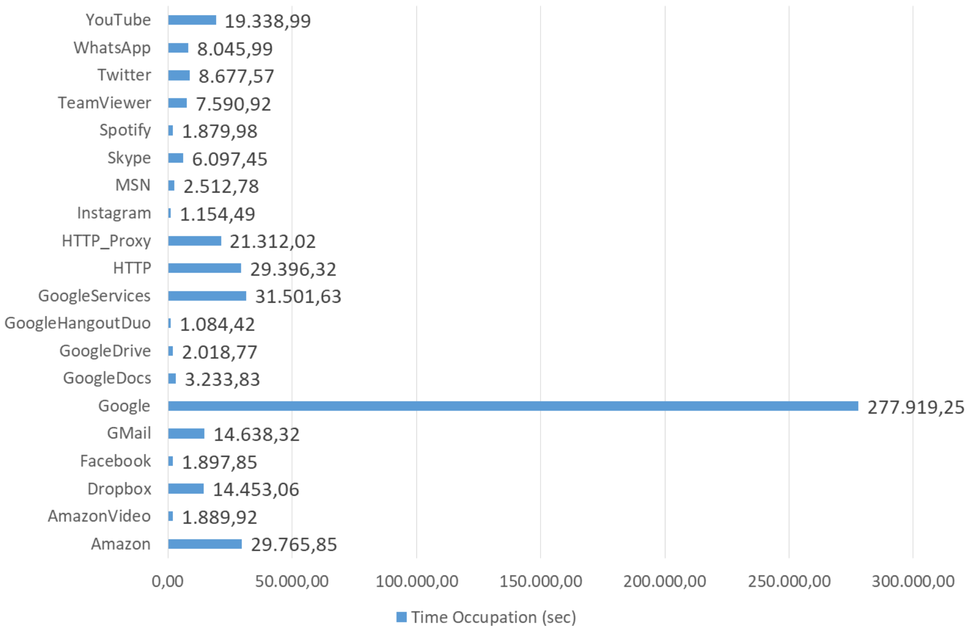


Figure 6.5. Time occupation – Medium consumption users

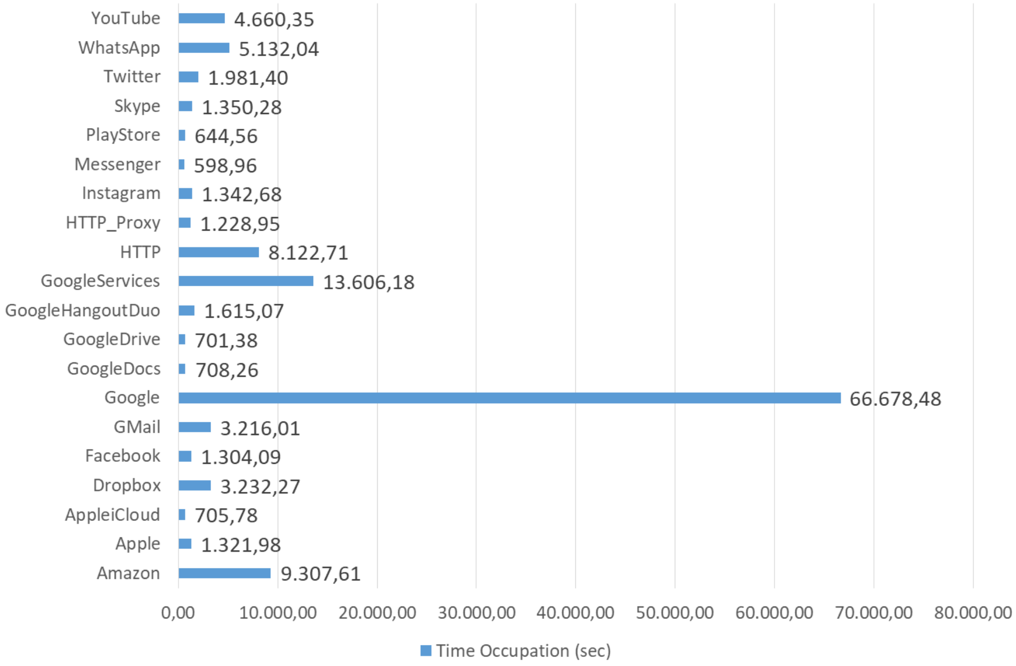


Figure 6.6. Time occupation – High consumption users

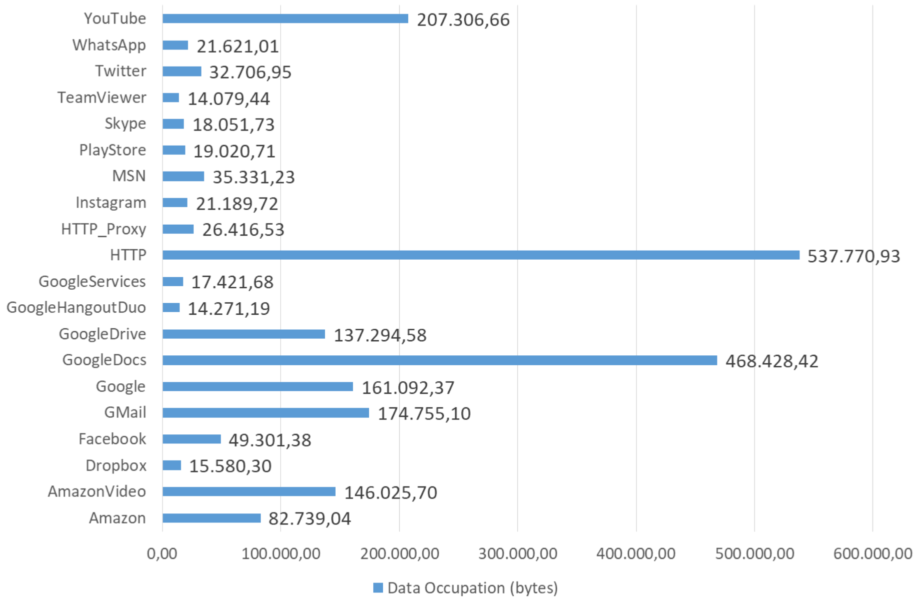


Figure 6.7. Data occupation – Low consumption users

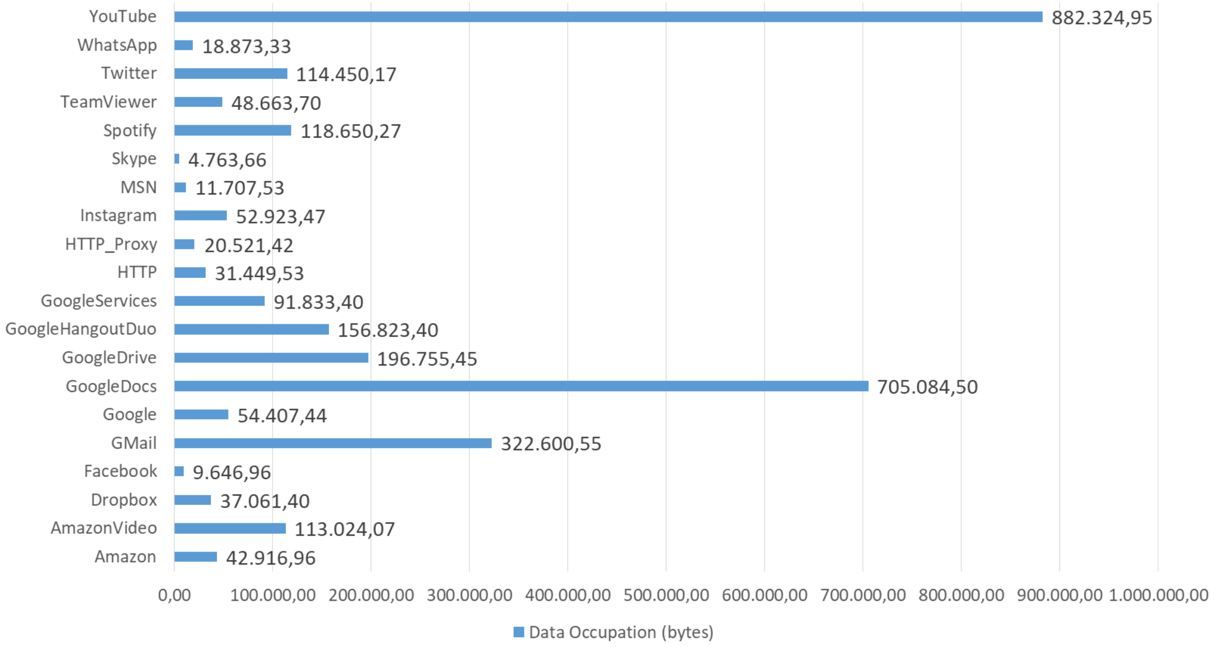


Figure 6.8. Data occupation – Medium consumption users

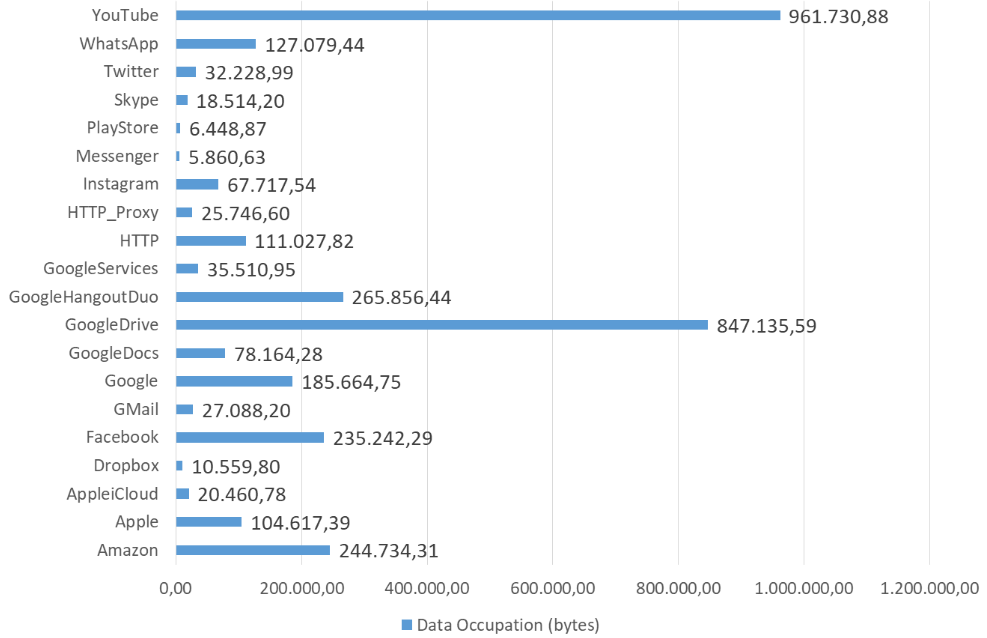


Figure 6.9. Data occupation – High consumption users

After analyzing the consumption behavior of the three groups of users and identifying the applications that demanded most network resources (7 applications for the low consumption group, 8 applications for the medium consumption group and 8 applications for the high consumption group), it is possible to propose a set personalized service degradation policies per group. Table 6.2 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	
Amazon Video	X		X			X
Apple		X		X		X
Apple Icloud		X		X		X
Dropbox		X		X		X
Facebook		X		X	X	
Gmail	X		X			X
Google	X			X	X	
Google Docs	X		X			X
Google Drive	X		X		X	

Google Hangout Duo		X	X		X	
Google Services		X		X		X
HTTP	X			X	X	
HTTP_Proxy		X		X		X
Instagram		X		X		X
Messenger		X		X		X
Skype		X		X		X
Spotify		X	X			X
Teamviewer		X		X		X
Twitter		X	X			X
WhatsApp		X		X	X	
Youtube	X		X		X	

Table 6.2. Policies recommendation

When a service degradation is identified, i.e., when a user exceeds his/her allowed 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 8 most used applications are still functional and the rest are blocked. Finally, in a similar fashion for the High Consumption group the 8 most used applications are degraded in the bit rate and the others are blocked.

