POLITECNICO DI TORINO

Collegio di Ingegneria Elettronica, delle Telecomunicazioni e Fisica (ETF)

Master degree course in
COMMUNICATIONS AND COMPUTER NETWORKS ENGINEERING

Master Thesis

User classification in a satellite network for traffic congestion avoidance

Academic Advisor

Prof. Roberto Garello

Company Advisor

Dr. Daniele Capirone

Candidate

Francesco Aquilino
(244011)

October 2019

Credits

Il lavoro di stesura di questa tesi è stato, fino ad ora, uno dei compiti più difficili in cui mi sia mai cimentato. Le settimane trascorse nello sviluppare codice, studiare nuovi argomenti e l'effettiva scrittura hanno richiesto tutte le mie energie ed il mio tempo. Tuttavia, tutto questo è stato reso più agevole da tutte le persone che sono state al mio fianco, che hanno creduto in me e nel mio progetto sia nell'ambito lavorativo che privato.

Desidero ringraziare Skylogic per avermi dato la opportunità di intraprendere il tirocinio nella loro azienda e per aver confermato la loro fiducia in me. Ringrazio il mio tutor aziendale Daniele Capirone per la grande pazienza, comprensione ed aiuto che mi ha fornito sia nelle fasi iniziali, che in quelle di sviluppo, che in quelle finali di stesura e rielaborazione. Ringrazio infine il mio tutor accademico e relatore, professor Roberto Garello, per la fiducia totale che ha riposto in me ed il supporto fornitomi.

I would also like to thank you all the English and non-English speaker who worked with me. Thank you to all the guys of the BB office, Behnaz and Eusem, who listened to my question and provided me everything that was needed in an office, from the work stuff to the funny stuff. Moreover, I wish to thank my colleague, Farzad, we started the journey together and we will proceed (hopefully) side by side. I wish you the best of luck for your future.

In questa sezione, desidero ringraziare le persone che, nonostante l'apparente estraneità alla tesi, hanno contribuito in maniera decisiva al suo completamento. Di fatti, desidero ringraziare tutti i miei amici di Asti: Iacopo, Murch, Axel, Sghe, Ricky, Jack, Lollo, Goggi, Sere, Ire e Vale per le uscite sempre divertenti e le decine di avventure in cui ci siamo cacciati. È grazie a questa spensieratezza che sono potuto arrivare fin qui. Non posso non citare anche tutte le persone che ho conosciuto da quando sono arrivato a Torino che, anche nel loro piccolo, hanno contribuito al raggiungimento di questo traguardo. Grazie a voi: Dylan, Eli, Carmen, Debs, Nina, Luca, Ari e Gaia.

Desidero ringraziare le persone a me più vicine, tutta la mia famiglia, alla nonna Vittoria, che ha sempre creduto in me, sostenuto e appoggiato le mie scelte. Un sentito grazie anche a tutta la famiglia Floris: mi avete accolto in casa vostra come un familiare ed aiutato incondizionatamente. Il vostro supporto è stato cruciale. Un grazie più che speciale ai miei genitori, che mi hanno fornito tutto il necessario, sotto ogni aspetto, e mi hanno consolato, consigliato e spronato in tutto questo. Vi voglio bene.

Infine, non per minor importanza, anzi per l'esatto opposto, desidero ringraziare la persona che ormai considero come la compagnia della mia vita, Irene. Sono ormai 6 anni che condividiamo gioie e tristezze, eventi speciali e quotidianità. Senza il tuo aiuto e supporto, tutto questo sarebbe stato infinitamente più difficile. Grazie per i tuoi consigli sul mio lavoro e per le notti in cui hai ascoltato, compreso ed alleggerito i miei dubbi e le mie preoccupazioni. Un grazie enorme dal profondo del mio cuore.

Summary

List of tables and figures iii
Acronyms v
Motivations vi
Introduction vii

Chapter 1 – Satellite Communication System 1

1.1 - A brief introduction 1
1.2 - Orbits 2
1.2.1 -Kepler's law 2
1.2.2 - Orbit parameter 3
1.2.3 - Orbit categories 4
1.3 - WiMAX 6
1.3.1 - Standard definition 6
1.3.2 - Surfbeam2 implementation 8
1.4 - Frequency usage 8
1.4.1 - C-band 10
1.4.2 - Ku-band 10
1.4.3 - Ka-band 10
1.5 - The control algorithms 10
1.5.1 - Power Control 11
1.5.2 - Adaptive Coding and Modulation 11
1.5.3 - Frequency Tracking 11
1.6 - The Ka-SAT platform 12
Chapter 2 – Machine Learning 14
2.1 - Data manipulations 14
2.1.1 - Space reduction 15
2.1.2 - Features extraction 15
2.1.3 - Principal Component analysis 15
2.2 - Unsupervised learning 15
2.2.1 - K-means clustering 16
2.3 - Supervised learning 17
2.3.1 - Support Vector Machine 17

2.3.2 - Random Forest 17
2.3.3 - Neural Network 18
Chapter 3 – Users identification 20
3.1 - Data formats 20
3.2 - Different ML analysis 21
3.2.1 - Neural Network 23
3.2.2 - Support Vector Machine 25
3.2.3 - Random Forest 28
3.2.4 - K-Means clustering 29
Chapter 4 – The protocol information 35
4.1 - Deep Packet Inspection approach 35
4.2 - Data format 36
4.3 - The classification 38
4.4 - User Features Results 42
4.5 - User Percentage Results 53
Chapter 5 – Identified categories 57
5.1 - Used Metric 57
5.2 – Products analyses 58
5.3 – Populations analyses 63
5.4 – Beams analyses 68
Chapter 6 – Conclusion and future works 75
References 77

List of tables and figures

Figure 1.1.1 - Satellite telecommunications system summary
Figure 1.2.1 (a, b) - Earth orbit satellite parameters
Figure 1.2.2 - Satellites orbit summary
Figure 1.2.3 - Satellites orbit categories
Table 1.3.1.1 - IEEE 802.16 standard parameters
Figure 1.4.1 - Frequency bands descriptions
Figure 1.4.2 – Path loss vs frequency
Figure 1.6.1 – Ka-SAT coverage area
Figure 1.6.2 – Ka-SAT network infrastructure
Figure 2.1.3.1 - PCA objectives
Figure 2.3.1.1 - Hyperplane and support vector representation
Figure 2.3.2.1 – Decision tree example
Figure 3.1.1 – User data and traffic graph
Figure 3.1.2 – Labelling examples
Figure 3.1.3 – Classes histogram
Table 3.2.1.1 – Neural Network 96 sample accuracy
Table 3.2.1.2 – Neural Network 96 sample, bigger dataset accuracy
Table 3.2.1.3 – Neural Network binary quantized accuracy
Table 3.2.1.4 – Neural Network ternary quantized accuracy
Table 3.2.1.5 – Neural Network (AVG-PEAK-PAR) (6h) accuracy
Table 3.2.1.6 – Neural Network (AVG-PEAK-ON) (24h) accuracy
Table 3.2.2.1 – SVM 96 samples accuracy
Table 3.2.2.2 – SVM binary quantized accuracy
Table 3.2.2.3 – SVM ternary quantized accuracy
Table 3.2.2.4 – SVM (AVG-PEAK-PAR) (6h) accuracy
Table 3.2.2.5 – SVM (AVG-PEAK-ON) (24h) accuracy
Table 3.2.3.1 – Random Forest 96 samples accuracy
Table 3.2.3.2 – Random Forest binary quantized accuracy
Table 3.2.3.3 – Random Forest ternary quantized accuracy
Table 3.2.3.4 – Random Forest (AVG-PEAK-PAR) (6h) accuracy
Table 3.2.3.5 – Random Forest (AVG-PEAK-ON) (24h) accuracy
Figure 3.2.4.1 – K-means clustering 96 samples average classes behavior
Figure 3.2.4.2 – K-means clustering binary quantized average classes behavior
Figure 3.2.4.3 – K-means clustering ternary quantized average classes behavior
Figure 3.2.4.4 – K-means clustering (AVG, PEAK, ON) (24h) classes behavior
Table 3.2.1 - Neural Network accuracy
Table 3.2.2 - Support Vector Machine accuracy
Table 3.2.3 - Random Forest accuracy
Table 4.1.1 – Protocol list
Figure 4.2.1 – User Data Format
Figure 4.2.2 – Stacked daily traffic profile

Figure 4.2.3 – Daily evaluated protocol features
Figure 4.2.4 – Percentage of type of traffic performed
Figure 4.3.1 – K-Means clustering metrics comparison varying # of classes
Figure 4.3.2 – Derivative of the metric TWCSS varying the # of classes
Figure 4.4.1 - User Features Class 1
Figure 4.4.2 - User Features Class 2
Figure 4.4.3 - User Features Class 3
Figure 4.4.4 - User Features Class 4
Figure 4.4.5 - User Features Class 5
Figure 4.4.6 - User Features Class 6
Figure 4.4.7 - User Features Class 7
Figure 4.4.8 - User Features Class 8
Figure 4.4.9 - User Features Class 9
Figure 4.4.10 - User Features Class 10
Figure 4.5.1 – User Percentage Class Histogram
Figure 4.5.2 – User Percentage Class Division
Table 4.4.11 - User features categories summary
Table 4.5.3 - User percentage categories summary
Figure 5.2.1 - Product Found Categories Histogram (Percentage)
Figure 5.2.2 - Product 21 and Category 7 behavior comparison
Figure 5.2.3 - Product 41 and Category 10 behavior comparison
Figure 5.2.4 - Product 51 and Category 5 behavior comparison
Figure 5.2.5 - Product Found Categories Histogram (AVG, PEAK, ONTime)
Figure 5.2.6 - Product 81 and Category 4 behavior comparison
Figure 5.2.7 - Product 91 and Category 6 behavior comparison
Figure 5.2.8 - Product 61 and Category 10 behavior comparison
Figure 5.3.1 - Population Found Categories Histogram (Percentage)
Figure 5.3.2 - Population 1 and Category 7 behavior comparison
Figure 5.3.3 - Population 71 and Category 4 behavior comparison
Figure 5.3.4 - Population 51 and Category 3 behavior comparison
Figure 5.3.5 - Population Found Categories Histogram (AVG, PEAK, ONTime)
Figure 5.3.6 - Population 61 and Category 4 behavior comparison
Figure 5.3.7 - Population 11 and Category 9 behavior comparison
Figure 5.3.8 - Population 9 and Category 9 behavior comparison
Figure 5.4.1 - Beam minimum distance category labelling (Percentage)
Figure 5.4.2 - Beam 1 and Category 7 behavior comparison
Figure 5.4.3 - Beam 2 and Category 2 behavior comparison
Figure 5.4.4 - Beam 3 and Category 1 behavior comparison
Figure 5.4.5 - Beam minimum distance category labelling (AVG, PEAK, ONTime)
Figure 5.4.6 - Beam 1 and Category 9 behavior comparison
Figure 5.4.7 - Beam 2 and Category 1 behavior comparison
Figure 5.4.8 - Beam 3 and Category 9 behavior comparison

Acronyms

ATM – Asynchronous Transmission Mode
BCSS - Between Cluster Sum of Square Error
CAC – Connection Admission Control
DHCP – Dynamic Host Configuration Protocol
DPI – Deep Packet Inspection
DVB – Digital Video Broadcasting standard
FTP – File Transfer Protocol
FWC – Forward Channel (Downlink, from gateway to the user terminal)
GEO – Geostationary Earth Orbit
GPS – Global Positioning System
GSO – Geosynchronous Earth Orbit
HTS – High Throughput Satellite
IM – Instant Messaging
LEO – Low Earth Orbit
LOS – Line Of Sight
MEO – Medium Earth Orbit
MF-TDMA – Multi Frequency Time Division Multiple Access
ML – Machine Learning
NN – Neural Network
OAM – Operation And Maintenance
OFDMA – Orthogonal Frequency Division Multiple Access
P2P – Peer To Peer
PCA – Principal Component Analysis
xPSK – x Phase Shift Keying
xQAM – x Quadrature Amplitude Modulation
QOS – Quality of Service
RF – Radio Frequency / Random Forest
RTT – Round Trip Time
RTC – Return Channel (Uplink, from user terminal to the gateway)
SNR – Signal to Noise ratio
(T)SS – (Total) Sum of Square Error to Grand Mean
SVM – Support Vector Machine
VOIP – Voice Over IP
VPN – Virtual Private Network
(T)WCSS – (Total) Within Cluster Sum of Square Error

Motivations

The always growing demand for internet speed and capacity is challenging infrastructure owners. On one side they want to provide a fast and reliable service, on the other side they want to exploit as much as they can the already in place infrastructure and minimize eventual expansion expenses. Sometimes expansion is not possible, like in the case of a satellite link. In this case bandwidth is very limited and its optimization is mandatory since it is a very scarce and precious resource that determines effective final user speed and experience. In this thesis we will explore some way to optimize and manage the available bandwidth. This is done by using at first some QOS principle like classification and resource partitioning.

We will try at first to classify users according to their behavior that is obtained by the data that describes the traffic performed in a day and then by analyzing more deeply the type of traffic performed, with the additional information of the protocols.
To find some correlations between the users, both supervised and unsupervised Machine Learning algorithms are used.

After the classification, we will try to understand which kind of users we have identified and how many we can manage to put on our platform.

The thesis is divided in the following chapters:

The first one will describe the satellite infrastructure that we are working with, the protocol and standards used to communicate, the topology, the bandwidth and the type of modulations and coding that are used.
The second chapter will introduce the Machine Learning topics, giving at first a brief explanation and then focusing on the variety of algorithms that we have used to distinguish among user categories and to recognize traffic patterns.
The third chapter will describe the user data that we have at our disposal, and how we will use it to distinguish among different users. This is done on different time scales, and different data aggregation (different features extraction).
In the fourth one we will add to the user data traffic also the protocol information, used to further describe which kind of behavior a user is following and to characterizing it better.
The fifth chapter will be devoted to analysis of the data and the projections of the found category to different level of aggregation of real data.
The last chapter is aimed at future work and improvement of this work.

Introduction

A satellite network usually relies on one or more satellite to provide a wide coverage of the earth. They can be placed on three main orbits called Low, Medium or Geostationary Earth Orbit (LEO, MEO, GEO), each one having peculiar advantages and disadvantages. However, the available bandwidth is usually the biggest concern of a satellite network planner since it determines the performance and the capacity of a network. An excessive booking of users on a given channel, characterized by a bandwidth, can lead to congestion. This translates in very poor speed, packet loss or in extreme cases to no access at all.

Traffic congestion is a serious threat in a packet network, since it can completely saturate the queues and lead to packet drops. It occurs when the rate at which packets leaving a router is smaller than the rate at which packets arrive in the router. Sometimes, also an equal relation is not enough to avoid it, since an eventual "contention" (when two packets coming from two distinct inputs want to go to the same output, they must be served one after the other generating a temporary queue growth) can lead to buffer space expiration and so packet drops. Therefore, the usual percentage at which a network is loaded is usually around 90%, allowing eventual contentions to be resolved easily. This concept can be summed up by saying that network should be used as far from congestion as possible, but this is in contrast with the one, very beloved by the infrastructure owners, that want the available network resources to be exploited as much as possible.

One powerful set of tools that allow to avoid traffic congestion is the QOS one. As an example, a provider can allow a maximum number of users in the network, depending on the actual network resource status (that can be average speed, peak speed etc.), this is called Connection Admission Control (CAC). But generally, all the algorithms that provide QOS rely on a preliminary classification of the users that allow the provider to know in advance the "expected behavior" of a classified user, usually in terms of how much bandwidth he need, the minimum speed that is needed to support particular applications, the tolerance to delay and the severity of eventual packet losses.

Therefore, the user classification problem is a very important matter in this kind of analysis. We can base our classification, or grouping, using already made categories (like VOIP, streaming, data transferring users, etc.) usually provided in QOS standards (ATM, Ethernet etc.) or use the blind approach where we can try to see if some of our users behave in a sort of similar manner. Classifying a user is very complicated, since it is characterized by a lot of features like average speed, instantaneous speed, activity time, cumulated traffic, protocol with which the traffic was performed etc.

Fortunately, there exists a very wide variety of algorithm that automatically discern and separate users (each one characterized by a personal set of features) in groups that "behave" in a similar manner. This kind of problems are defined as classification ones and they are tackled by a branch of the machine learning defined as unsupervised learning.