This is an example in how the network administrator can execute the decision making process based on the gathered dataset, the IL model, the consumption analysis performed on the users’ behavior and save network resources while 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. Therefore, the reference model and each process is adaptable to the specific needs and interest of the network operator.

As an example Tables 6.3 and 6.4 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 minimum expected speed for 5G according to the technical specification is 50 Mbps [163] and also, speeds that can be offered in average on a 5G network according to the report presented by OpenSignal are 291.2 Mbps in Singapore and 52.3 Mbps in USA [166]. It is important to clarify that such speed is not the sum of both upload and download links but one link only. 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 Qos flows (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.

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.

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

Policy rule name	**SDF template**	**SDF GBR**	**SDF MBR**	**SDF 5QI/ARP**	**SDF Gating status**	**Qos flow summary**

Facebook blocking policy	UL:(192.168.121.2, *,*,*,Facebook)
	DL:(*,192.168.121.2, *,*,Facebook)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 9
				ARP = 9
					Closed (not permitted)	QCI= 9
						ARP= 9
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

YouTube degradation	UL:(192.168.121.2, *,*,*,Youtube)
	DL:(*,192.168.121.2, *,*,Youtube)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 8
				ARP = 8
					Open (permitted)	QCI= 8
						ARP= 8
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

Gmail degradation	UL:(192.168.121.2, *,*,*,Youtube)
	DL:(*,192.168.121.2, *,*,Gmail)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 8
				ARP = 8
					Open (permitted)	QCI= 8
						ARP= 8
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

Table 6.3. Policies structure example – Low Consumption users.