Chapter 1
Satellite communication systems

1.1 A brief introduction

Satellite broadband systems are used to provide almost worldwide high-speed internet access using different network segments. They usually employ a communication satellite as a relay between ground stations, receiving a signal transmitted from the ground stations, amplifying it (and optionally also processing it), and retransmitting it back to the same or another portion of the Earth. The information stream usually just passes by the satellite, without being terminated or originated on the satellite itself. In the mid-1960s, satellite communication industry started to appear on the market, and in less than 50 years it became a mainstream technology, offering a wide variety of capabilities in application like voice (at first), data and video to fixed and mobile user in both a point-to-point or broadcast communication. As seen the Figure 1.1, the satellite is the fulcrum of the whole telecommunication infrastructure since it is the place where the information, encoded in different way depending on the data type (source encoding) passes by. Usually the stream of information is originated on the Earth surface (or just above it, in case of planes) and are passed to a terrestrial interface, that generates and modulates the RF radio waves that will be propagated trough the air using a dish antenna. The satellite receives the radio waves that convey the information, and usually either just amplify or eventually process it after reception before retransmitting it to the receiving ground station usually on a different band. The receiving equipment can be either another fixed dish antenna or a mobile terminal that is moving under the satellite footprint.

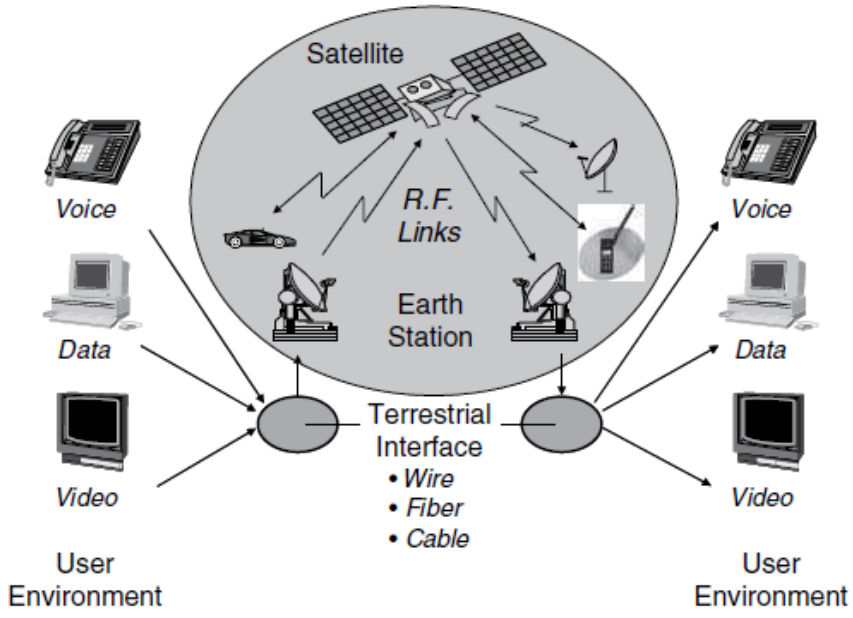

Figure 1.1.1 - Satellite telecommunications system summary (taken from [1], fig 1.1)

1.2 Orbits

The orbit of a satellite is a key parameter since it determines the coverage and operational characteristic of services provided by the system.

The satellites obey to the same motion laws that are followed by the planets. These laws were at first based on the work of Johannes Kepler that derived them from the observations of Tycho Brahe, and then they were refined by Isaac Newton (1687) in his publication "Philosophiae Naturalis Principia Mathematica" that included the Laws of Mechanics and Gravitation.
As described in [1], an orbiting object is subjected to two main forces: orbital velocity, that try to pull the satellite away from the Earth, and the gravity, that try to pull it down on Earth. These two forces can be depicted as:

$$F_{centripetal} = m\frac{\mu}{r^2} \qquad F_{centrifugal} = m\frac{v^2}{r}$$

Where m is the satellite mass, v the satellite velocity, r the distance between the satellite and the center of the Earth and μ, the Kepler's Constant (3.986×10^5 km^3/s^2).

Posing an equal sign between $F_{centripetal}$ and $F_{centrifugal}$ we obtain the velocity required by the satellite to maintain the orbit without either fly to the outer space or collapse to the Earth.

$$v = \sqrt[2]{\frac{\mu}{r}}$$

Note that this discussion doesn't take care of other source of gravity like the Sun, the Moon and other bodies. These are the source of perturbation that need to be counteracted to maintain the satellite in the predicted orbit.

1.2.1 Kepler's law

The Kepler's laws can be applied to any combination of two bodies in the space that are subject to gravity forces. They are:

- **First law**: When an object A revolves around the Earth E it follows an ellipse trajectory where the Earth's center of mass is one of the foci of the ellipse. The size of the ellipse is determined by the speed at which object A travels and its mass.

- **Second law**: An object that is orbiting around the Earth "sweeps out equal areas in equal time intervals", that is to say, when an object is "far" from the Earth it will move slower, sweeping "narrower" sectors, instead when the object is "near" the Earth, it will move faster, sweeping "wider" sectors. In either situation, the areas that it will sweeps will be the same in the same time interval.

- **Third law**: says that 'the square of period of the orbit (T) is proportional to the cube of the mean distance (a) between the two bodies. This concept can be expressed and condensated in the following formula:

$$T^3 = \frac{4\pi}{\mu} a^3$$

This relation also highlights a very important result, that states that the Orbital Radius is proportional to the Orbital Period to the 2/3. This is used to define where to put a satellite to obtain a given orbital period.

1.2.2 Orbit parameters

Orbits are uniquely identified in [1] by the definition of eight parameters, namely:

- **Apogee**
  Represents the farthest point of the orbit from the Earth.
- **Perigee**
  Represents the closest point of the orbit to the Earth.
- **Line of Apsides**
  Is the line joining apogee and perigee passing trough the center of the Earth.
- **Ascending Node**
  Is the point where the orbit crosses the equatorial plane, going from South to North.
- **Descending Node**
  Is the point where the orbit crosses the equatorial plane, going from North to South.
- **Line of Nodes**
  Is the line joining ascending and descending nodes passing through the center of Earth.
- **Argument of the Perigee (ω)**
  Angle from ascending node to perigee, measured in the orbital plane.
- **Right Ascension of the Ascending Node (θ)**
  Angle measured Eastward, in the equatorial plane, from the line of the first point of Aries (Y) to the ascending node.

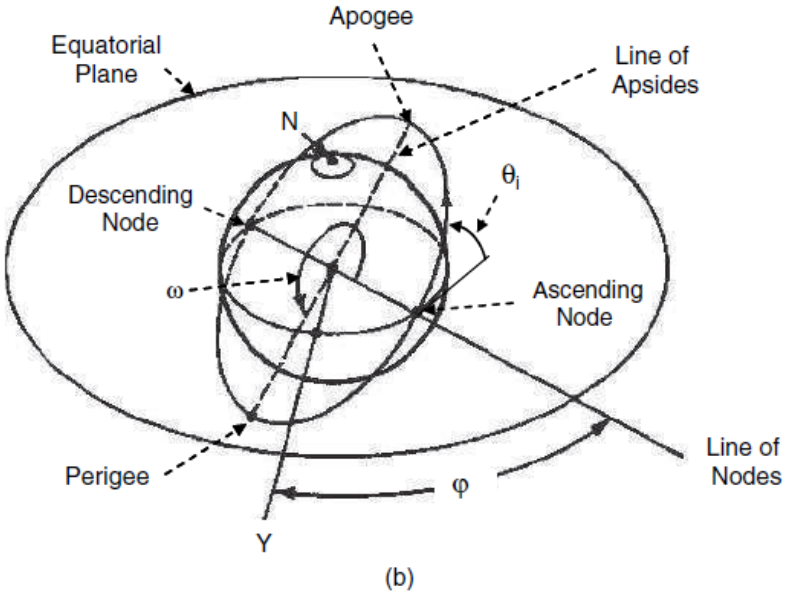

Figure 1.2.1 – Earth orbit satellite parameters (taken from [1], fig 2.4)

The **eccentricity** is another parameter that measures the 'circularity' of the orbit:

$$e = \frac{r_a - r_p}{r_a + r_p}$$

where e is the eccentricity, r_a is the distance from the Earth center to the apogee point and r_p is the distance between the Earth center and the perigee.

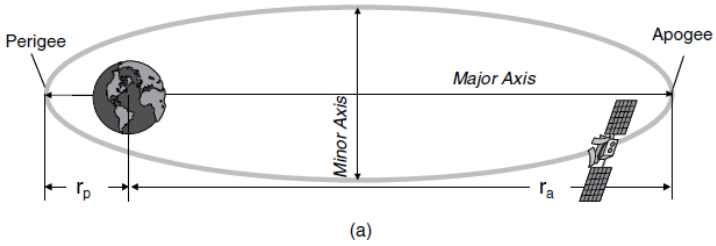

Figure 1.2.1 (continued) (taken from [1], fig 2.4)

1.2.3 Orbit categories

The orbits are also classified depending on the satellite distance from the Earth. They are grouped in three major categories: Low, Medium and Geostationary Earth Orbit (LEO, MEO, GEO).

- **Low Earth Orbit**
  Satellites that orbit at altitude between 160 and 2500 km fall in the categories of LEO satellites. Due to the relatively small distance to the Earth surface they have quite a lot of advantages like very low latency, making them well suited for mobile satellite communications, less path loss due to the shorter path, making possible to employ smaller and low power antenna to overcome path loss, they can cover higher latitude,

they require less energy to be put in orbit. However, they have also some disadvantages like very small footprint, so a "constellation" of them is needed to provide global coverage; they are not geostationary, so they require tracking and handover procedure to maintain a stable connection. Furthermore, as a more technical detail, their orbit drift westward of several degree per day, due to the non-spherical shape (oblateness) of their orbit. A very common fix is to give to the orbit an inclination of 63° which balances all the rotational forces and keep the major axis fixed. They are used as a relay in terrestrial mobile communication and low budget applications.

- **Medium Earth Orbit**
  In this category fall all the satellites that orbit at a geostationary height between 10000 and 20000 km and, being the category that is in between all the other, is called MEO. They have some desirable features, like the fact that they hover on the same ground periodically, and their observation time is around 1-2 hours. They are particularly used for positioning and navigation systems (like GPS satellites) and meteorological and remote sensing applications.

- **Geostationary Earth Orbit**
  It is the most popular one used by communication satellites where the revolution period is chosen to be the same as the Earth rotation period. Its eccentricity is equal to 0 (circular orbit) and the inclination angle with respect to the equatorial plane is 0° (they are located on the equatorial plane) nominally at 'geostationary height' of around 36000 km (height from the Earth surface considering an Earth radius at the equator of 6378 km and mean sidereal day of 86164,09 s). This orbit however is an ideal one, practically unachievable in real life due to other bodies perturbation in the gravity field, that would require higher fuel consumprtion to maintain it, resulting in a shorter satellite operational life.
  Hence, another more feasible orbit is used, known as **geosynchronous earth orbit (GSO)** that permits an eccentricity and inclination angle greater than 0. It usually does not require Earth Tracking and the coverage area is around 120° degree of the Earth, so as little as three GSO satellites are enough to provide global coverage (Poles area excluded).
  These orbits advantages are the already mentioned no need of Earth tracking, since they are fixed in the sky; the fact that the can provide global coverage with very few satellites and, ultimately, the easiness in computing the path loss, since the path distance does not change (just the eventual atmospheric event fade need to be taken in account as a worst case scenario).
  The downside depends on the very high distance from the Earth, that causes a very long RTT from an Earth station, a very high either fuel or time consumption to be put into the orbit and, unsurprisingly, being the most popular and convenient orbit, it is very crowded.

	LEO	MEO	GEO/GSO
Eccentricity	Variable	Variable	0°/~0°
Observation time	8-15 mins	1-12 hours	24 hours/day
Inclination angle	Variable (usually 63°)	Variable	0°/~0°
Geostationary height [km]	160-2000 km	10000-20000 km	36000 km
RTT	Very small (~1-13 ms)	Medium (~30-60 ms)	High (~260 ms)
PROs	-Low launch cost -Small RTT -Small Path Loss -Small antenna needed -Low power	-Possibility of hover over the same ground each day -Good compromise between RTT and launch cost	-Appears as fixed in the space -Provide global coverage with small number of satellites (3) -Fixed path loss
CONs	-Global coverage need high number of satellites (constellation) -Handover procedure needed -Shorter lifespan	-RTT not negligible anymore -Higher Path Loss	-RTT very high -Very high path loss -Requires big antenna and high power

Table 1.2.3 – Satellite orbits summary

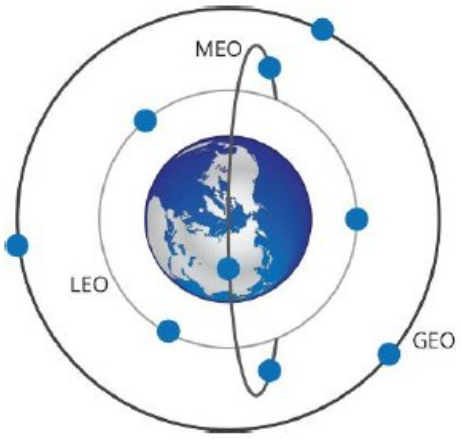

Figure 1.2.4 - Satellites orbit categories (taken from [11])

1.3 WiMAX standard

WiMAX is a set of wireless broadband communication standards based on IEEE 802.16 that can adapt to multiple physical layer and Media Access Control options to provide connectivity everywhere. In this short section we will give just some hint necessary to understand the following description of the satellite network. Therefore, just a subset of entities and aspects will be investigated and described.

1.3.1 Standard definition

The WiMAX structure is based on the IEEE 802.16, 802.16a and 802.16e standard that regulates spectrum, modulations, bandwidths, bit rates and other parameters as summed up in the table below and where extracted by [2].

	802.16	***802.16a***	***802.16e***
Spectrum	10-66 GHz	2-11 GHz	2-6 GHz
Channel Bandwidth	20, 25, 28 MHz	1.5-20 MHz	1.5-20 MHz with UL sub channels
Modulation	QPSK, 16QAM, 64 QAM	OFDM 256 sub carriers QPSK, 16QAM, 64 QAM	OFDM 256 sub carriers QPSK, 16QAM, 64 QAM
Bit Rate	32-134 Mbps (28 MHz)	75 Mbps (20 MHz)	15 Mbps (5 MHz)
Channel conditions	LOS	Non-LOS	Non-LOS
Typical cell radius	2-5 Km	7-10 (max 50) Km	2-5 Km
Application	Fixed	Fixed and Portable	Mobility

Table 1.3.1.1 – IEEE 802.16 standard parameters (Taken from [2], table 1.2)

The main entities that form the networks are:

- **Subscriber Station (SS) / Mobile Station (MS)**
  This term indicates the equipment usually located at the end user facility (also known as Customer Premises Equipment, CPE for short). They can be located inside a building or outside, taking the name of indoor or outdoor CPE. The indoor ones have the advantages of usually being easy to install by the user itself, while the outdoor ones have better communication performances, but they require specialized technicians to be installed and configured.

- **Base Station (BS)**
  This equipment provides connectivity between users CPE and core network. It manages the function of DHCP proxy, radio resource management, service flow manager, key management, authentication relay. It also takes care of handovers in case of mobility terminals.

- **Access Service Network Gateway (ASN - GW)**
  This device operates as a layer 2 switch between the front and back haul. It also acts as a traffic aggregation point. It provides the functionality of intra-AS communications, Connection Admission Control, Authentication, Accounting and Authorization, QOS policy applications like policies compliance check and routing functionalities.

- **Authentication, Authorization and Accounting Server (AAA Server)**
  This server, as the name describe, provide authentication capabilities and checks, manages if a user is eligible to be authenticated in the network and accounts for the traffic that the users perform in order to check if they are compliant with their service plan contracts.

1.3.2 Surfbeam2 implementation

The Surfbeam2 is a proprietary technology of the provider ViaSat that slightly modifies the WiMAX standard to better adapt it to satellite communications.
It uses the same MAC and it adheres to the following IEEE protocols: 802.16, 802.16a, 802.16b, 802.16c, 802.16e, 802.16f, 802.16g and 802.16h that state the types of communications, the frequency ranges, the modulations types, code rates and symbol rates.
Regarding the PHY layer, it employes the same waveforms as the one included in DVB-S2 standard, exploiting OFDMA on the downlink and MF-TDMA on the uplink.

The entities envisioned in the WiMAX standard change name and in the Surfbeam2 became:

WiMAX	**Surfbeam2**
Subscriber Station (SS)	User Terminal (UT)
Base Station (Bs)	Mac Processing Sub System (MPSS)
ASN – GW	ASN – GW
AAA Server	AAA Server

1.4 Frequency usage

The Radio Frequency (RF) part is one of the fundamental of a communication satellite on which the performances of the satellite are determined. It exploits the RF spectrum as the medium through which information's are conveyed. The part of the spectrum usually referred to as the radio wave part is between 100 MHz and 100 GHz as described in figure 1.4.1. Satellite communications usually operates around three main frequency band called C, Ka, Ku bands using the IEEE standard. Each band has its advantages and disadvantages, like antenna seize, presence of noise, bandwidth etc. The free space loss depending on the frequency is described by the figure 1.4.2, that shows peak of absorption at certain frequencies.

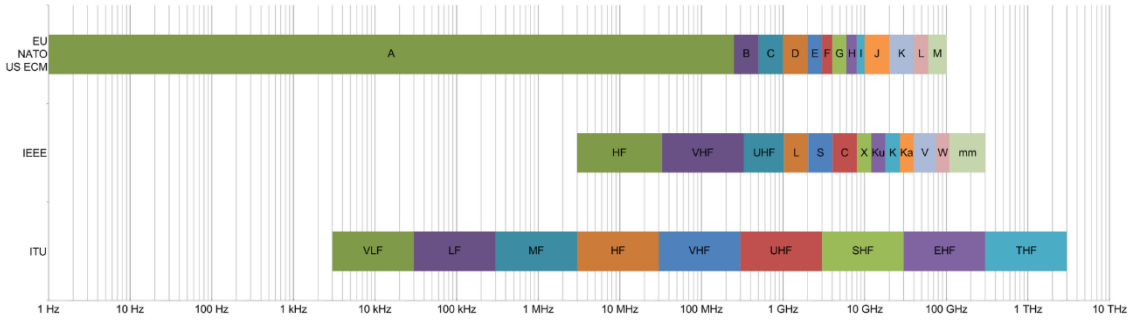

Figure 1.4.1 – Frequency bands descriptions (taken from [12])

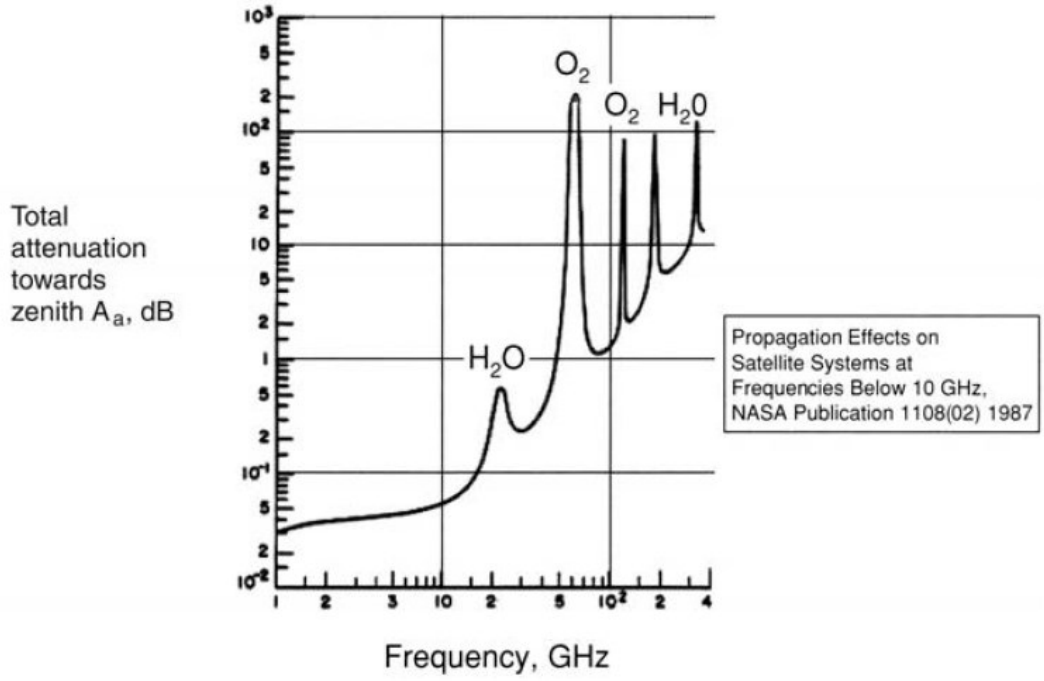

Figure 1.4.2 – Path loss vs frequency, (taken from [1], fig 4.15)

1.4.1 C band

The C-band is the historically older communication band used and lies in the interval 5.85 -8.2 GHz. Its wide usage is due to the very low attenuation both due to atmospheric event and free space path loss, and initially, less crowded than lower frequency bands, like the L band. It was the first used since in the 1990s, technology to communicate at this frequency band was already available, so no advance in technology was needed to make it sprout. The lower frequency, with respect to the other bands, makes the beam width very wide, making the usage of this band suitable for broadcast communication like TV-service, and opens to new problems like EM compatibility and frequency coordination among the operators. An additional downside is the need of large parabolic antenna diameter at the Base Stations to receive a good signal power (in the order of 30 meters) and also on the Subscriber Station (around 3 to 4 meters).

1.4.2 Ku-band

The Ku-band is located around 12 – 18 GHz and is used for satellite communications. Some portions of this band are not shared with other applications, so power restrictions aren't a problem. This translates in an antenna dish size that can be smaller, with obvious advantages from the users since antennas requires less space, and for the provider, since smaller antenna usually are cheaper to be manufactured. Furthermore, the usage of short wavelength permits to obtain a higher angular resolution, as described by the Rayleigh Criterion that links the diameter of the antenna dish with a given angular beam width to the wavelength in a directly proportional way. So, using higher frequency allow to tighten the beam size and reduce interference from other sources. Actual size of the parabolic antennas is in the order of less than 1 meter in diameter. Furthermore, this band is not afflicted by rain attenuation rather than other higher frequencies. Despite this, some attenuation effect takes places around the 10 GHz where the absorption peak due to liquid water occurs. The snow instead is usually not a problem, apart from eventual accumulation of snow or ice on the antenna dish that alters its focal point.

1.4.3 Ka-band

The Ka-Band is a portion of the spectrum around 26.5 – 40 GHz. This band is much more free than Ku band and inherits all its benefits, like no power restriction and smaller antenna needed. Furthermore, wide ranges of spectrum are available since there aren't so many applications using this band. Incidentally, the atmospheric attenuation is much higher, and it is more vulnerable to rain fade. Precisely, there is water vapor resonance at 22.24 GHz. Furthermore, recent 5G standards plan to use parts of this band for terrestrial communications. Due to the usage of these bandwidths, the satellite link is not reliable in case of rain fade. So, power control and adaptive coding algorithm are needed to provide a good reliability of the channel.

1.5 The control algorithms

In this section we will describe the techniques used to counteract the fades that occurs along a satellite links, ranging from not only rain, to haze and snow fade but also wrong pointing of the user's antenna.

1.5.1 Power Control

The power control algorithm is based on previous evaluations of the link budget in clear sky conditions. The SS periodically transmits a message (called Periodic Management Message, PMM) containing its received SNR from the satellite. In the meanwhile, also the BS checks the link SNR between BS and satellite. An optimal power is established, and it is communicated to both the BS and SS. The latter, in case of need, tries to increase the power as much as they can until a maximum power related to the input back-off of the Power Amplifier is reached. This is done to avoid incurring in Non-Linear behavior like spectrum enlarging and constellation distortions.

1.5.2 Adaptive Coding Modulation

This algorithm is used to provide as fast as possible the best possible speed sustainable by a SS, as well as the best possible link quality in case of bad weather. The SS monitors the SNR of the received link, and in case of very high signal quality either increase the coding rate, reducing the redundancy bits, or increase the modulation cardinality, passing as an example from a BPSK to a QPSK. If a low SNR is detected, the power control algorithm tries to compensate as much as it can, and if it eventually runs out of power, the ACM algorithm starts as a first counter measure and, if it is not enough, it decreases also the modulation cardinality and coding rate. This will increase the possibility of maintaining a reliable link in case of bad weather events, and partially cancels out the possible outage that can occur due to the frequency bands that are used.

1.5.3 Frequency Tracking

In this part we will give a hint on how the frequency tracking algorithm works, since even if the satellite should remain fixed in the sky, some shifts in frequency still occurs. The most common sources of frequency error are system oscillators not synchronized between Tx, Rx and the Satellite, doppler frequency shifts and scintillations in the ionosphere, that plays an important role in shifting the carriers differently depending on the time of the day and geographical positions. As a rule of thumb, the sum of these errors must be smaller than the demodulator tracking capabilities (usually in the range of hundreds of Hz) and does not need to be exactly zero. One possible tool to recover the frequency and phase offset is the Costa's loop, that can be configured to be of the first, second or third order, obtaining more and more capabilities of following frequency drift but requiring much more complex systems. In the Surfbeam2 a closed loop system between GS and SS is instantiated, over which periodic burst of clean carriers at some predefined frequency are transmitted to allow both ends to evaluate the frequency shift. Two loops are needed since downlink and uplink transmissions occurs on different bandwidths.

1.6 The Ka-SAT platform

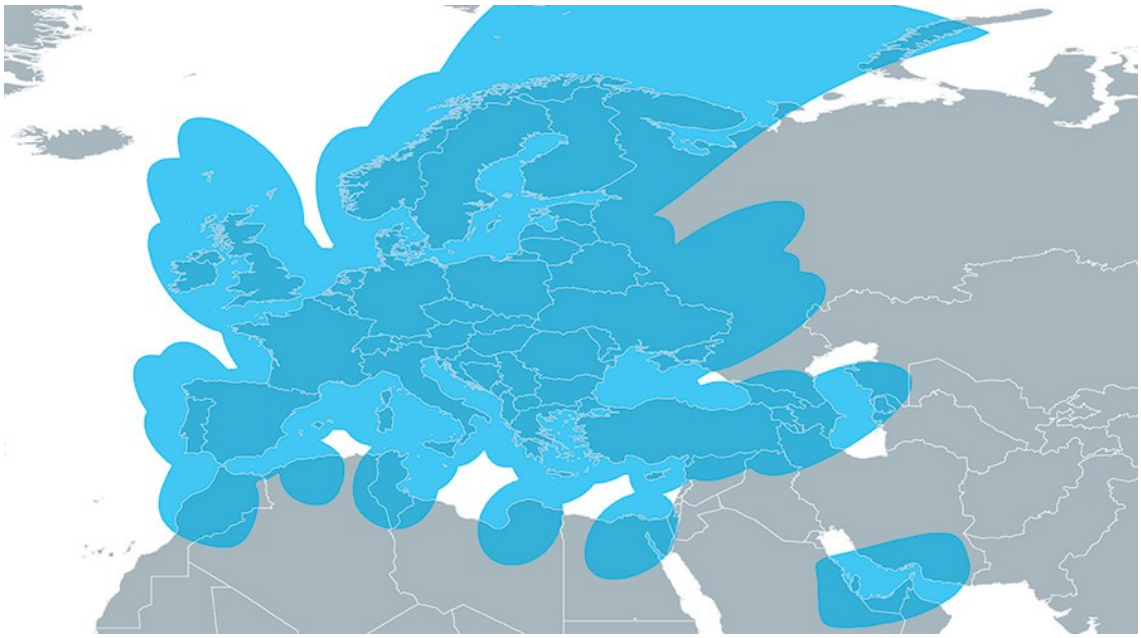

Figure 1.6.1 – Ka-SAT coverage area (taken from [13])

The Ka-SAT is a broadband telecommunication platform that provides high speed internet access in many regions of the Europe, North Africa and Middle East.
Being based on a satellite link is composed by three segments, namely the Ground, the Space and the Control one.

The ground segment is composed by an optical fiber based back-bone that interconnects the Base Stations to the Core Nodes and is based on the Multi-Protocol Label Switching (MPLS) technology. To support a high service availability several redundancy precautions have been put in place:

- **8 Base Stations** (+2 redundant) provide the connectivity between the User Terminal (SS), passing through the Satellite, and the Internet, passing through the back-haul.
- **2 Core nodes** (+1 redundant) provide the Management, the Routing, the AAA services, Security, Peering and Shaping

Every BS is connected to the Core nodes using a double star topology to avoid single point of failures.

The space segment is composed by just one High Throughput Satellite called Ka-SAT. The Ka-SAT satellite operates as a transponder between the Subscriber Station (SS) and the network Base Station (BS). As the name suggests, it operates in the Ka-band (26.5 – 40 GHz) and it is based on a multi-spot beam technology that gives the possibility to handle high throughput over a wide area. The multi-spot approach exploits a frequency re usage strategy that minimize adjacent beam interference, using combinations of different frequency band and different polarizations. Each couple of beams is amplified

using a Travelling Waves Amplifiers that increase the signal power of up to 60 dB to overcome the path loss between the Satellite and either the BS or the SS. Its total aggregated throughput reaches the astonishing speed of 90 Gbps.

The control segment is based on one central Network Operatic Center (NOC) based in Turin, that keeps track of the satellite status and operational functionality. It is composed by several team that deals with each part of the platform, ranging from the *Radio Frequency* team, that keeps the satellite in the correct orbit, adjust the amplifiers gain, adjust the BS antenna pointing and monitor the correct functionality of the analog part of the transmission chain; the *Base Band* team, that takes care of the correct modulation, demodulation, ADC and DAC conversion; the *Network* team that guarantee the connectivity between the Core Node and the Base Station and in general to every other server, machine, interface of the platform; the *Developer* team, that ensure the correct functionality of the database, web tool and improve them etc. .

The network topology is inspired to the previously described WiMAX one is described in the figure 1.6.2 where the entities and their interconnection can be observed.

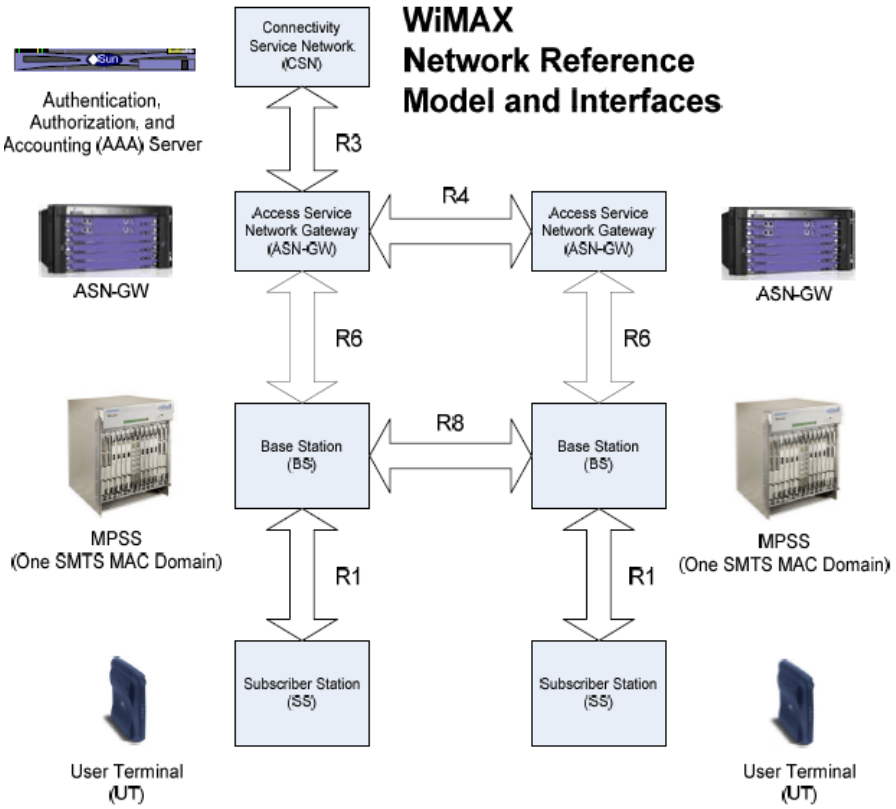

Figure 1.6.2 – Ka-SAT network infrastructure

Chapter 2
Machine Learning

The recent explosion in data availability called for new methods to process them. Helped by the ever-increasing computational power of the processors, machine learning has seen a new dawn in the last years, after some decades in the shadows. In fact, machine learning is around since the 1970, when some initial ideas were developed and perfectioned, but at that time, it required too much time for the task to be performed by a computer.
The fields over which we can apply this kind of algorithm are countless, ranging from image analysis, that automatically extracts the information of some shapes, like the auto detection of eventual diseases from some x-rays, object recognition, to catalog the images that contain a specific object or to recognize which object is present in the image, to deep learning, that automatically figures out the very complex relation that links the input to the output, like a written digit extracted from a picture.
The list could be extended much more, but the core idea is the same over every application: the machine automatically learns from the training data how to complete the task for which it was created for. Other data will be later fed to the algorithm, in order to evaluate the ability of the machine in generalizing the learned pattern to never seen data.