Policy rule name	SDF template	SDF GBR	SDF MBR	SDF 5QI/ARP	SDF Gating status	Qos flow summary

Instagram blocking policy	UL:(192.168.121.7, *,*,*,Instagram)
	DL:(*,192.168.121.7, *,*,Instagram)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 9
				ARP = 9
					Closed (not permitted)	QCI= 9
						ARP= 9
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

Spotify degradation	UL:(192.168.121.7, *,*,*,Spotify)
	DL:(*,192.168.121.7, *,*,Spotify)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 8
				ARP = 8
					Open (permitted)	QCI= 8
						ARP= 8
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

Twitter degradation	UL:(192.168.121.7, *,*,*,Twitter)
	DL:(*,192.168.121.7, *,*,Twitter)
		N.A.	UL: 25Mbps
			DL: 25Mbps
				5QI = 9
				ARP = 9
					Open (permitted)	QCI= 9
						ARP= 9
						MBR-UL: 25 Mbps
						MBR-DL: 25 Mbps

Table 6.4. Policies structure example – Medium Consumption users

With this it can be concluded that the decision making stage from the reference model has been applied, and that a set of personalized service degradation policies considering the consumption trends of OTT applications of users has been proposed, following the 5G architecture presented in the 3GPP technical specification 23.503 [163].

Summary

The major conclusion and contribution of this chapter is the recommended set of personalized service degradation policies based on the data analysis performed on the datasets built through the implementation of the reference model. This recommendation can change completely depending on the needs and interests of the network operator and the process can be replicated if new information is gathered.

Specifically, this chapter intended to apply the decision making process explained on the reference model of chapter three with the aim of proposing a set of personalized service degradation policies based on the users' consumption behavior obtained from the analysis of the gathered dataset and the IL model. The chapter is divided into four sections.

The first section presented a detailed overview of the 5G architecture as described in the 3GPP technical specification 23.503, focusing on the components closely related to the implementation of QoS policies inside the network. Then an example of the interaction and the elements that are commonly related in the application of QoS policies were described. These elements include: the Session Management Function (SMF), the User Plane Function (UPF), the Policy Charging Function (PCF) and the PCC rules divided in their two types predefined and dynamic. Also, the different parameters that must be considered in order to propose QoS policies inside a 5G network were presented and analyzed. In the fourth section, a set of personalized service degradation policies was presented having in mind the behavior observed from the clustering analysis. Considering that most of the parameters of a QoS policy (PCC rule) are standardized through the 5G QoS Class Identifier (5QI), it was 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 the maximum average bit rate of networks of different countries as presented on a technical report from OpenSignal [166], or block the service flows by setting the gating control parameter to closed. With this two possibilities a set of dynamic PCC rules were presented, having in mind the consumption behavior of each group (high, medium and low consumption).

Chapter 7

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.

7.1. 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 works related to user profiling it can be concluded that none of the found approaches consider the proposal of a reference model that can be replicated and obtain an overview of the users' OTT consumption behavior. In fact, in most of the cases, the model proposed by other researches is a ML model capable of doing a specific task (classification or prediction) but it does not offer guidelines on how to perform the user profiling nor can be implemented on a context different from the one considered on the work.

- From the works related to service degradation, it can be concluded that although there are multiple analysis and approaches that talk about 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 look into a personalization considering their consumption behavior.

- From the works related to incremental learning, traffic classification and knowledge defined networking, it can be concluded that IL is becoming more and more important in the general ML scope due to its adaptation capabilities and that, eventually, the KDN paradigm will include this kind of algorithms to offer a better support to network managers in their respective responsibilities.

- We observed that a major limitation in the research domain is the lack of a guideline (specification) that would clarify how an IL-based network traffic classification should be implemented. With this in mind, we introduced a reference model that could be considered as a guideline for IL-based network traffic classification. We defined its main components, actors, and workflow that can be regarded as the reference guide for proposing more efficient solutions and understanding the requirements of such a solution. We believe that network operators and researches on this domain can benefit from analyzing and applying the reference model and even adapting certain steps to their own use cases since it presents a clear path from the data gathering to the model selection.

- By applying the reference model, a set of 4 datasets holding labeled network traffic flows and users' consumption profiles has been obtained and shared within the research community for further experimentation in the area of IL and the ML scope. So far, the datasets have been useful since they are continuously implemented by other researchers in their respective investigations.

- We observed that several tools exist for creating a labeled dataset containing labeled network traffic flow records. However, the fact is that the calculations performed by the existing tools often yield different results. Many traffic flow features are not appropriately calculated, while essential features are entirely missing. Therefore, an application named Flow Labeler was developed and implemented aiming at fulfill the needs when creating datasets related to network traffic (flow statistics calculation and flow labeling). Flow Labeler obtained satisfactory results in the processing of raw IP packets while facilitating the creation of new network flow datasets.

- After the preprocessing stage of the dataset presented on Chapter 4, 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. Therefore Deep Packet Inspection (DPI) remains as the main solution to resolve such problem.

- A set of attributes, taken from the flow statistics of a traditional network monitoring and focused on time and data occupation of network resources, 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, in general, inside the campus of the Universidad Del Cauca, the users consume the same applications, however vary in the intensity of their consumption.

- From the comparison between IL and traditional batch learning algorithms, it is possible to conclude that the IL approach surpasses the overall performance of batch learning models when the training and testing datasets are different. However, both approaches show a similar performance when the training and testing sets are the same.

- From the IL algorithms comparison, we observed that ARF, LB-ARF, and MLP achieve the best results. The results also show that ARF and the composition of LB and ARF not only achieve high accuracy but can also maintain classification performance over time, in the event of changes in the behavior of the observed samples. As such, they are suitable for real-world implementation where their main advantage relies on their efficacy, robustness, and low resource consumption.

- After analyzing the 5G architecture on chapter 6, a set of attributes that must be considered in order to propose a recommendation of 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 recommendation of 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 for a 5G network.

- When we consider the volatility of the Internet and OTT applications, the incremental learning approach may be a suitable option when dealing with the possible changes that users may present in their OTT consumption behavior over time. This represents an important advantage for network administrators since such approach overcomes the weakness that a traditional model presents about their incapability of adapting to new data without a new training process.

7.2. Future works

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

- Build a mechanism that enables the automation and implementation of the set of personalized service degradation policies within an Internet network.

- Develop and integrate a knowledge plane that enables the implementation of artificial intelligence techniques inside a network implementing IL models like ARF and LB/ARF that obtained good results in the classification.

- Integrate an infrastructure that facilitates the gathering of data related to the consumption of OTT applications from users inside a network.

- Develop a proposal of personalized charging policies that are suited for a user based on his/her consumption behavior.

- Develop a system capable of recommending data plans for users based on their consumption behaviors and interests.

- Develop a mechanism capable of associating application labels to patterns detected in flow statistics to obtain a labeling alternative to DPI.

- Include the experience perceived by the users in the study of their consumption behavior of OTT applications.

References

[1] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the internet,” in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe, Germany, Aug. 2003, pp. 3–10, doi: 10.1145/863955.863957.

[2] A. Mestres *et al.*, “Knowledge-Defined Networking,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Sep. 2017, doi: 10.1145/3138808.3138810.