The machine learning algorithm are usually divided in two main categories, as described in [3]:

- **Supervised (Predictive)**
  This kind of algorithm learns how to give the correct outputs by looking at training set composed of input data and given output labels. Since the correct answers are already provided, the error metric can be easily computed both on the training data set and validation data set.
- **Unsupervised (Descriptive)**
  This algorithm tries to figure out possible aggregation of data looking just at the input data, so no output labels are given and, therefore, no easy error metrics can be defined.

In the following chapter just few of them will be described, in particular the ones used in this thesis work.

2.1 Data manipulations

The input data are the only source of information out of which the algorithm can extracts the patterns and learns how to perform the task for which it was created for. Therefore, they are very precious and requires specific treatment depending on the application field to be correctly “processed” by the machine. In this thesis work different manipulations were necessary in order to explore all the possible results that the different machine learnings provided.

2.1.1 Space reduction

Usually the data that are extracted from raw measurements present a very high dimensionality. Many different parameters can change randomly making the work of the learning very hard and difficult. This effect can be resumed in the "curse of dimensionality" described in [2], where we describe the trade off between having a very high number of parameters to better understand the input and having just the usually very few important parameters as an input. Different techniques were envisioned in order to reduce this effect out of which we can describe the Principal Component Analysis.

2.1.2 Features extraction

As described before, the extraction of the important features is a crucial step to obtain a good and trained machine learning algorithm. In this thesis work it was decided to either average out or to quantize the input to obtain the so-called features extraction.

2.1.3 Principal Component Analysis

The PCA is a way of reducing the number of dimensions of the inputs. Its behavior is very simple to understand: the algorithm tries to find new basis over which the data can be represented. The objective is to redistribute the variance of the data over the dimension in an equal way and minimizing the distance between the new basis and the input data, eventually also ignoring some of the dimensions. If the input data are highly correlated, very few dimensions are given as an input that still describe the input, otherwise, if the input data are almost independent, the application of the PCA results in an output with the same number of dimensions.

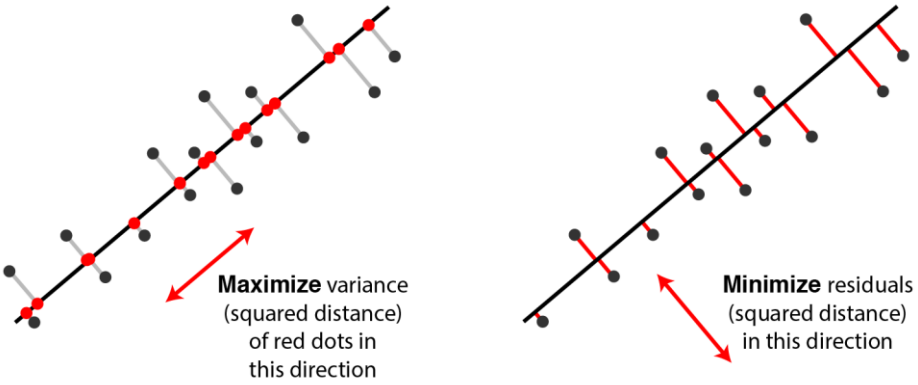

Figure 2.1.3.1 – PCA objectives (taken from [14])

2.2 Unsupervised learning

As previously described unsupervised learning deals with just input data without having given output labels. Its focus is on finding correlation, grouping, clustering and dimensionality reduction.

2.2.1 K-Means clustering

This algorithm, as the title suggests deals with finding clusters in which encase the input according to their similarity. This similarity is usually evaluated according to the "coordinates" over the dimensions of each point.
The actual functioning is based on a procedure in which the user provides the amount of looked for clusters K to the algorithm, the algorithm initially places the centroids of each this cluster randomly, assigns each input point to the nearest centroid cluster, and iteratively moves the centroids according to the mean of all the point assigned to that cluster, updating the assigned cluster for each point. The algorithm stops either because the centroids position did not update with respect to the previous iteration or because a predefined number of steps has been reached.

More formally, in each iteration we proceed to:

1. Assign each point to the nearest cluster centroid

$$\forall\, i = 1 \dots N\,, choose\ j \mid D_{i,j} = \left|x_i - \ c_j\right| \ is\ minimum$$

2. Update the centroid position based on the mean of assigned points

$$\forall\, j = 1 \dots K\,, move\ c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_{i,j}$$

Where:

- x_i is the input point i
- N is the cardinality of the input points set
- c_j is the centroid of cluster j
- K is the total number of searched clusters
- N_j is the total number of points belonging to cluster j
- $x_{i,j}$ is the i^{th} point that belongs to class j

The choice of K is deliberately leaved to the user choice, that, according to some heuristic, can choose the considered right number of clusters. As an example, the "elbow method" can take into consideration different metrics like the average distance between a cluster center and the point belonging to that cluster, the distances between clusters etc.… and stop as soon as the improvement decrease below a certain threshold, that can be represented as an "elbow" curve where on the x axis we have the number of chose cluster and on the y axis the considered metric. The used metric is described in the fourth chapter, where this heuristic was applied.

2.3 Supervised learning

In contrast with the other type of learning, supervised uses input data and given output label to train the model to perform different task like classification, prediction and recognition.

2.3.1 Support Vector Machine

This algorithm, as described in [5], can separate the input vector in two classes looking for the hyperplane that better separates the inputs along the many dimensions. The algorithm iteratively tries to find this hyperplane trying to maximize the so-called margin, that is simply the distance between the points of one cluster and the points of the other cluster. The support vectors are simply data points that are closest to the found hyperplane. It is mostly used for classification and regression.

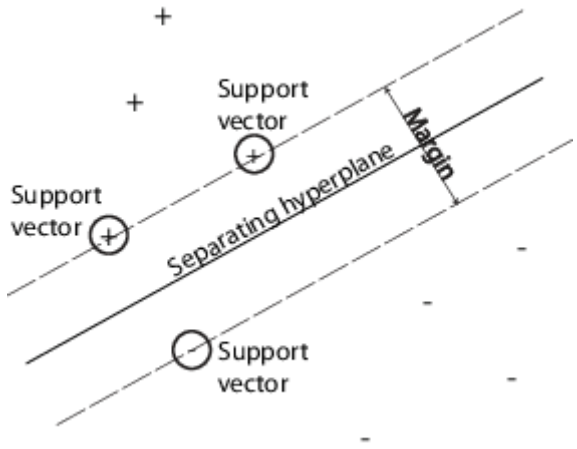

Figure 2.3.1.1 – Hyperplane and support vector representation (taken from [15])

The kernel, that is to say, the function used to find this hyperplane can be of many forms, depending on the way in which the separation is evaluated. In this thesis work, the linear, quadratic, cubic and different level of Gaussian were used.

2.3.2 Random Forest

This algorithm, as the name could hint, is composed by many decision trees. The main details on the working principles were extracted by [4], and we provide here a short summary. A decision tree works by taking iterative decisions and, depending on the answer, continue on one branch or to the other, in case of binary decision, or to one of the others, in case of multiple possible answers. This sequence of decisions can be represented as a decision tree where the root is the initial set of observations, the nodes are the decision points, and the branches are the possible choices taken. In general, the choice of the decision metric is based on the maximization of both separations between two different branches and the likelihood of the observations in the same branch.

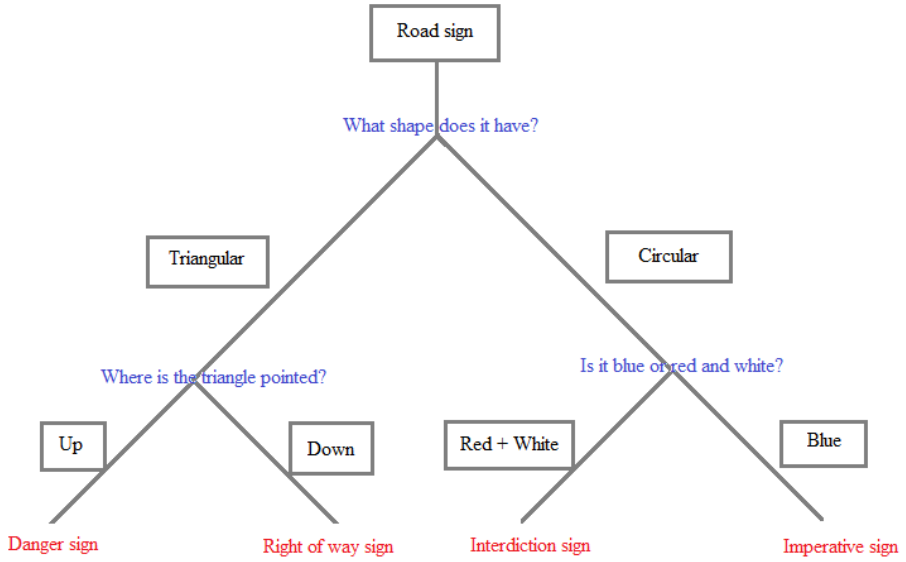

Figure 2.3.2.1 – Decision tree example

A random forest is simply a collection of decision tree that works all together in an ensemble way. The ensemble way of operating consists in giving the same observations set to every decision tree, each one characterized by slightly different decision rules, and choosing as output the output that was obtained by the majority of the trees.
Making the trees work together, but independently one from each other is what gives to this model the powerful advantages of not being biased to any particular output. In fact, the final decision is protected by a single tree error, since the majority of the other should compensate for it.

2.3.3 Neural Network

Neural networks algorithms are based on a collection of artificial neurons, called perceptron, that fires or not depending on specific values, weight and activation functions. They are able to learn and recognize pattern in the input data and are usually used when a very difficult and not easily obtained function between the input and output is present. There exists a wide variety of them, starting from the classic feed-forward, going to the convolutional and finishing with the adversarial.
The network is structured in layers, commonly divided in three main types: the input one, where the input data are put; the hidden one (or ones), where the perceptron are put and where all the evaluation are made; and the output one, where the final guess is outputted either in an probabilistic way, assigning values near to 1 to more confident answer and more near to 0 to less confident answers, or in an “hot vector” fashion, where a 1 means present and 0 means absent.

It is not the scope of this paragraph to explain in detail how a neural network is made and works, but it was preferred to focus on the parameter over which a network can be built and how it will affect its performances.

The first parameter is the depth, that characterizes how many hidden layers are present between the input and the output. The second parameter is the amount of perceptron present in a single hidden layer, usually configured in a heuristic way considering the input and output dimensions. The third parameter is the activation function, that explain how the single perceptron will behave according to the input that it will receive. The last parameter is the learning rate, used to define how much we will modify our values in the learning phase in order to obtain (hopefully) better performances.

The way in which a neural network learns can be easily resumed in the following steps:

- The input data are fed the to input layer, that forward propagates their value to the hidden layer(s).
- In each hidden layer, each perceptron receives every (or just some) of the previous layer values, multiplying each of them according to a weight. Then, a bias is summed, and the result is given to a non linear function, called activation function, that decide what will be the output for the next layer.
- When the output layer is reached, the obtained output is confronted with the expected output and an error is evaluated (usually the MSE error, but also other loss function can be used like the Cross-Entropy function).
- After the error is evaluated, the gradient of this error is calculated and a correction in the direction of a smaller loss function is back propagated to the hidden layers.
- Using the previously evaluated corrections, the weight and the biases are adjusted using the correction multiplied by the learning rate.
- The correction is then iteratively applied to every hidden layer in a backward propagating way.

Usually a neural network is trained until a either a desired error performance is obtained, or a predefined number of iterations are performed. However there exist some reasons why the learning can slow down or either happen in a wrong way.
One of them is called gradient vanishing and occurs when a network has many hidden layers: due to the way back propagation works, the gradient is doomed to be smaller and smaller going through the back propagation. This results in a slower learning for the initial hidden layers with the respect to the ones more near to the output.
Furthermore, another bad effect can take place when the network became very good at recognize perfectly just the training data and lose the ability to generalize to unseen new data. This effect is called over-fitting, and usually is tackled by partitioning the input data in three different sets: the training one, used to make the network adjust their weights and biases, the validation one, used to prevent the network to overfit by stopping the learning procedure, and the test one, where the network performance in recognize never seen input is evaluated.

Chapter 3
User identification

This chapter will describe the results obtained by analyzing the data of the users, at first using raw data, then using refined data and finally extracting some features out of them.

3.1 Data formats

To describe the users, the data collected by the company during normal monitoring and accounting was used. Each user was characterized by its Account ID, its Population ID, the Forward Link Speed and the Return Link Speed. In fact, these last two measurements quantified the amount of data exchanged by users during the days.
The data were collected in a database, aggregated under different time scale (to ease and speed up database queries), starting from 1 minute (360 samples per day) to 4 hours (4 samples per day).
The time scale mostly used in these analyses was the 15 minutes one, which implied that a user day traffic graph was characterized by 96 sample per day per channel (Forward and Return).

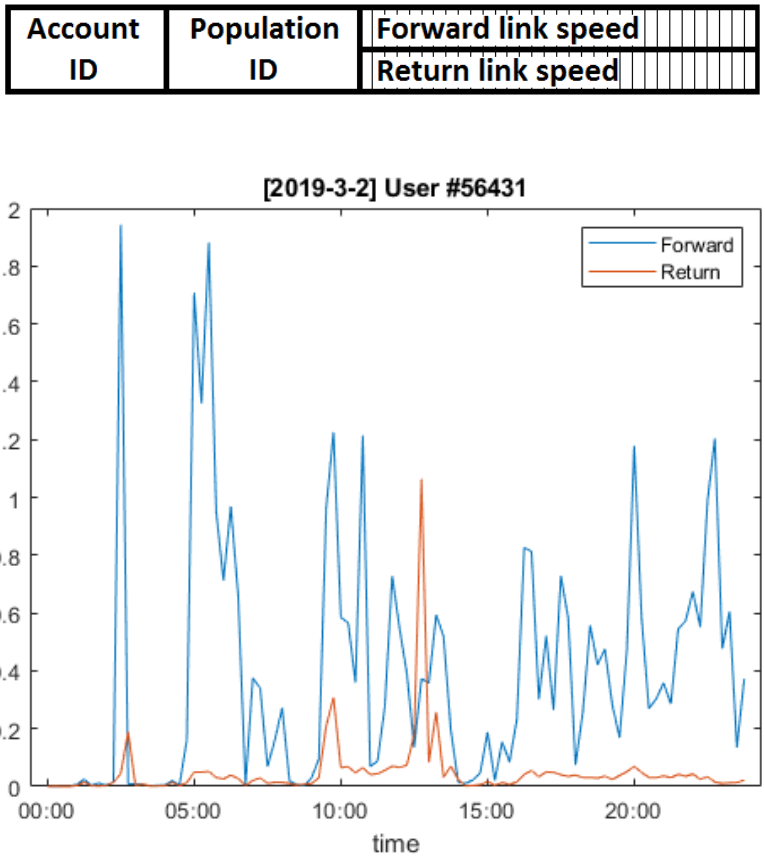

Figure 3.1.1 – User data and traffic graph

The available amount of data was very huge, so just some single days coming from a week (02-03-2019 → 09-03-2019) were used.

Proceeding in the analysis data manipulations were needed, to reduce data dynamics and to better represent the users. So, data were "quantized" in two levels using a threshold to distinguish between active and silent users. The threshold was set to 50 Kbps, considered enough to avoid false activation due to control traffic of the network.

In the last part, also a quantization on three level was considered, allowing to a better description and distinction between silent user, slowly active users and fast active users. The thresholds were empirically set to 50 Kbps and 3 Mbps

Finally, out of the previously averaged data, some features were extracted to better represent users. Averaging over 6 hours or 24 hours windows was used to condensate this information.

Two sets of features were used in the analysis:

- The first set consisted in
  - Average speed [b/s] $AVG = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - Peak speed [b/s] $PEAK = max(x_i) \quad for\ i = 1....N$
  - Peak to Average Ratio [dB] $PAR = 10\log(\frac{PEAK}{AVG})$

- The second set was composed by
  - Average speed [b/s] $AVG = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - Peak speed [b/s] $PEAK = max(x_i) \quad for\ i = 1....N$
  - ON time [%] $ON = (\frac{1}{N}\sum_{i=1}^{N} \dot{x}_i) * 100$
    where $\dot{x}_i = 1\ if\ x_i \geq threshold$
    $\dot{x}_i = 0\ if\ x_i > threshold$

3.2 Different ML analysis

In the same way as mentioned in [9], the previously descripted manipulated data was fed to a set of machine learning algorithms to evaluate their ability to discern between category and to see how well they were able to spot some uneasily seeable relationship between the users.

The tested Machine Learning algorithm came from both the two know categories of supervised and unsupervised learning and were:

- Supervised
  - Neural Network
  - Support Vector Machine
  - Random Forest
- Unsupervised
  - K-Means clustering

Due to the nature of the supervised algorithm it was needed to label each input to provide a reference over which the algorithm could learn. So, using a 6-hours window convention, an automated algorithm for labelling was envisioned. The 6-hours window were called, for simplicity, Night (0-6), Morning (6-12), Afternoon (12-18), Evening (18-24). In this way a 4-bit label was enough to characterize a full user day (as an example, a 0110 user, labeled 6 was active in the Morning and in the Afternoon). A user was considered active as soon as it exceeded the 50 Kbps threshold on the FWC. An example can be seen below

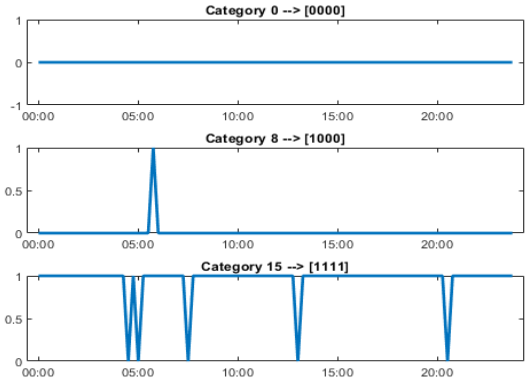

Figure 3.1.2 - Labelling examples

To better understand the general behavior of the found labels, a preliminary histogram of the 16 categories was evaluated.

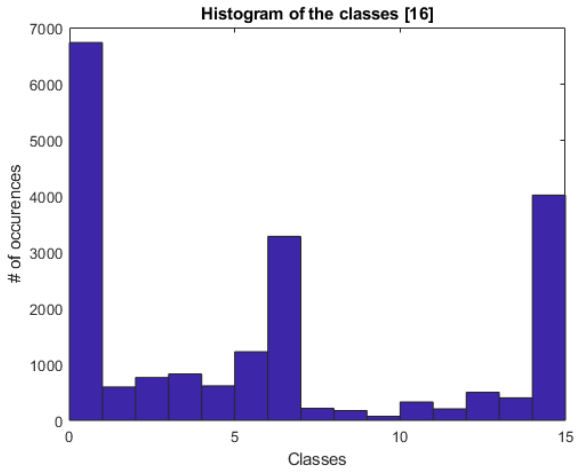

Figure 3.1.3 – Classes histogram

As the graph says, most of the users is always silent, since they belong to the 1st category (0→0000). The rest of them are spread over the other categories, particularly on the 6th one (7→0111) active in the middle of the day and evening and 16th one (15→1111) always active.

3.2.1 Neural Network

The first algorithm used to look for categories was the Neural Network (NN) one. The used NN was a two-layer feed-forward network created using the Neural Net Pattern Recognition MATLAB APP available in the Deep Learning Toolbox, that was composed by four main layers:

- **Input one**, where data were provided (variable dimensions)
- **Two hidden ones**
  - One made by 10 neurons using sigmoid activation function
  - One made by as many outputs as category using softmax
- **Output one**, defined as the hot-vector that represented the identified category (variable dimensions)

The network was trained using a scaled conjugate gradient backpropagation. Input data were divided in random batches before being feed to the NN.

To begin with, 96 samples per users was feed to the NN. The users were labeled manually using categories depending from time of activation and speed of activation.
Business, company and small office users were identified in case of a traffic present in just the working hours (8-20). Their speed was usually nominal for a mail exchange and browsing traffic, characterized by a low average speed and limited burstiness. Domestic users instead had a more relaxed time-window that could span from the non-working hours to all day long. For them the speed could range from low to very-high, probably because they usually watched videos or movies, played videogames, browsed the internet and/or downloaded software upgrade, especially for the smartphones. The size of the training set was 1172 users, of which we destined 70% for learning, 15% for validating and 15% for testing. The Cross-Entropy function represented the achieved minimum of the cost function after the learning was completed (the lower, the better). The performances were evaluated in base of the accuracy in recognition.

	Samples	*Cross-Entropy function value*	*Error Percentage [%]*
Training	820	1.18	42.44
Validation	176	3.08	38.64
Testing	176	3.07	46.59

Table 3.2.1.1 – Neural Network 96 sample accuracy

The average percentage error was very high, as high as giving random response, since it was near the 50%. Definitely, this approach was not good enough.

In accordance with what is described in the Machine Learning chapter, it was took in consideration to extract more data from the database to enlarge the set of input to avoid the possibility of not giving to the Neural Network enough data to learn how to classify

and, in a certain sense, in base of what behavior to classify. So, more user data were extracted, labeled and provided to the Neural Network, exactly for 39422 users.

	Samples	*Cross Entropy function value*	*Error Percentage [%]*
Training	27596	1.38	30.63
Validation	5313	3.78	31.39
Testing	5913	3.79	31.56

Table 3.2.1.2 – Neural Network 96 sample, bigger dataset accuracy

Using more data as input no improvement at all was seen, instead less accuracy was obtained, so data manipulations was considered as mandatory to pursuit the goal of a better identification of the categories.

After seeing that enlarging the training set didn't improve the performance, a different approach was envisioned. Probably the wide range of the input (from bits to Megabits) was too much, not allowing the NN to generalize and so learn. Therefore, reducing it could have eased the learning and increased the accuracy of the neural network. Following this reasoning, binary quantized data were provided to the NN.

	Samples	*Cross Entropy function value*	*Error Percentage [%]*
Training	820	1.16	32.8
Validation	176	3.11	37.5
Testing	176	3.13	36.36

Table 3.2.1.3 – Neural Network binary quantized accuracy

Unfortunately, even in this case the classification didn't provide good results.

Looking for another approach, the ternary quantized data were provided, to better describe and differentiate between silent, slow or fast users.

	Samples	*Cross Entropy function value*	*Error Percentage [%]*
Training	820	1.17	36.7
Validation	176	3.11	42.61
Testing	176	3.11	43.18

Table 3.2.1.4 – Neural Network ternary quantized accuracy

Better results were achieved with respect to the binary quantized one, but it was still not enough and far from a reliable classification algorithm.

After all this experiment, it was clear that the input dataset "as it was" simply didn't describe well the users. So, features extraction was mandatory to obtain some

improvements. The features were extracted using 6 hours averaging window and were the previously described Average, Peak and Peak to Average Ratio.

	Samples	*Cross Entropy function value*	*Error Percentage [%]*
Training	820	1.06	50.49
Validation	176	2.73	44.89
Testing	176	2.73	48.86

Table 3.2.1.5 – Neural Network (AVG-PEAK-PAR) (6h) accuracy

These trials behaved in a similar way to no extraction at all, making it unworthy the effort of extracting the features.

A last attempt was made, substituting the PAR with the ON time, since the former information was thought to be already described by the AVG and PEAK parameter and thus could be considered as redundant. In addition, these features were obtained averaging over 24 hours. So, finally AVG, PEAK and ON were fed to the NN.

	Samples	*Cross Entropy function value*	*Error Percentage [%]*
Training	820	0.892	75.61
Validation	176	2.14	72.16
Testing	176	2.14	71.03

Table 3.2.1.6 – Neural Network (AVG-PEAK-ON) (24h) accuracy

This last set of examples obtained far more better results, leading us to the conclusion that the NN behaves in the best way when less, carefully extracted features, are fed to them. By the way, they didn't seem to be a reliable and accurate way of classifying the users, so other algorithms were tested to compare their performance with the Neural Network approach.

3.2.2 Support Vector Machine

This algorithm still falls in the supervised one and can classify users by projecting the input data on support vectors and trace lines in hyperspaces that acts as a border between categories. The Kernel function decide in which way these lines are extracted and can be of many types. The used SVM is the one available using the Classification Learner MATLAB APP, available in the Statistic and Machine Learning Toolbox. The examined Kernels were:

- **Linear**, that makes a simple linear separation between classes.
- **Quadratic**, using quadratic separation
- **Cubic**, using cubic separation

- **Coarse Gaussian**, making coarse distinction between classes using a Gaussian kernel scaled to $\frac{\sqrt{P}}{4}$ where P is the number of predictors
- **Medium Gaussian**, making less distinction than the fine one between classes using a Gaussian kernel scaled to $\sqrt{P}$ where P is the number of predictors
- **Fine Gaussian**, making finely-detailed distinction between classes using a Gaussian kernel scaled to $\frac{\sqrt{P}}{4}$ where P is the number of predictors

The usual set of inputs were provided labeled using the previously described algorithm that associated each user to a 4-bit label. At first, the 96 sample per user was fed to the SVMs. The observations were 1172, and the predictors 96 per observation.

Type	***Accuracy [%]***
Linear	85.5
Quadratic	84.8
Cubic	83.4
Coarse Gaussian	80.8
Medium Gaussian	80.8
Fine Gaussian	57.3

Table 3.2.2.1 – SVM 96 samples accuracy

It can be noticed that the accuracy is much higher than the Neural Network, and some version, particularly the simpler one gave better result than more refined ones.

No data enlargement was considered in this analysis.

The binary quantized 96 sample per users obtained the following results.

Type	***Accuracy [%]***
Linear	92
Quadratic	91
Cubic	88.6
Coarse Gaussian	52.8
Medium Gaussian	86.3
Fine Gaussian	85.8

Table 3.2.2.2 – SVM binary quantized accuracy

Even in this test, is clear that less complicated model gives in general better results. As a side note, the complexity of the kernel algorithm increases going from up to down in the table. The advantages, by the way, are not following this trend.

Later, the ternary quantized 96 samples per user were used to test the SVMs.

Type	***Accuracy [%]***
Linear	93.1
Quadratic	91.4
Cubic	89.7
Coarse Gaussian	58.6
Medium Gaussian	85.9
Fine Gaussian	86.7

Table 3.2.2.3 – SVM ternary quantized accuracy

These set of tests behaved very similar to the binary, maybe slightly better. As previously said, the description of the two different speed is not worth it, and, moreover, it is better to move from the plain 96 data to features.

In fact, the 6 hours averaged features extracted AVG, PEAK and PAR gave these results:

Type	***Accuracy [%]***
Linear	93.8
Quadratic	94.7
Cubic	93.9
Coarse Gaussian	81.3
Medium Gaussian	93
Fine Gaussian	86.1

Table 3.2.2.4 – SVM (AVG-PEAK-PAR) (6h) accuracy

General improvement using all the kernel functions has been obtained, specifically in the average complicated ones like Coarse and Cubic.

The top accuracy percentage was finally almost in line with the project specification, so the more promising test on the features AVG, PEAK, and Ontime was performed.

Type	***Accuracy [%]***
Linear	94.8
Quadratic	95.1
Cubic	95.9
Coarse Gaussian	91.6
Medium Gaussian	94.4
Fine Gaussian	85.1

Table 3.2.2.5 – SVM (AVG-PEAK-ON) (24h) accuracy

These last tests provided very good results, in fact the top performing accuracy was finally in line with our project specification of 95% demonstrating that this way of manipulating the data was the best one.

3.2.3 Random Forest

The last supervised algorithm used in these analyses was the Random Forest, a set of Decision Trees used in conjunction to reduce overfitting and provide more accurate results.

Different types of branching (rule that describes how and when to split in two more branches) were used, to compare the advantages and disadvantages of each approach.

- **Fine Tree**, with many leaves that allow many fine decisions (max 100 splits)
- **Medium Tree**, with medium flexibility and fewer leaves (max 20 splits)
- **Coarse Tree**, with few leaves that makes coarse decisions (max 4 splits)

The first test performed was the plain 96 sample per day one.

Type	***Accuracy [%]***
Coarse	59.7
Medium	63
Fine	63.9

Table 3.2.3.1 – Random Forest 96 samples accuracy

From these results, it was seen that the random forest didn't seem very good at distinguishing the categories. But still it gave better result than the Neural Network one.

Quantizing using two levels gave these results:

Type	***Accuracy [%]***
Coarse	76.1
Medium	73.7
Fine	65

Table 3.2.3.2 – Random Forest binary quantized accuracy

Better results were obtained using these manipulated inputs, but they were still behind the target of this analysis.

Quantizing over three levels, instead of two gave the following results:

Type	***Accuracy [%]***
Coarse	75.1
Medium	73.7
Fine	66.5

Table 3.2.3.3 – Random Forest ternary quantized accuracy

The results were almost identical to the binary quantized, so the extra effort was not justified, as observed in all the previous tests.

The features extraction was then used, over the usual AVG, PEAK and PAR obtained using a 6 hours window.

Type	***Accuracy [%]***
Coarse	84.3
Medium	79
Fine	71.5

Table 3.2.3.4 – Random Forest (AVG-PEAK-PAR) (6h) accuracy

In this case better results with respect to previous trials were obtained and, as seen in previously, coarse provided the best result.

Finally, the AVG, PEAK and ON time obtained using a 24 hours window features set was fed to the Random Forest.

Type	***Accuracy [%]***
Coarse	93.1
Medium	89
Fine	58.8

Table 3.2.3.5 – Random Forest (AVG-PEAK-ON) (24h) accuracy

In this last test, consistent and valid results were obtained for the coarse kernel, making it evident that these last set of features is the one easiest recognized by the Random Forest approach. Unfortunately, the fine-tuned tree didn't follow the expected behavior, obtaining a very low accuracy with respect to not only other method, but also to previous input types. Probably, the excessive overhead and distinction didn't allow to generalization and so, bad results were achieved.

3.2.4 K-Means clustering

The last and only unsupervised algorithm tested was the K-means clustering one, which didn't need any prior labelling and was able to automatically find "cluster". The users were assigned to different categories based on an evaluation of centroids and minimum distance over the many dimensions provided as input.

Since here we didn't have an assumed true label to confront with the output of the algorithm, no accuracy could be evaluated and, subsequently, used as a performance parameter in our analysis. Therefore, just the average behavior of the user that were recognized to belong to a specific class were plotted, to give the programmer a glance on how a specific class behaved and so, in which programmer-defined category to put it (the previously described business and office user, high speed user and so on).

The set of manipulated inputs were the same as the previous supervised learning trials. The number of categories to be looked for was set to 10, considered appropriated to balance user division and readability of the outputs. The number of iterations of the algorithm was set to 10.

At first the 96 sample per user was fed to the classifier. The resulting average behaviors of each class were evaluated and plotted.

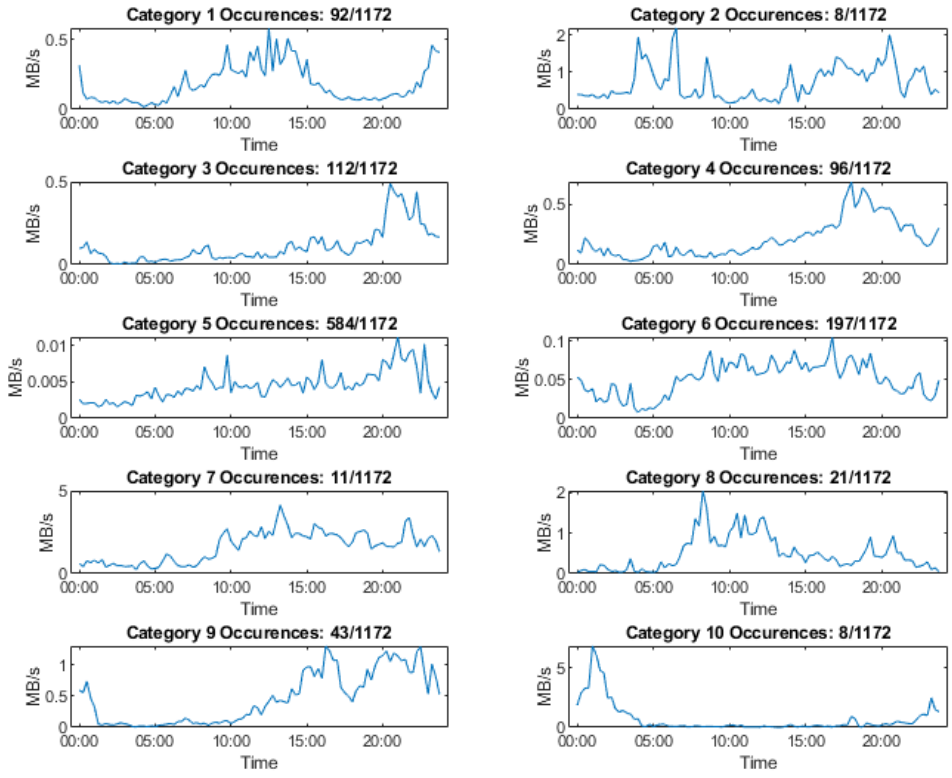

Figure 3.2.4.1 – K-means clustering 96 samples average classes behavior

The identified categories seem to represent different types of users. As an example:

- 10th - seems to represent users that use the internet in a very fast way during the late evening and night (could be movie watcher as well as software upgrade)
- 8th – represents an average user that use the internet through the day with medium speed. During the evening the usage is limited so it could also be a business user.
- 5th – very slow, always active users fell in this category, in fact, their average speed was around 5 kbps, considered as normal control traffic. Notice also that this is the most crowded category.