[3] S. Ponmaniraj, R. Rashmi, and M. V. Anand, “IDS Based Network Security Architecture with TCP/IP Parameters using Machine Learning,” in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, Sep. 2018, pp. 111–114, doi: 10.1109/GUCON.2018.8674974.

[4] L. Zhang, X. Meng, and H. Zhou, “Network Fault Diagnosis Using Hierarchical SVMs Based on Kernel Method,” in *2009 Second International Workshop on Knowledge Discovery and Data Mining*, Jan. 2009, pp. 753–756, doi: 10.1109/WKDD.2009.79.

[5] S. Troia *et al.*, “Machine Learning-assisted Planning and Provisioning for SDN/NFV-enabled Metropolitan Networks,” in *2019 European Conference on Networks and Communications (EuCNC)*, Jun. 2019, pp. 438–442, doi: 10.1109/EuCNC.2019.8801956.

[6] 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)*, Oct. 2016, pp. 2451–2455, doi: 10.1109/CompComm.2016.7925139.

[7] J. S. Rojas, Á. Rendón, and J. C. Corrales, “Personalized Service Degradation Policies on OTT Applications Based on the Consumption Behavior of Users,” in *Computational Science and Its Applications – ICCSA 2018*, 2018, pp. 543–557, doi: https://doi.org/10.1007/978-3-319-95168-3_37.

[8] A. Gepperth and B. Hammer, “Incremental learning algorithms and applications,” presented at the European Symposium on Artificial Neural Networks (ESANN), 2016, Accessed: Apr. 15, 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01418129/document.

[9] D. L. Silver, “Machine Lifelong Learning: Challenges and Benefits for Artificial General Intelligence,” in *Artificial General Intelligence*, 2011, pp. 370–375.

[10] Wesley Clover, “Over-The-Top (OTT) a dramatic makeover of global communications.” 2014, [Online]. Available: http://www.wesleyclover.com/wp-content/uploads/2014/07/Wesley-Clover-Newsletter-July2014.pdf.

[11] Wedge Green and Barbara Lancaster, “Over The Top Services,” *Pipeline*, Volume 4, Issue 7, p. 9, 2006.

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

[13] 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, doi: 10.1145/2207676.2208714.

[14] 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, doi: 10.1145/2702123.2702218.

[15] 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)*, Aug. 2015, pp. 1485–1494, doi: 10.1109/PICMET.2015.7273046.

[16] J. S. Rojas Meléndez, “Personalized service degradation on ott applications,” Apr. 2018, Accessed: Jun. 04, 2020. [Online]. Available: http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1367.

[17] “IP Network Traffic Flows Labeled with 75 Apps.” https://kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps (accessed Apr. 15, 2019).

[18] “OTT consumption profile - Unicauca dataset.” https://kaggle.com/jsrojas/ott-consumption-profile-dataset (accessed Apr. 15, 2019).

[19] “LTE Policy and Charging Control (PCC),” *Network Manias*. https://www.netmanias.com/en/?m=view&id=techdocs&no=6562 (accessed Apr. 03, 2018).

[20] J. S. Rojas, Á. Rendón, and J. C. Corrales, “Consumption Behavior Analysis of Over the Top Services: Incremental Learning or Traditional Methods?,” *IEEE Access*, vol. 7, pp. 136581–136591, 2019, doi: 10.1109/ACCESS.2019.2942782.

[21] 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, doi: 10.1016/0169-7552(95)00096-1.

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

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

[24] Jan Krämer and Lukas Wiewiorra, “Data Caps and Two-Sided Pricing: Evaluating Managed Service Business Models,” presented at the ECIS 2014 - European Conference on Information Systems, Tel Aviv, Israel, 2014, [Online]. Available: https://aisel.aisnet.org/ecis2014/proceedings/track10/10/.

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

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

[27] “Quality of Service Regulation Manual.” https://www.itu.int/pub/D-PREF-BB.QOS_REG01-2017 (accessed Mar. 01, 2018).

[28] B. Trammell and B. Claise, “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information.” https://tools.ietf.org/html/rfc7011 (accessed Apr. 23, 2020).

[29] N. Brownlee, “Flow-Based Measurement: IPFIX Development and Deployment,” *ResearchGate*, 2011.

https://www.researchgate.net/publication/220243050_Flow-Based_Measurement_IPFIX_Development_and_Deployment (accessed Jun. 23, 2020).

[30] R. Hofstede *et al.*, “Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX,” *IEEE Commun. Surv. Tutor.*, 2014, doi: 10.1109/COMST.2014.2321898.

[31] 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, doi: 10.1109/SURV.2013.100613.00161.

[32] 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, doi: 10.1145/2480741.2480747.

[33] R. R. Ade and P. R. Deshmukh, “Methods for Incremental Learning : A Survey,” *Int. J. Data Min. Knowl. Manag. Process IJDKP*, vol. 3, no. 4, Jul. 2013, doi: 10.5121/ijdkp.2013.3408.

[34] 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, 2008, pp. 68–77, [Online]. Available: http://dl.acm.org/citation.cfm?id=2227115.2227123.

[35] M. A. Azad, M. Alazab, F. Riaz, J. Arshad, and T. Abullah, “Socioscope: I know who you are, a robo, human caller or service number,” *Future Gener. Comput. Syst.*, vol. 105, pp. 297–307, Apr. 2020, doi: 10.1016/j.future.2019.11.007.

[36] A. Hess, I. Marsh, and D. Gillblad, “Exploring communication and mobility behavior of 3G network users and its temporal consistency,” in *2015 IEEE International Conference on Communications (ICC)*, Jun. 2015, pp. 5916–5921, doi: 10.1109/ICC.2015.7249265.

[37] D. Rajashekar, A. N. Zincir-Heywood, and M. I. Heywood, “Smart Phone User Behaviour Characterization Based on Autoencoders and Self Organizing Maps,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 319–326, doi: 10.1109/ICDMW.2016.0052.

[38] E. Mucelli Rezende Oliveira, A. Carneiro Viana, K. P. Naveen, and C. Sarraute, “Mobile data traffic modeling: Revealing temporal facets,” *Comput. Netw.*, vol. 112, pp. 176–193, Jan. 2017, doi: 10.1016/j.comnet.2016.10.016.

[39] S. Zhao *et al.*, “Gender Profiling From a Single Snapshot of Apps Installed on a Smartphone: An Empirical Study,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 2, pp. 1330–1342, Feb. 2020, doi: 10.1109/TII.2019.2938248.

[40] M. K. Ehsan, “Performance Analysis of the Probabilistic Models of ISM Data Traffic in Cognitive Radio Enabled Radio Environments,” *IEEE Access*, vol. 8, pp. 140–150, 2020, doi: 10.1109/ACCESS.2019.2962143.

[41] D. López, E. Rivas, and O. Gualdron, “Primary user characterization for cognitive radio wireless networks using a neural system based on Deep Learning,” *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 169–195, Jun. 2019, doi: 10.1007/s10462-017-9600-4.