By the way, all these categories highlight the fact that the most crowded hours are around midday and evening and the less crowded hours are around 5 AM. Finally, it should be noted that the high dynamics of the traffic speed, ranging from Kbps to Mbps can also make this prediction not so accurate due to the presence of an average in the evaluation of the plots.

Using the binary quantized input, since the daily traffic shape was constituted by a collection of 1 and 0, it was measured the percentage of active users in a category to better represent a class behavior.

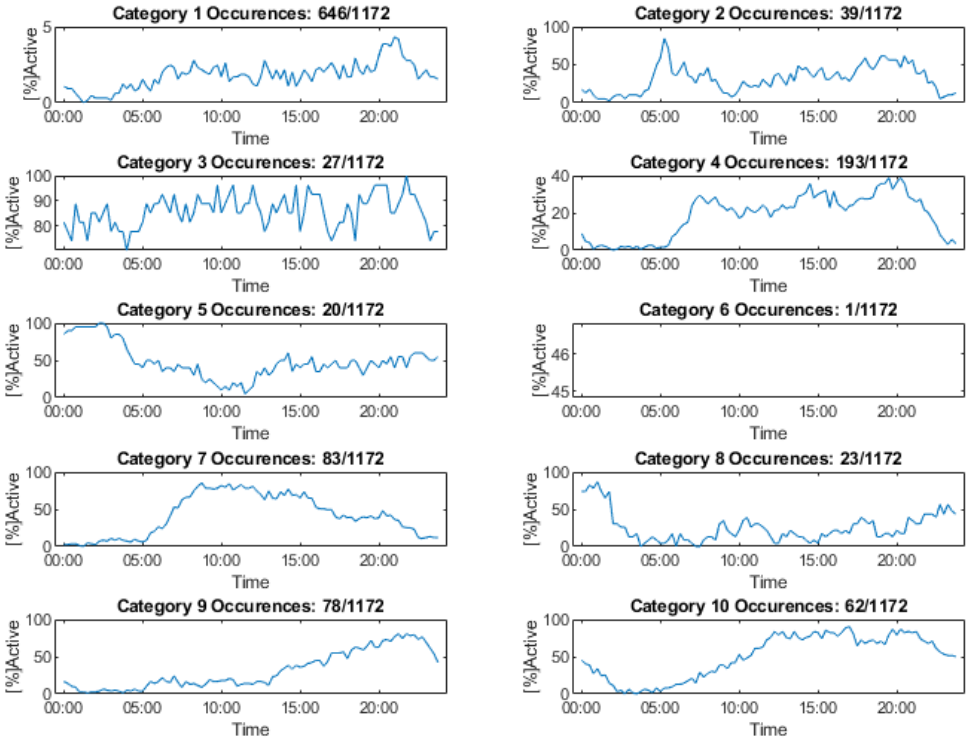

Figure 3.2.4.2 – K-means clustering binary quantized average classes behavior

Using the binary quantized features, it can be better understood how much users belonging to an identified category were active at a specific hour of the day. As an example, the 7th category can be categorized as business and office users one, to the 9th one, users that connect to the internet particularly and more intensively in the evenings, to the 3rd users that are more or less always connected during the day in a sort of cyclic way. The 6th category appears to be blank just because it was averaged over just one user (even if the experiments were repeated many times, this separation was still present).

Using the ternary quantized input, the percentage of active slow and active fast user were plotted. It was decided to differentiate between the two categories to make it more evident when slow users were active with respect to fast users. It should be noticed, by the way, that at every sample each user is quantized according to its instantaneous speed, so a user could be at first silent (0), but just after fast (2) and immediately later slow (1) in the space of three samples.

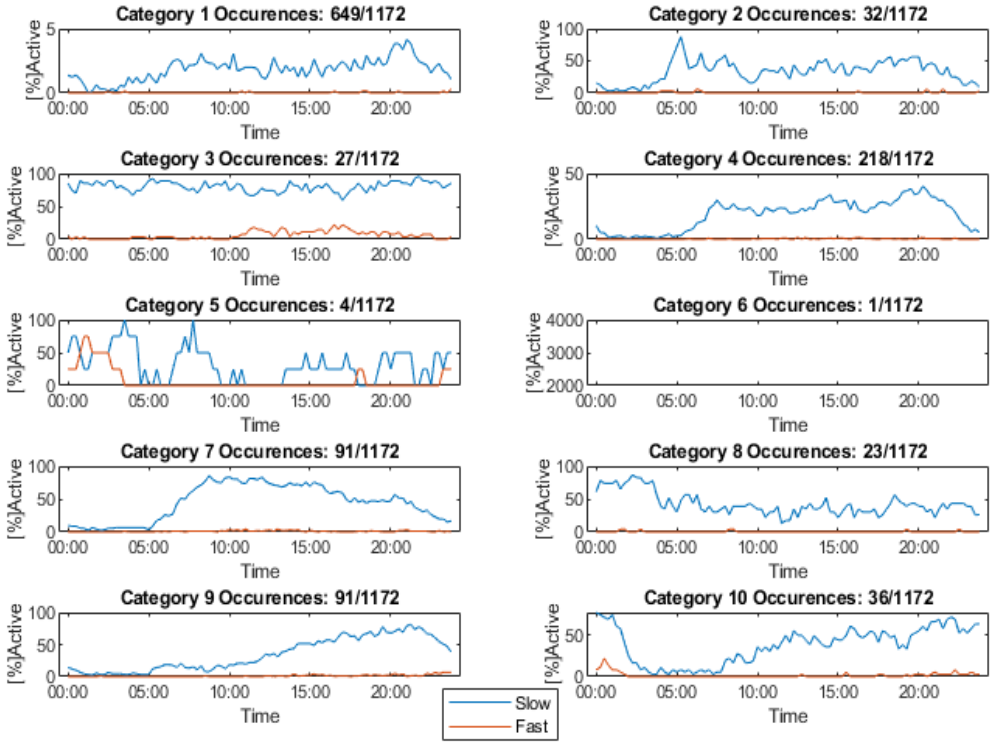

Figure 3.2.4.3 – K-means clustering ternary quantized average classes behavior

By these graphs most of the users can be identified as slowly active (the fast threshold was > 3 Mbps). It should be noticed, by the way, that these speeds are evaluated using a cumulated traffic counter that is updated once every 15 mins, so very fast burst are "mitigated" by this averaging process. Nevertheless, high speed users were spotted in the 5th category, where some users were active during the night or in the late evening. From the 1st category, it can be spot that the widest slice of the users is either slowly active during all day, or totally silent.

Finally, just the features set composed by AVG, PEAK and ON time were considered and gave to the classifier. The resulting plot didn't describe the average of each feature, since that would have not given an idea on how users belonging to a specific category behave. Instead, the whole set of user's features was plotted sorting it by category to show how different users behaved in a similar way when put in a specific category. Clearly, for a given abscissa over each of the graph, the same user was represented.

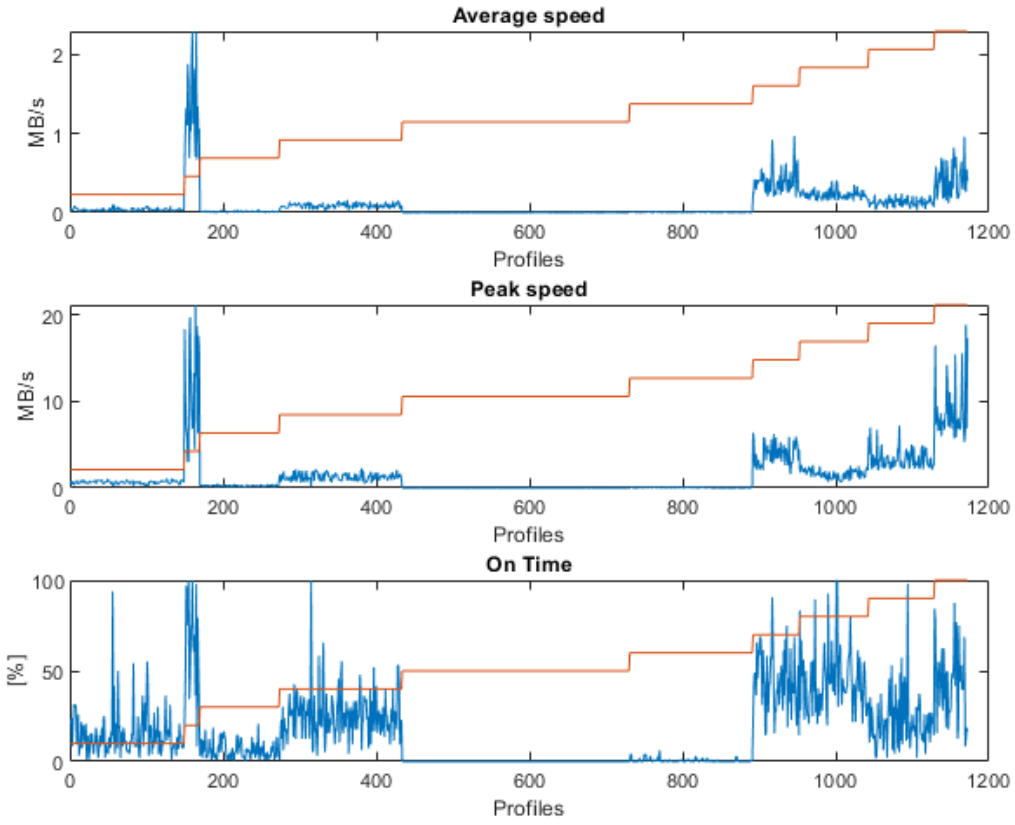

Figure 3.2.4.4 – K-means clustering (AVG, PEAK, ON) (24h) classes behavior

By these last graphs it can be noticed how a very big part of the user is actually just silent. In fact, the 5th categories (that lies almost in the center of the graphs) is characterized by AVG, PEAK and ON time equal to 0 generating what we can call the "silent users" category. On the opposite side, we can spot the fastest and with higher burstiness users, identified in the 2nd category, with a pretty high AVG of 1.5 Mbps (always remember that the average is computed over a 24 hours period composed by 96 samples), a PEAK speed of about 10 Mbps and ON time at around 60 to 80 %. All the other categories described all the possible variations from these two extremes, like the really slow and rarely active 6th one and the high burstiness but low average 10th one.

The previously spotted trend that seems to favorite the features with respect to the other way of representing the data can now be also seen and understand easily by a human eye, that finds much more comfortable reading and understanding these results with respect to the previous way of viewing the data's results.

To give a final overview, all the different supervised learning algorithms performances are summed up in the tables below. These are useful to better identify the most accurate with respect to different input data manipulations.

Input Data	Accuracy [%]
96 Samples	46.6
Binary Quantized	36.6
Ternary Quantized	43.2
6h Features	48.9
24h Features	71

Table 3.2.1 - Neural Network accuracy

	L.	Q.	C.	C.G.	M.G.	F.G.
96 Samples	85.5	84.8	83.4	80.8	80.8	80.8
Binary Quantized	92	91	88.6	52.8	86.3	85.8
Ternary Quantized	93.1	91.4	89.7	58.6	85.9	86.7
6h Features	93.8	94.7	93.9	81.3	93	86.1
24h Features	94.8	95.1	95.9	91.6	94.4	85.1

Table 3.2.2 - Support Vector Machine accuracy

	Coarse [%]	Medium [%]	Fine [%]
96 Samples	59.7	63	63.9
Binary Quantized	76.1	73.7	65
Ternary Quantized	75.1	73.7	66.5
6h Features	84.3	79	71.5
24h Features	93.1	89	58.8

Table 3.2.3 - Random Forest accuracy

Chapter 4
The protocol information

In this chapter we will further try to differentiate the users using another set of features extracted from a traffic identifier, which is able to distinguish and identify the protocol with which the traffic was performed. At first, we will give a brief description of how many and which protocols are present in this analysis, and then we will try to highlight eventual grouping using "aggregation" of users that belong to a specific population or beam. Finally, we will again try to extract some features out of this set of data to further proceed in the analysis.

4.1 Deep Packet Inspection approach

The device used to perform this action is called Deep Packet Inspector. It employed different methods and algorithms to examine the traffic and provide a classification of it. The DPI approach is different than packet filtering, since the latter simply looks at the packet header, the IP destination and source address, port number and packet size, while the former looks also inside the packet, to identify specific application and traffic type.

The categories that the DPI advertise to recognize are listed in the table below.

#	Category	#	Category	#	Category
1	Google	17	Whatsapp	33	Email
2	Http	18	Hangouts	34	PlayStation
3	Https	19	Snapchat	35	Other Gaming
4	eBay	20	Telegram	36	Bit Torrent
5	Other Browsing	21	Other IM	37	E Donkey
6	Twitter	22	Dropbox	38	P2P Live
7	Facebook	23	Google Drive	39	Other P2P
8	Vine	24	Other Cloud	40	Skype
9	Instagram	25	ITunes	41	Whatsapp Call
10	Other Social	26	Apple Update	42	Viber
11	YouTube	27	Google Play	43	SIP
12	Netflix	28	Amazon App Store	44	Other VoIP
13	Sky	29	Windows Update	45	FTP
14	Sky Go	30	Other SW Update	46	Other File Access
15	Facebook Video	31	VPN	47	Network Admin
16	Other Streaming	32	Other VPN	48	Other

Table 4.1.1 – Protocol list

To reduce the cardinality of our user's matrices, an aggregation was done according to the look-up table and colors.

#	Category	#	Category	#	Category
1	Browsing	6	App Store	11	P2P
2	Social	7	Update	12	VoIP
3	Streaming	8	VPN	13	FTP
4	IM	9	Email	14	OAM
5	Cloud	10	Gaming	15	Other

Table 4.1.2 – Aggregated Protocol List

4.2 Data format

In this analysis the users were characterized by a set of information similar to the previous chapter one. Each user was described by:

- **MAC**, a unique 12 hexadecimal string
- **Population ID**, referring to the Population to which it belongs
- **Beam ID**, referring to the Beam under which the user was served
- **Product ID**, referring to the Product activated on that user account
- **Forward Channel Traffic Matrix**, composed by the traffic performed from the satellite to the user over a day divided per each protocol
- **Return Channel Traffic Matrix**, composed by the traffic performed from the user to the satellite over a day divided per each protocol

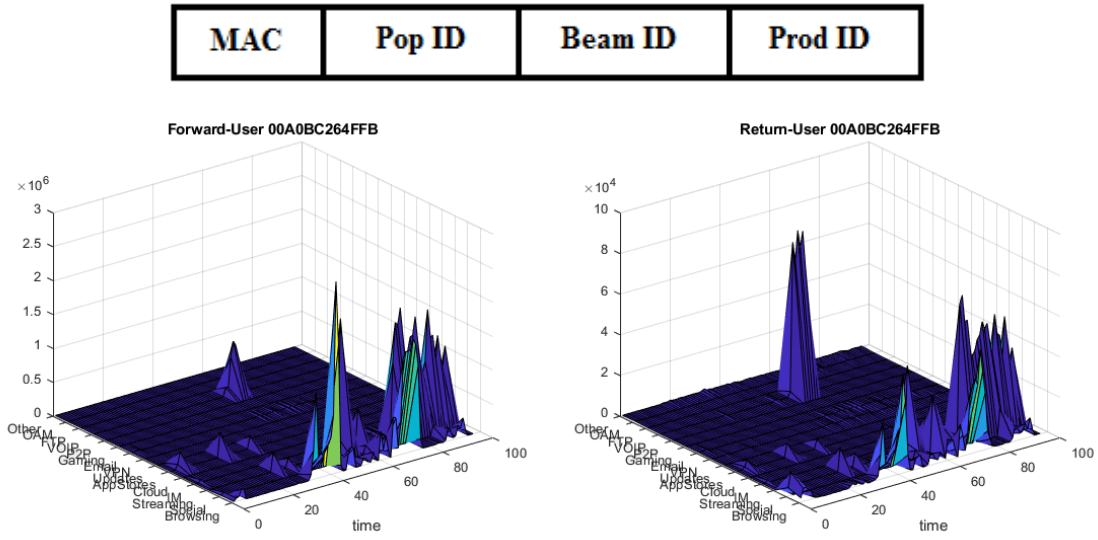

Figure 4.2.1 – User Data Format

The traffic matrix was also plotted in a different way, to ease visualization and to better represent the protocol with which the traffic was performed.