[42] J. Hernández, D. López, and N. Vera, “Primary user characterization for cognitive radio wireless networks using long short-term memory,” *Int. J. Distrib. Sens. Netw.*,

vol. 14, no. 11, p. 1550147718811828, Nov. 2018, doi: 10.1177/1550147718811828.

[43] D. A. L. Sarmiento, J. C. B. Ordoñez, and E. Rivas, “User Characterization through Dynamic Bayesian Networks in Cognitive Radio Wireless Networks,” International Journal of Engineering and Technology, Aug-Sep 2016, doi: 10.21817/ijet/2016/v8i4/160804043.

[44] S. Yan, M. Peng, M. A. Abana, and W. Wang, “An Evolutionary Game for User Access Mode Selection in Fog Radio Access Networks,” *IEEE Access*, vol. 5, pp. 2200–2210, 2017, doi: 10.1109/ACCESS.2017.2654266.

[45] A. Nigam, “Beyond Who and What: Data Driven Approaches for User Characterization,” WSDM '18: Proceedings of the Eleventh ACM International Conference on Web Search and Data MiningFebruary 2018 Pages 753–doi: 10.1145/3159652.3170455.

[46] M. V. Vinupaul, R. Bhattacharjee, R. Rajesh, and G. S. Kumar, “User characterization through network flow analysis,” in *2016 International Conference on Data Science and Engineering (ICDSE)*, Aug. 2016, pp. 1–6, doi: 10.1109/ICDSE.2016.7823965.

[47] A. Bakhshandeh and Z. Eskandari, “An efficient user identification approach based on Netflow analysis,” in *2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, Aug. 2018, pp. 1–5, doi: 10.1109/ISCISC.2018.8546856.

[48] F. Shaman, B. Ghita, N. Clarke, and A. Alruban, “User Profiling Based on Application-Level Using Network Metadata,” in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, Jun. 2019, pp. 1–8, doi: 10.1109/ISDFS.2019.8757503.

[49] N. J. Qasim, S. M. Mohammed, A. S. Sosa, and I. Albarazanchi, “Reactive protocols for unified user profiling for anomaly detection in mobile Ad Hoc networks,” Periodicals of Engineering and Natural Sciences, Vol 7, No 2 (2019), doi: 10.21533/PEN.V7I2.497.

[50] M. Mamun, R. Lu, and M. Gaudet, “Tell Them from Me: An Encrypted Application Profiler,” in *Network and System Security*, Cham, 2019, pp. 456–471, doi: 10.1007/978-3-030-36938-5_28.

[51] G. Alotibi, N. Clarke, Fudong Li, and S. Furnell, “User profiling from network traffic via novel application-level interactions,” in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec. 2016, pp. 279–285, doi: 10.1109/ICITST.2016.7856712.

[52] A. Parres-Peredo, I. Piza-Davila, and F. Cervantes, “MapReduce Approach to Build Network User Profiles with Top-k Rankings for Network Security,” in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb. 2018, pp. 1–5, doi: 10.1109/NTMS.2018.8328702.

[53] M. Dahmane and S. Foucher, “Combating Insider Threats by User Profiling from Activity Logging Data,” in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, Apr. 2018, pp. 194–199, doi: 10.1109/ICDIS.2018.00039.

[54] W. Alswiti, J. Alqatawna, B. Al-Shboul, H. Faris, and H. Hakh, “Users Profiling Using Clickstream Data Analysis and Classification,” in *2016 Cybersecurity and*

Cyberforensics Conference (CCC), Aug. 2016, pp. 96–99, doi: 10.1109/CCC.2016.27.

[55] T. Tuna *et al.*, “User characterization for online social networks,” *Soc. Netw. Anal. Min.*, vol. 6, no. 1, p. 104, Nov. 2016, doi: 10.1007/s13278-016-0412-3.

[56] P. N. Terevinto, A. Pont, J. A. Gil, and J. Domenech, “A flexible workload model based on roles of interactive users in social networks,” in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2016, pp. 524–529, doi: 10.1109/IFIPNetworking.2016.7497254.

[57] D. Villanueva, I. González-Carrasco, J. L. López-Cuadrado, and N. Lado, “SMORE: Towards a semantic modeling for knowledge representation on social media,” *Sci. Comput. Program.*, vol. 121, pp. 16–33, Jun. 2016, doi: 10.1016/j.scico.2015.06.008.

[58] J. Zheng, S. Liu, and L. M. Ni, “User characterization from geographic topic analysis in online social media,” in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, Aug. 2014, pp. 464–471, doi: 10.1109/ASONAM.2014.6921627.

[59] K. Zahra, F. Azam, W. H. Butt, and F. Ilyas, “A Framework for User Characterization based on Tweets Using Machine Learning Algorithms,” ICNCC 2018: Proceedings of the 2018 VII International Conference on Network, Communication and Computing, December 2018, Pages 11–16, doi: 10.1145/3301326.3301373.

[60] I. Paul, A. Khattar, P. Kumaraguru, M. Gupta, and S. Chopra, “Elites Tweet? Characterizing the Twitter Verified User Network,” *ArXiv181209710 Cs*, Mar. 2019, Accessed: Jul. 16, 2020. [Online]. Available: http://arxiv.org/abs/1812.09710.

[61] D. Sánchez-Moreno, V. F. L. Batista, M. D. M. Vicente, A. B. G. González, and M. N. Moreno-García, “A session-based song recommendation approach involving user characterization along the play power-law distribution,” *Complexity*, vol. 2020, pp. 1–13, Jun. 2020, doi: 10.1155/2020/7309453.

[62] B. Gao, S. Du, X. Li, and F. Liu, “Research on the Application of Persona in Book Recommendation System,” Journal of Physics: Conference Series, Volume 910, The 2017 International Conference on Cloud Technology and Communication Engineering (CTCE2017) 18–20 August 2017, Guilin, China, doi: 10.1088/1742-6596/910/1/012023.

[63] R. M. Filho, J. M. Almeida, and G. L. Pappa, “Twitter population sample bias and its impact on predictive outcomes: A case study on elections,” in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug. 2015, pp. 1254–1261, doi: 10.1145/2808797.2809328.

[64] R. Cui, G. Agrawal, and R. Ramnath, “Tweets can tell: activity recognition using hybrid gated recurrent neural networks,” *Soc. Netw. Anal. Min.*, vol. 10, no. 1, p. 16, Mar. 2020, doi: 10.1007/s13278-020-0628-0.

[65] Y. Gu, Z. Ding, S. Wang, and D. Yin, “Hierarchical User Profiling for E-commerce Recommender Systems,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, Houston, TX, USA, Jan. 2020, pp. 223–231, doi: 10.1145/3336191.3371827.

[66] İ. Topalli and S. Kilinç, “User profiling for TV program recommendation based on hybrid television standards using controlled clustering with genetic algorithms and

artificial neural networks," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, Art. no. 3, May 2020, doi: 10.3906/elk-1909-139.