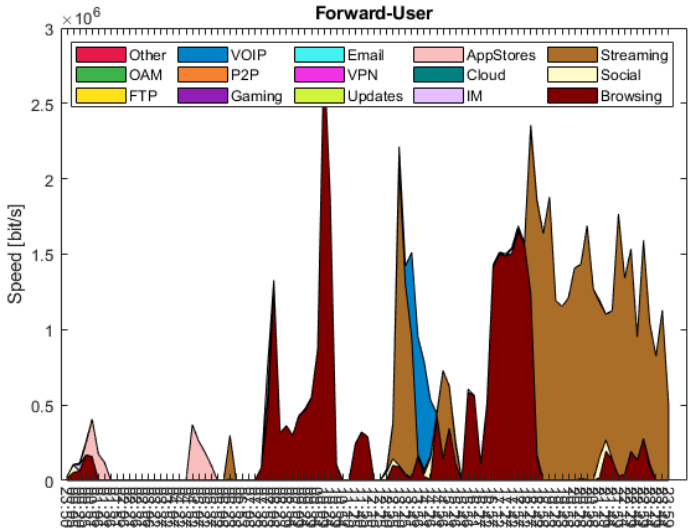

Figure 4.2.2 – Stacked daily traffic profile

In the later analysis we will refer to this type of plot as stack plot, since it stacks the traffic performed by each protocol one above each other, finally characterizing the speed that the user was reaching in each moment of that day.

According to the previous chapter analysis, a set of features was evaluated to describe a user. They were the same as the last analysis one, namely AVG, PEAK and On Time evaluated over the 24 hours window. Although the type of features where the same, their numerosity was much more, since they were evaluated for each of the summarized 15 protocols, generating for each user 45 features (3 features for each of the 15 protocols).

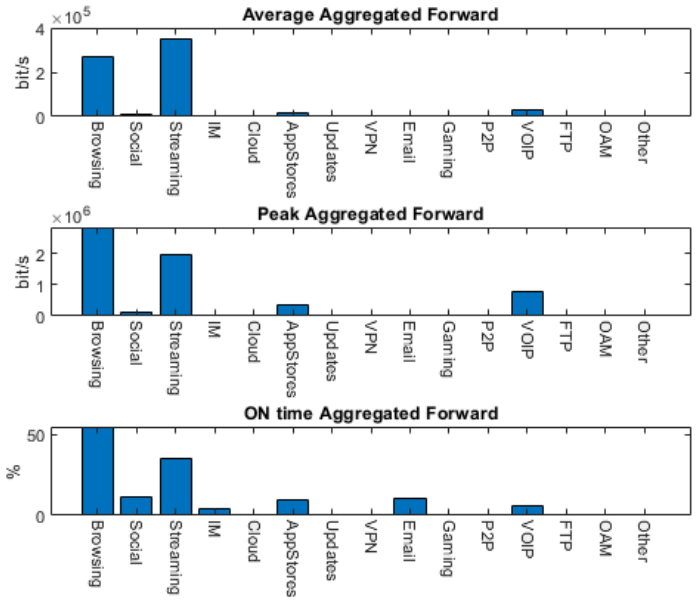

Figure 4.2.3 – Daily evaluated protocol features

Finally, to compare different user's behavior, a plot showing the percentages of traffic performed for each protocol for each day was made. In this way it was better highlighted not how much traffic was made but its type.

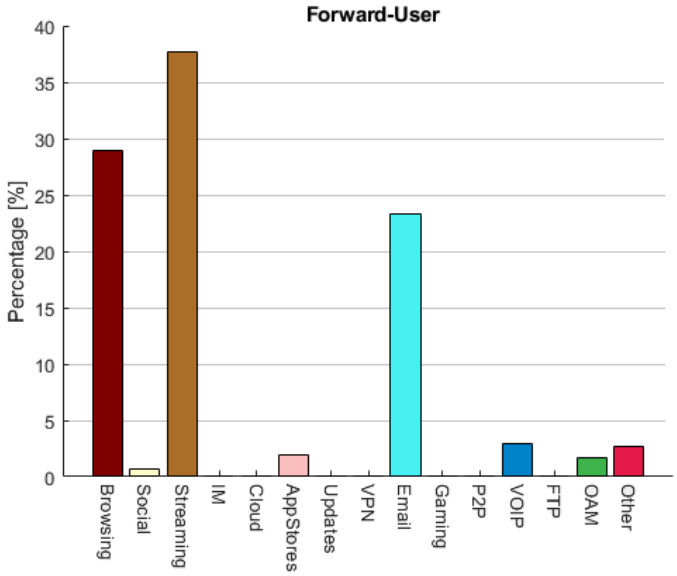

Figure 4.2.4 – Percentage of type of traffic performed

4.3 The classification

The algorithm used to divide the big number of users that were at disposal was the previously cited K-means clustering. The same idea as the one used in [10].

Since in this analysis the number of categories was very important, a brief study on the numerosity of this category was made. The automated algorithm had a built-in functionality that automatically looked for the best number of categories out of a given input. Using it, we obtained that the right number of categories was exactly 10.

To motivate and verify this choice, the algorithm way of choosing the right number of categories was investigated. But first, it is needed to introduce some statistical metrics used to evaluate the performance of a clustering algorithm.

The **Within Cluster Sum of Square Error (WCSS)** formal definition is:

The sum of the squared deviations from each observation and the cluster centroid

$$WCSS_c = \sum_{i=1}^{N} (x_{c,i} - C_c)^2$$

Where

- $x_{c,i}$ are the coordinates of the observation *i* than belongs to class *c*
- C_c are the coordinates of the centroid of class *c*
- N is the cardinality of class *c*

This quantity measures the variability of the observations within each cluster. As a rule of thumb, the smaller the quantity, the more compact the cluster. Cluster with high values show more variability of the observation within the cluster.

Furthermore, other two metrics, dependent on the TWCSS were evaluated, to even better represent the metrics changing and improvements. These two metrics were:

The **Sum of Square Error to Grand Mean (SS)**, that determines

The dispersion of the observations with respect to the grand mean

$$SS_c = \sum_{i=1}^{N} (x_{c,i} - M)^2$$

Where

- $x_{c,i}$ are the coordinates of the observation *i* than belongs to class *c*
- M is the mean of all the centroids
- N is the cardinality of class *c*

This is similar to the previous WCSS, but the deviation is evaluated with respect to the Grand Mean, that is to say, the mean of all the centroids.

The **Between Cluster Sum of Square Error (BCSS)**, that describes

The squared average distance between all the centroids

$$BCSS_c = \frac{1}{N} \sum_{i=1}^{N} (C_c - C_i)^2$$

Where

- C_c are the coordinates of the centroid of class *c*
- C_i are the coordinates of the centroid of class *i*
- N is the total number of classes

This metric better describes the way in which the centroids were place, in order to better understand what the algorithm was printing as an output and to better judge its performance.

After this brief introduction it can be explained how the automated algorithm decided to use exactly 10 categories for our analysis. The selection of the best number of classes is based on an iterative procedure that tries to increase the number of categories from 1 to a maximum amount specified by the user. Each time that it increases the number of categories by one, it evaluates the previously described metrics, pick the WCSS of the

actual step and confront it with the previous step WCSS. If this variation is lower than a certain threshold, it stops it and give the previous step as good, since the improvement were not worth the additional category.

The variation between the actual step WCSS and previous step WCSS is called Post Reduction Error (PRE) and is calculated using the formula:

$$PRE_i = \frac{WCSS_{i-1} - WCSS_i}{WCSS_{i-1}}$$

The threshold is evaluated using the minimum between two values:

$$0.8$$

or

$$0.02 + \frac{10}{Number\ of\ training\ rows} + \frac{2.5}{Number\ of\ model\ features^2}$$

In our specific case the threshold was exactly:

$$th = 0.02 + \frac{10}{7655} + \frac{2.5}{45^2} = 0.025$$

The two PRE between 10 and 11 categories were:

$PRE_{10} = 0.038$ → above threshold

$PRE_{11} = 0.0225$ →below threshold

That's why the algorithm stopped exactly there.

But to better understand the overall behavior of this metric, and to assure our self to not be stuck in a local minimum, it was decided to investigate the metric over a span of number of categories.

In the following study, the sum of all these metrics were used as a metric. This was referred to as the "Total" of each of these quantities. The number of categories varied from 1 to 50 and each of the metrics were plotted to understand the general behavior.

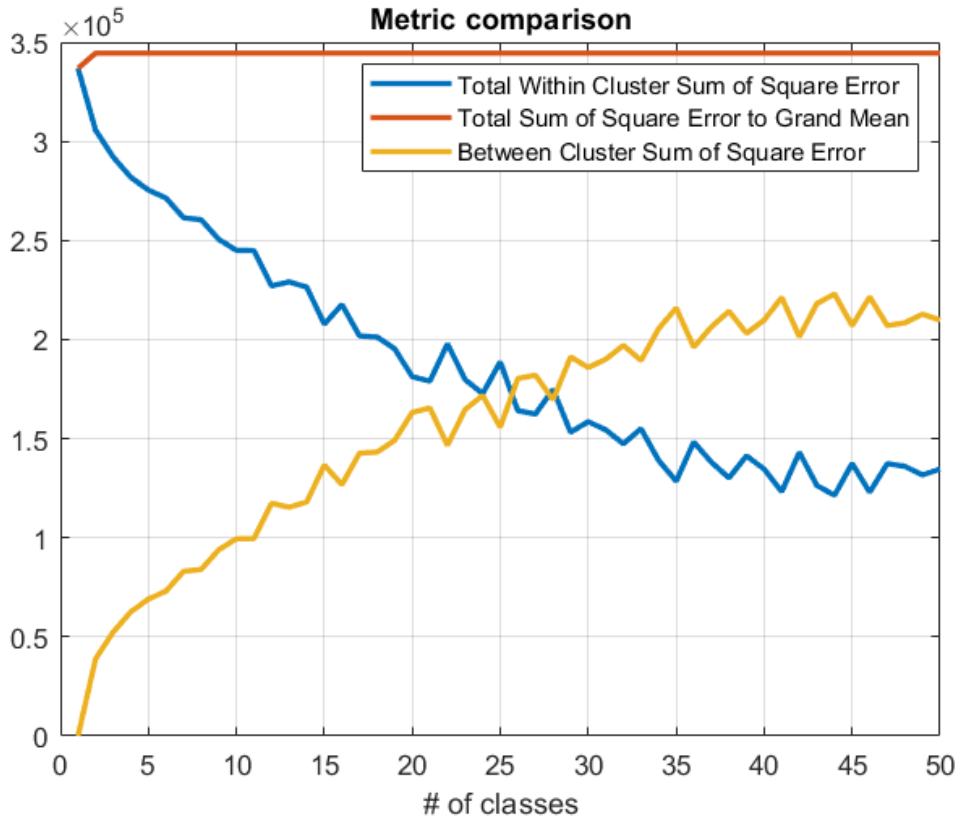

Figure 4.3.1 – K-Means clustering metrics comparison varying # of classes

As we can see, the three metrics evolved in dependent way.
In fact, the TSS was the sum of TWCSS and BCSS. By the way, it can be noticed that this quantity doesn't provide us so much information, since it was stable along all our evaluation. Probably the algorithm used this metric as parameter and kept it fixed for every trial (apart from the case where we had just 1 category).
Regarding the other two metrics, they behaved in an inversely proportional way (since their sum was bounded by the TSS). Focusing on the TWCSS, since it was the one used to stop the optimization algorithm it can be easily seen that steps between 10 and 11 is the first one that exhibit a "flat" behavior. But the suspects were indeed true, in fact, the quantity keeps going down until it reaches a "less steep" condition after the number of categories increases. This represent the fact that, after a certain threshold, there is no advantage in increasing the number of categories used.

To even better understand this kind of "saturation" of improvements, the derivative of the TWCSS was evaluated (and averaged) to identify the point in which adding more categories was no more beneficial.

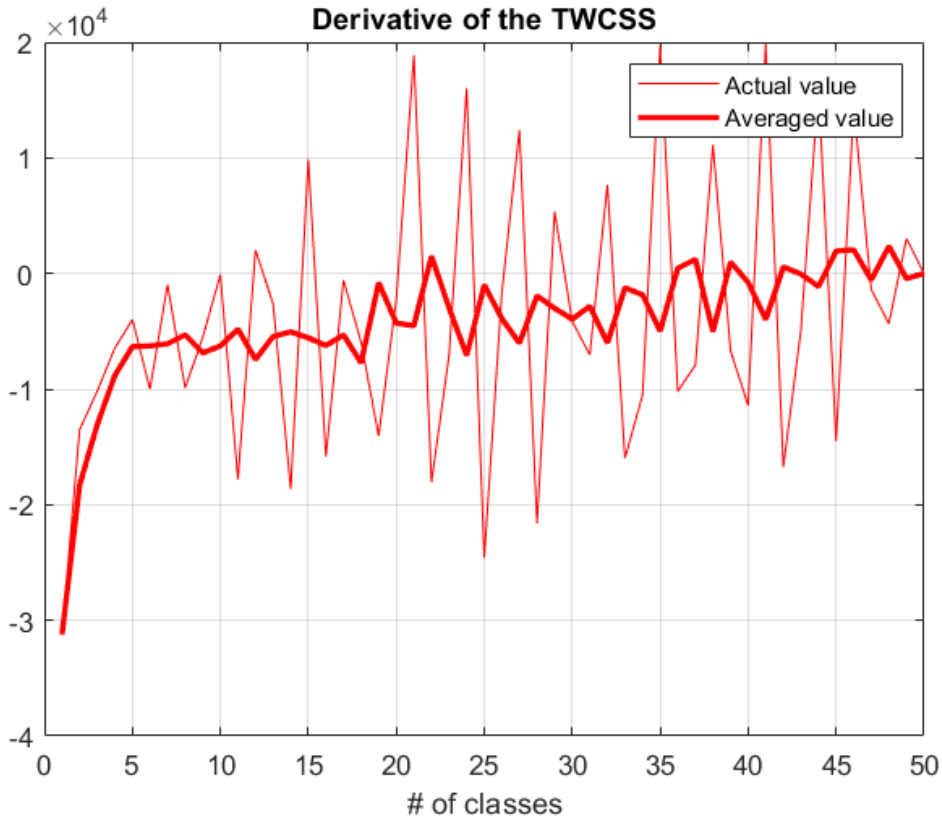

Figure 4.3.2 – Derivative of the metric TWCSS varying the # of classes

It can be seen that the algorithm stopped exactly where the derivative reached almost 0, that is to say, when the number of categories was 10. After that point, we can see an oscillating behavior, symptom that it is not worthy to go beyond that number of categories, also because sometimes the derivative became again positive, sign that we are stepping back from our optimum. Finally, the very low improvement here is easily detectable, particularly, the Elbow method can be easily be employed.

The Elbow method consists in a heuristic way of evaluating the best number of categories in a clustering analysis. Its main idea is that there is no reason in going for more categories if the metric that describe our improvements didn't get much better. The elbow point is exactly that point were the derivative of our function almost reaches 0.

4.4 User Features Results

After having found what was the right number of categories, the 45 users features were extracted and fed into the k-means clustering algorithm. The following result show the average behavior of each of the founded classes. The speeds are displayed in Kbps, tenth of Kbps, hundredth of Kbps, Mbps and tenth of Mbps, to ease visualization.