[67] P. Mathur, R. Sawhney, S. Chopra, M. Leekha, and R. Ratn Shah, "Utilizing Temporal Psycholinguistic Cues for Suicidal Intent Estimation," *Adv. Inf. Retr.*, vol. 12036, pp. 265–271, Mar. 2020, doi: 10.1007/978-3-030-45442-5_33.

[68] M. Francisco and J. L. Castro, "A fuzzy model to enhance user profiles in microblogging sites using deep relations," *Fuzzy Sets Syst.*, May 2020, doi: 10.1016/j.fss.2020.05.006.

[69] Y. Li, L. Yang, B. Xu, J. Wang, and H. Lin, "Improving User Attribute Classification with Text and Social Network Attention," *Cogn. Comput.*, vol. 11, no. 4, pp. 459–468, Aug. 2019, doi: 10.1007/s12559-019-9624-y.

[70] R. Logesh, V. Subramaniyaswamy, V. Vijayakumar, and X. Li, "Efficient User Profiling Based Intelligent Travel Recommender System for Individual and Group of Users," *Mob. Netw. Appl.*, vol. 24, no. 3, pp. 1018–1033, Jun. 2019, doi: 10.1007/s11036-018-1059-2.

[71] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "On the regulation of personal data distribution in online advertising platforms," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 13–29, Jun. 2019, doi: 10.1016/j.engappai.2019.03.013.

[72] K. Zarei, R. Farahbakhsh, and N. Crespi, "Deep Dive on Politician Impersonating Accounts in Social Media," *2019 IEEE Symp. Comput. Commun. ISCC*, 2019, doi: 10.1109/ISCC47284.2019.8969645.

[73] S. Wang *et al.*, "Anchor Link Prediction across Attributed Networks via Network Embedding," *Entropy*, vol. 21, no. 3, Art. no. 3, Mar. 2019, doi: 10.3390/e21030254.

[74] Y. Xu *et al.*, "NeuO: Exploiting the sentimental bias between ratings and reviews with neural networks," *Neural Netw.*, vol. 111, pp. 77–88, Mar. 2019, doi: 10.1016/j.neunet.2018.12.011.

[75] I. Chakraborty, "Hierarchical Bayesian Modeling for Clustering Sparse Sequences in the Context of Group Profiling," Sep. 2018, Accessed: Jul. 16, 2020. [Online]. Available: https://openreview.net/forum?id=SyerAiCqt7.

[76] B. Xu, M. M. Tadesse, P. Fei, and H. Lin, "Multi-granularity Convolutional Neural Network with Feature Fusion and Refinement for User Profiling," in *Information Retrieval*, Cham, 2019, pp. 161–172, doi: 10.1007/978-3-030-31624-2_13.

[77] B. GayathriDevi and V. Pattabiraman, "Towards User Profiling From Multiple Online Social Networks," *Procedia Comput. Sci.*, vol. 165, pp. 456–461, Jan. 2019, doi: 10.1016/j.procs.2020.01.006.

[78] W. Kudo and F. Toriumi, "Sequential User Profiling from Newspapers' Access Log," *Trans. Jpn. Soc. Artif. Intell.*, vol. 34, no. 5, p. wd-E_1-9, 2019, doi: 10.1527/tjsai.wd-E.

[79] Y. Zheng, L. Li, L. Zhong, J. Zhang, and J. Liu, "Using Sentiment Representation Learning to Enhance Gender Classification for User Profiling," *ArXiv181006645 Cs*, Oct. 2018, Accessed: Jul. 16, 2020. [Online]. Available: http://arxiv.org/abs/1810.06645.

[80] Z. Li, B. Guo, Y. Sun, Z. Wang, L. Wang, and Z. Yu, "An Attention-Based User Profiling Model by Leveraging Multi-modal Social Media Contents," in *Cyberspace*

Data and Intelligence, and Cyber-Living, Syndrome, and Health, Singapore, 2019, pp. 272–284, doi: 10.1007/978-981-15-1925-3_20.

[81] S. On-at, M.-F. Canut, A. Péninou, K. Srisombat, and F. Sèdes, “Toward Egocentric Network-based Learner Profiling in Adaptive E-learning Systems: A concept paper,” in *Proceedings of the 2019 7th International Conference on Information and Education Technology*, Aizu-Wakamatsu, Japan, Mar. 2019, pp. 14–19, doi: 10.1145/3323771.3323791.

[82] S. Jiang, X. Chen, L. Zhang, S. Chen, and H. Liu, “User-Characteristic Enhanced Model for Fake News Detection in Social Media,” in *Natural Language Processing and Chinese Computing*, Cham, 2019, pp. 634–646, doi: 10.1007/978-3-030-32233-5_49.

[83] A. Garg, V. Syal, P. Gudlani, and D. Patel, “Mining Credible and Relevant News from Social Networks,” in *Big Data Analytics*, Cham, 2017, pp. 90–102, doi: 10.1007/978-3-319-72413-3_6.

[84] R. Benkhelifa, I. Biskri, F. Z. Laallam, and E. Aïmeur, “User content categorisation model, a generic model that combines text mining and semantic models,” *Int. J. Comput. Sci. Eng.*, vol. 21, no. 4, pp. 536–555, Jan. 2020, doi: 10.1504/IJCSE.2020.106867.

[85] V. Kumar, D. Khattar, S. Gupta, M. Gupta, and V. Varma, “User Profiling Based Deep Neural Network for Temporal News Recommendation,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 765–772, doi: 10.1109/ICDMW.2017.106.

[86] R. Shigenaka, Y.-Y. Chen, F. Chen, D. Joshi, and Y. Tsuboshita, “Image-based user profiling of frequent and regular venue categories,” in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 541–546, doi: 10.1109/ICME.2017.8019330.

[87] G. Kazai, I. Yusof, and D. Clarke, “Personalised News and Blog Recommendations based on User Location, Facebook and Twitter User Profiling,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa, Italy, Jul. 2016, pp. 1129–1132, doi: 10.1145/2911451.2911464.

[88] A. Gorrab, F. Kboubi, H. Ben Ghezala, and B. Le Grand, “Towards a dynamic and polarity-aware social user profile modeling,” in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, Nov. 2016, pp. 1–7, doi: 10.1109/AICCSA.2016.7945626.

[89] “Quality of Service (QoS) and Policy Management in Mobile Data Networks | Ixia.” https://www.ixiacom.com/resources/quality-service-qos-and-policy-management-mobile-data-networks (accessed Dec. 01, 2016).

[90] 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.

[91] V. Agababov *et al.*, “Flywheel: Google’s Data Compression Proxy for the Mobile Web,” 2015, Accessed: Nov. 29, 2016. [Online]. Available: http://research.google.com/pubs/pub43447.html.

[92] 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)*, Dec. 2015, pp. 1–8, doi: 10.1109/Kaleidoscope.2015.7383647.

[93] 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, doi: 10.1109/TNET.2014.2354262.

[94] 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)*, Dec. 2012, pp. 1938–1943, doi: 10.1109/GLOCOM.2012.6503399.

[95] 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)*, May 2015, pp. 224–228, doi: 10.1109/CSCWD.2015.7230962.

[96] S. So, “Mobile instant messaging support for teaching and learning in higher education,” *Internet High. Educ.*, vol. 31, pp. 32–42, Oct. 2016, doi: 10.1016/j.iheduc.2016.06.001.

[97] G. Sun, S. Li, T. Chen, Y. Su, and F. Lang, “Traffic Classification Based on Incremental Learning Method,” in *Advanced Hybrid Information Processing*, 2018, pp. 341–348.

[98] H. R. Loo and M. N. Marsono, “Online network traffic classification with incremental learning,” *Evol. Syst.*, vol. 7, no. 2, pp. 129–143, Jun. 2016, doi: 10.1007/s12530-016-9152-x.

[99] D. M. Divakaran, L. Su, Y. S. Liau, and V. L. L. Thing, “SLIC: Self-Learning Intelligent Classifier for network traffic,” *Comput. Netw.*, vol. 91, pp. 283–297, Nov. 2015, doi: 10.1016/j.comnet.2015.08.021.

[100] R. Patel, D. Downing, M. Covington, M. Covington, and C. Covington, “Dictionary of Computer and Internet Terms Tenth Edition,” vol. Tenth Edition, 2009, Accessed: Jul. 06, 2020. [Online]. Available: https://www.academia.edu/40907623/Dictionary_of_Computer_and_Internet_Terms_Tenth_Edition.

[101] The Jargon File: “user concept definition.” http://catb.org/jargon/html/U/user.html (accessed Jul. 06, 2020).

[102] P. O’Neil, *Database--principles, programming, performance*. San Francisco : Morgan Kaufman Publishers, 1994.

[103] “Network Consultant Definition.” https://www.fieldengineer.com/skills/network-consultant (accessed Jul. 06, 2020).

[104] “Review of business intelligence through data analysis,” *ResearchGate*. https://www.researchgate.net/publication/274060897_Review_of_business_intelligence_through_data_analysis (accessed Jul. 06, 2020).

[105] “Data Analyst job description and duties | Robert Half,” Sep. 11, 2017. https://www.roberthalf.com.au/employers/it-technology/data-analyst-jobs (accessed Jul. 06, 2020).

[106] “What is a Network Administrator?,” *wiseGEEK*. http://www.wisegeek.com/what-is-a-network-administrator.htm (accessed Jul. 06, 2020).

[107] T. Zseby, M. Molina, N. G. Duffield, S. Niccolini, and F. Raspall, “Sampling and Filtering Techniques for IP Packet Selection,” *RFC*, 2009, doi: 10.17487/RFC5475.

[108] “Choosing the Right Clustering Algorithm for your Dataset,” *KDnuggets*. https://www.kdnuggets.com/choosing-the-right-clustering-algorithm-for-your-dataset.html/ (accessed Jul. 07, 2020).

[109] K. Mintwal, “Comparison the Various Clustering and Classification Algorithms of WEKA Tools,” 2014. /paper/Comparison-the-Various-Clustering-and-Algorithms-of-Mintwal/b1fe363c80131e2c5fa2e40c49244e6e377dd30b (accessed Jul. 07, 2020).

[110] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data Via the EM Algorithm,” *J. R. Stat. Soc. Ser. B Methodol.*, vol. 39, no. 1, pp. 1–22, 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

[111] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.

[112] D. C. Corrales, A. Ledezma, and J. C. Corrales, “A Conceptual Framework for Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal”. Telematics Engineering Group, Universidad del Cauca, Campus Tulcán, Popayán, Colombia and Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911, Leganés, Spain, *J. Comput.*, vol. 10, no. 6, pp. 396–405, Nov. 2015, doi: 10.17706/jcp.10.6.396-405.

[113] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991, doi: 10.1007/BF00153759.

[114] A. Cervantes, C. Gagné, P. Isasi, and M. Parizeau, “Evaluating and Characterizing Incremental Learning from Non-Stationary Data,” *arXiv*, Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.06610.

[115] “fitdist function | R Documentation.” https://www.rdocumentation.org/packages/fitdistrplus/versions/0.2-1/topics/fitdist (accessed Apr. 15, 2019).

[116] “Measures of Shape: Skewness and Kurtosis.” https://brownmath.com/stat/shape.htm (accessed Apr. 15, 2019).

[117] “Evaluating Kolmogorov’s Distribution,” *ResearchGate*. https://www.researchgate.net/publication/5142829_Evaluating_Kolmogorov%27s_Distribution (accessed Apr. 15, 2019).

[118] W. W. Daniel, *Applied nonparametric statistics*. PWS-Kent Publ., 1990.

[119] A. Pekar, drnpkr/flowRecorder. GitHub 2020. https://github.com/drnpkr/flowRecorder

[120] “nfstream - a Flexible Network Data Analysis Framework.,” *nfstream*. https://nfstream.github.io/ (accessed Apr. 23, 2020).

[121] J. S. Rojas, jsrojas/FlowLabeler. *GitHub 2020*. https://github.com/jsrojas/FlowLabeler

[122] ntop, “nDPI - Open and extensible LGPLv3 Deep Packet Inspection library.,” Feb. 02, 2012. https://www.ntop.org/products/deep-packet-inspection/ndpi/.

[123] “Labeled Network Traffic flows - 141 Applications.” https://kaggle.com/jsrojas/labeled-network-traffic-flows-114-applications.

[124] “Determining The Optimal Number Of Clusters: 3 Must Know Methods,” *Datanovia*. https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/ (accessed Apr. 23, 2020).

[125] “Cluster Validation Statistics: Must Know Methods,” *Datanovia*. https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/ (accessed Apr. 23, 2020).

[126] D. H. Fisher, “Knowledge acquisition via incremental conceptual clustering,” *Mach. Learn.*, vol. 2, no. 2, pp. 139–172, Sep. 1987, doi: 10.1007/BF00114265.

[127] G. Upton and I. Cook, *Understanding Statistics*, Illustrated. OUP Oxford, 1996.

[128] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J Artif Int Res*, vol. 16, no. 1, pp. 321–357, Jun. 2002.

[129] G. Sun, T. Chen, Y. Su, and C. Li, “Internet Traffic Classification Based on Incremental Support Vector Machines,” *Mob. Netw. Appl.*, vol. 23, no. 4, pp. 789–796, Aug. 2018, doi: 10.1007/s11036-018-0999-x.

[130] P. Chapman *et al.*, “CRISP-DM 1.0: Step-by-step data mining guide,” *SPSS Inc*, vol. 16, 2000.

[131] M. Stewart, “The Limitations of Machine Learning,” *Medium*, Jul. 29, 2019. https://towardsdatascience.com/the-limitations-of-machine-learning-a00e0c3040c6 (accessed May 28, 2020).

[132] S. Zheng, J. J. Lu, N. Ghasemzadeh, S. S. Hayek, A. A. Quyyumi, and F. Wang, “Effective Information Extraction Framework for Heterogeneous Clinical Reports Using Online Machine Learning and Controlled Vocabularies,” *JMIR Med. Inform.*, vol. 5, no. 2, p. e12, May 2017, doi: 10.2196/medinform.7235.

[133] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *ArXiv161200796 Cs Stat*, Jan. 2017, Accessed: May 19, 2020. [Online]. Available: http://arxiv.org/abs/1612.00796.

[134] F. Zenke, B. Poole, and S. Ganguli, “Continual Learning Through Synaptic Intelligence,” in *International Conference on Machine Learning*, Jul. 2017, pp. 3987–3995, Accessed: May 19, 2020. [Online]. Available: http://proceedings.mlr.press/v70/zenke17a.html.

[135] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.

[136] R. E. Schapire, “A brief introduction to boosting,” in *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, San Francisco, CA, USA, Jul. 1999, pp. 1401–1406, Accessed: Nov. 09, 2020. [Online].

[137] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1023/A:1022627411411.

[138] “Fast Training of Support Vector Machines Using Sequential Minimal Optimization,” *ResearchGate*. https://www.researchgate.net/publication/234786663_Fast_Training_of_Support

_Vector_Machines_Using_Sequential_Minimal_Optimization (accessed Apr. 03, 2018).

[139] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, Aug. 1992, doi: 10.1080/00031305.1992.10475879.

[140] P. Joshi and P. Kulkarni, “Incremental Learning: Areas and Methods – A Survey,” *Int. J. Data Min. Knowl. Manag. Process IJDKP*, vol. 2, no. 5, Sep. 2012, doi: 10.5121/ijdkp.2012.2504.

[141] “(PDF) A Survey on Supervised Classification on Data Streams,” *ResearchGate*. https://www.researchgate.net/publication/270787580_A_Survey_on_Supervised_Classification_on_Data_Streams (accessed Apr. 23, 2020).

[142] J. Zhong, Z. Liu, Y. Zeng, and L. C. and Z. Ji, “A Survey on Incremental Learning,” presented at the 2017 5th International Conference on Computer, Automation and Power Electronics, Nov. 2017, [Online]. Available: https://webofproceedings.org/proceedings_series/article/artId/777.html.

[143] Q. Yang, Y. Gu, and D. Wu, “Survey of incremental learning,” in *2019 Chinese Control And Decision Conference (CCDC)*, Jun. 2019, pp. 399–404, doi: 10.1109/CCDC.2019.8832774.

[144] S. Madhavan and N. Kumar, “Incremental methods in face recognition: a survey,” *Artif. Intell. Rev.*, Aug. 2019, doi: 10.1007/s10462-019-09734-3.

[145] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, “Scikit-Multiflow: A Multi-output Streaming Framework,” *J. Mach. Learn. Res.*, vol. 19, no. 72, pp. 1–5, 2018.

[146] A. Chefrour, “Incremental supervised learning: algorithms and applications in pattern recognition,” *Evol. Intell.*, vol. 12, no. 2, pp. 97–112, Jun. 2019, doi: 10.1007/s12065-019-00203-y.

[147] G. Hulten, L. Spencer, and P. Domingos, “Mining time-changing data streams,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, California, Aug. 2001, pp. 97–106, doi: 10.1145/502512.502529.

[148] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, “MOA: Massive Online Analysis,” *J. Mach. Learn. Res.*, vol. 11, no. 52, pp. 1601–1604, 2010.

[149] “(PDF) Online Bagging and Boosting,” *ResearchGate*. https://www.researchgate.net/publication/2453583_Online_Bagging_and_Boosting (accessed Apr. 23, 2020).

[150] A. Bifet, G. Holmes, and B. Pfahringer, “Leveraging Bagging for Evolving Data Streams,” in *Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg, 2010, pp. 135–150, doi: 10.1007/978-3-642-15880-3_15.

[151] H. M. Gomes *et al.*, “Adaptive random forests for evolving data stream classification,” *Mach. Learn.*, vol. 106, no. 9, pp. 1469–1495, Oct. 2017, doi: 10.1007/s10994-017-5642-8.

[152] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, “Learn++: an incremental learning algorithm for supervised neural networks,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001, doi: 10.1109/5326.983933.

[153] R. Lippmann, “An introduction to computing with neural nets,” *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, Apr. 1987, doi: 10.1109/MASSP.1987.1165576.

[154] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[155] “Data Splitting. Z. Reitermanová. Introduction. Cross-validation techniques - PDF.” https://docplayer.net/26609777-Data-splitting-z-reitermanova-introduction-cross-validation-techniques.html (accessed Jul. 02, 2019).

[156] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, and B. Pfahringer, “Efficient Online Evaluation of Big Data Stream Classifiers,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2015, pp. 59–68, doi: 10.1145/2783258.2783372.

[157] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation.” Dec. 2007, [Online]. Available: https://www.researchgate.net/publication/228529307_Evaluation_From_Precision_Recall_and_F-Factor_to_ROC_Informedness_Markedness_Correlation.

[158] “Beyond Kappa: A Review of Interrater Agreement Measures,” *ResearchGate*. https://www.researchgate.net/publication/227685875_Beyond_Kappa_A_Review_of_Interrater_Agreement_Measures (accessed May 19, 2020).

[159] “Scikit-multiflow’s documentation.” https://scikit-multiflow.github.io/scikit-multiflow/.

[160] “Performance analysis of Hoeffding trees in data streams by using massive online analysis framework | Request PDF,” *ResearchGate*. https://www.researchgate.net/publication/288658928_Performance_analysis_of_Hoeffding_trees_in_data_streams_by_using_massive_online_analysis_framework (accessed Apr. 23, 2020).

[161] D. Stathakis, “How many hidden layers and nodes?,” *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009, doi: 10.1080/01431160802549278.

[162] “Adam: A Method for Stochastic Optimization,” *ResearchGate*. https://www.researchgate.net/publication/269935079_Adam_A_Method_for_Stochastic_Optimization (accessed Apr. 23, 2020).

[163] 3GPP, “Policy and charging control framework for the 5G System (5GS).” Sep. 2020, [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123503/16.05.01_60/ts_123503v160501p.pdf.

[164] R. Pathak, “5G Network Architecture-A Beginners Guide | LinkedIn,” 2020. https://www.linkedin.com/pulse/5g-network-architecture-a-beginners-guide-rajarshi-pathak/ (accessed Jul. 30, 2020).

[165] R. Pathak, “Policy and Charging Control in a 5G Network | LinkedIn,” 2020. https://www.linkedin.com/pulse/policy-charging-control-5g-network-rajarshi-pathak/ (accessed Jul. 30, 2020).

[166] “5G download speed is now faster than Wifi in seven leading 5G countries,” *Opensignal*, May 06, 2020. https://www.opensignal.com/2020/05/06/5g-download-speed-is-now-faster-than-wifi-in-seven-leading-5g-countries (accessed Jul. 30, 2020).