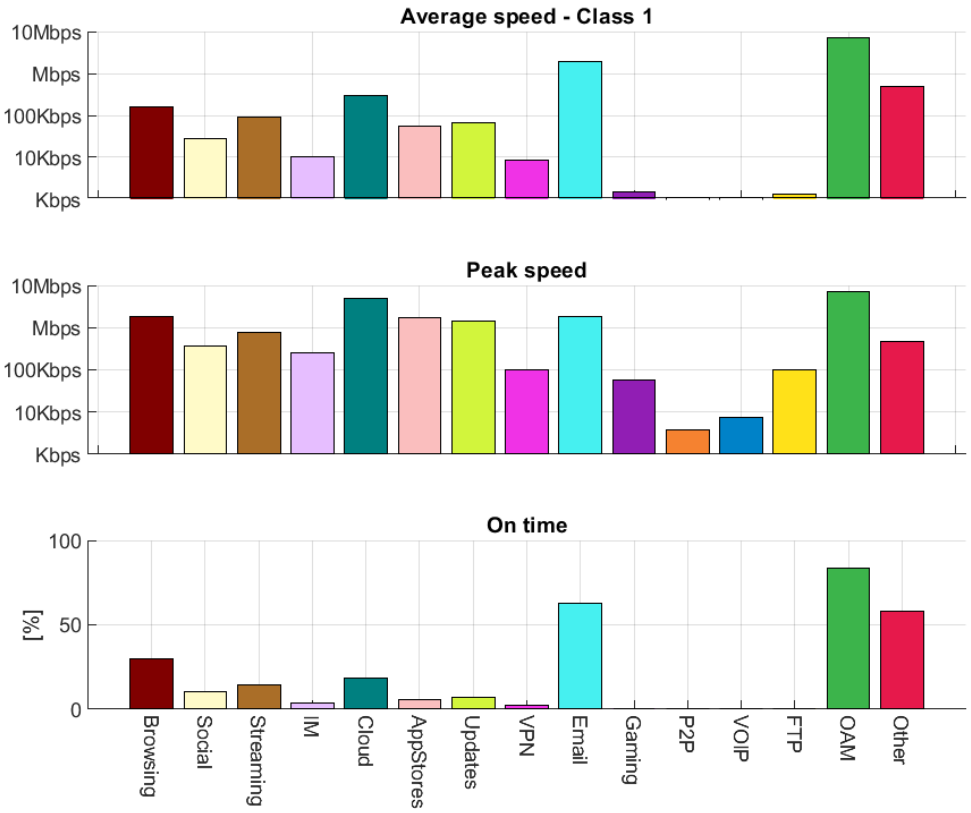

Figure 4.4.1 - User Features Class 1

The first identified category presents a wide variety of employed protocol, out of which stand out the OAM and Emails. The OAM was on for more than 80% of the time, with the avg speed very similar to the peak speed, sign that it was a continuous, high speed transmission. The Email were also active for a lot of time at a very high speed, even if usually they produce very little speeds. Some minor Browsing, Streaming and Cloud was performed, with the last one having far the highest peak speed between the three. All the other protocols were still employed, as we can see entries in the P2P, VOIP and FTP, but the percentage of active time was very low with respect to the other ones.
The number of user present in this category was still very limited, just 24 out of 7655, sign that this is a very peculiar class that not so many users belonged to.

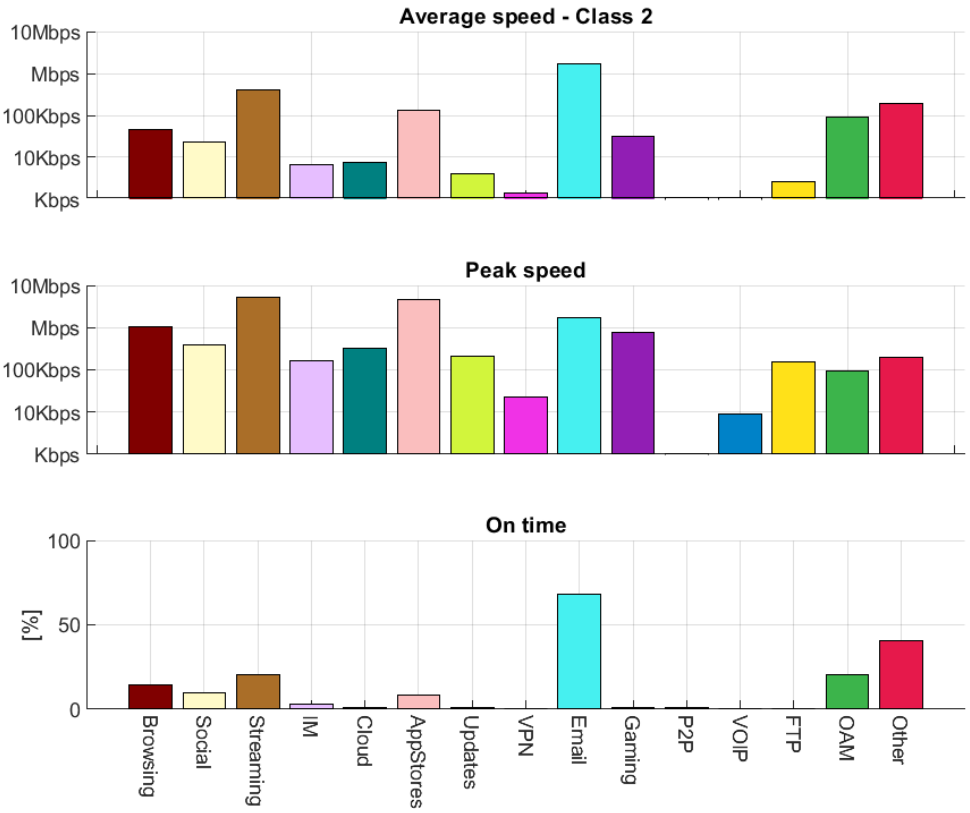

Figure 4.4.2 - User Features Class 2

In this class we can find users that used mostly the Email protocol with a considerable OnTime, and a maintained average speed, confirmed by the small difference with respect to the peak speed. The second highest peak speed are Streaming one, followed by the Appstore, that reached very high speed for a short amount of time, sign that the user watched some video out of our identified protocol and updated the phone apps, (things that employs a very high instantaneous speed and resolves in, usually, a small amount of time).

The Browsing and Social experience were used for a small fraction of time, with average speed and peak speed that were nominal considering this applications. Of particular interest is the presence of Gaming protocol, not spotted in our previous class, that employed decent peak speed, probably for a console software update or to download a game from the online store. The remaining protocol performed some minor traffic that didn't characterize much more this class.

The numerosity of this category is not so huge, but still significant, since we found 188 out of 7655 users whose behavior followed the average features of this class.

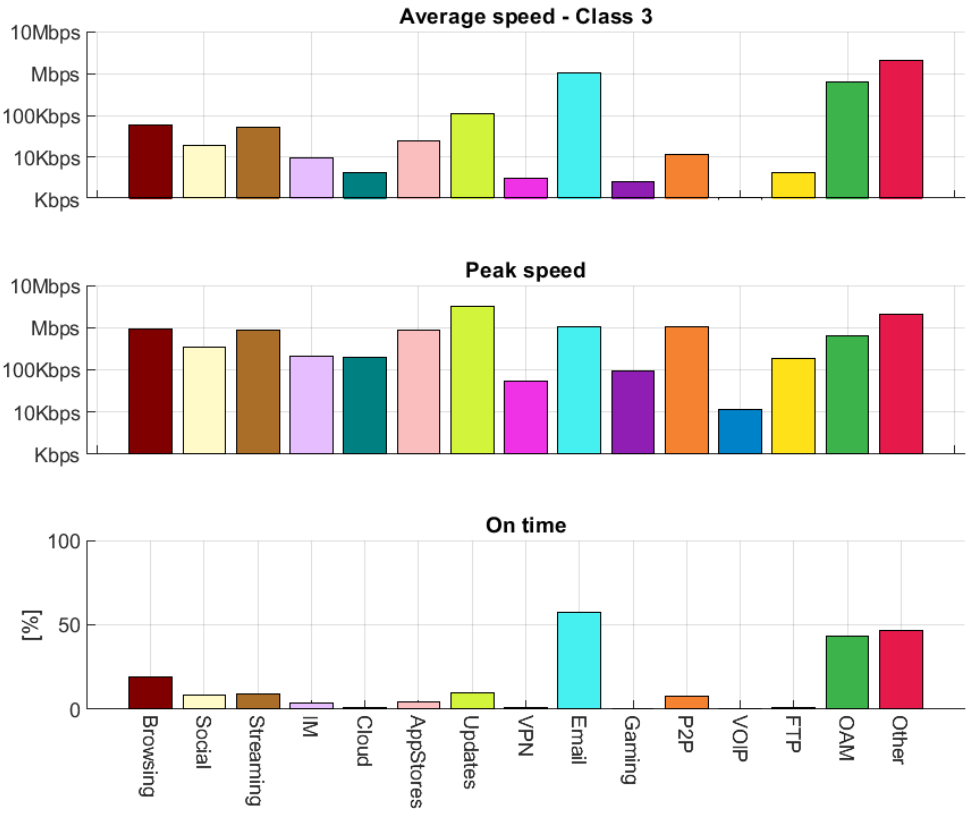

Figure 4.4.3 - User Features Class 3

The third identified category was composed by people that used not only the Email, similarly all the other categories, but also Update and P2P protocol, differently from the other classes. In particular, both the Update and P2P reached pretty high peak speed, similar to the Streaming one. In fact, the Streaming protocol usually behave in a very bursty way, that is to say, its speed changes with an intermittent behavior where very high-speed spike are followed by very low speed transmission, that are followed by high-speed spike again. In this class, the Other protocol obtained pretty high average speed and peak speed, actually very similar one to another, but with an average OnTime.
The cardinality of this class was not too much, in fact 92 out of 7655 were found conforming to this category.

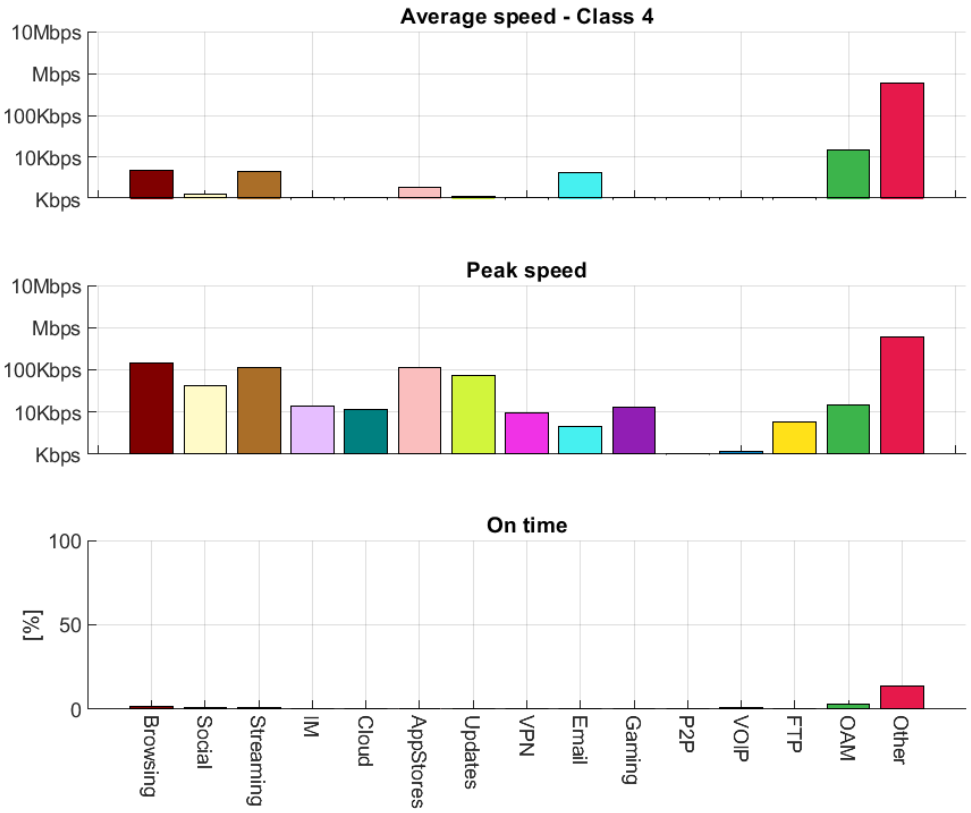

Figure 4.4.4 - User Features Class 4

This one is the most peculiar category that was found in our analysis. The number of people conformant with these average values was huge and more than 63% of the total (4855 out of 7655 precisely). The strange behavior was a very high average and peak speed of transmission performed just by protocol Other and almost no other traffic with other protocol, as we can see from the OnTime graph. Some other of the remaining protocol spikes, actually very limited in speed, were detected, but their activity time was surely less than 2% at day (< 30 minutes at day).

To investigate more deeply, some investigations were made with the network engineers, confronting these data with the one collected by the accounting. It was discovered that the accounting didn't registered any session of these Other and FTP protocol, but just the small spikes performed by the remaining protocols. In the end, it was discovered that the probe that collected the traffic to be delivered to the DPI was placed not at the user premises, but in a "previous" part of the network, and so, collected all the telemetry traffic that was performed daily by the platform to ensure that the service was functioning in a good way. So, finally these kinds of users were just the silent one, mistakenly identified as super active one due to the telemetry traffic appointed to them by the DPI.
By the way, this strange effect was not present in the accounting and didn't impact at all the user (since they for sure would have noticed if their data caps were slowly decreasing even if they didn't perform any traffic).

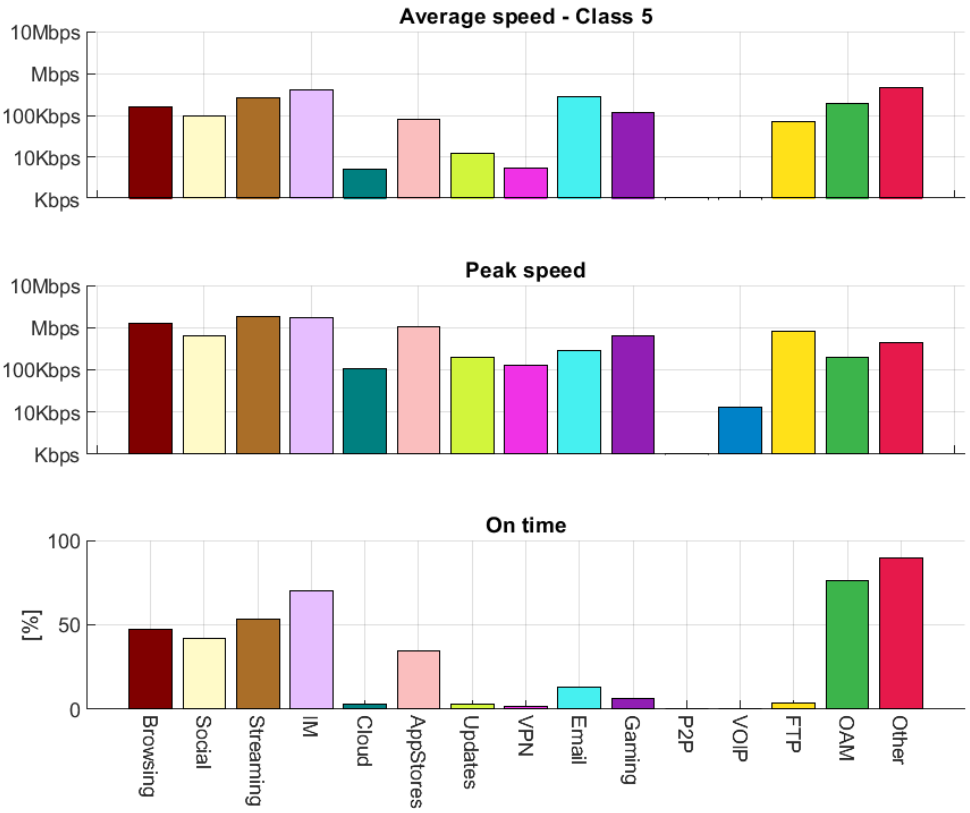

Figure 4.4.5 - User Features Class 5

The fifth class was characterized by users that performed Instant Messaging for the majority of the time (the Other protocol high OnTime can be related to the discussion made in the description of class 4) and with a pretty high average speed. A possible reason is that there exists some special type of user like newspaper of certain country that exploit the capillarity and reachability of the instant messaging app like WhatsApp, Telegram and Facebook Messenger to inform and share the latest news and hotel, mountain refuges, touristic villages that rent their capacity to their guest, that usually use them to communicate, share their holiday picture on social media and browse the internet to find some good place to go nearby their resting place. In fact, this last reasoning seems to be much more credible if we apply it on the graphs, since we have exactly the previously described behavior, with some minor spike of every protocol, P2P excluded. Even the Email protocol, used by most of the user in the previous class was used very rarely by the users (maybe just by the receptionist or business men). The number of users falling in this category was limited, just 68 out of 7655, validating more our hotel like users' hypothesis.

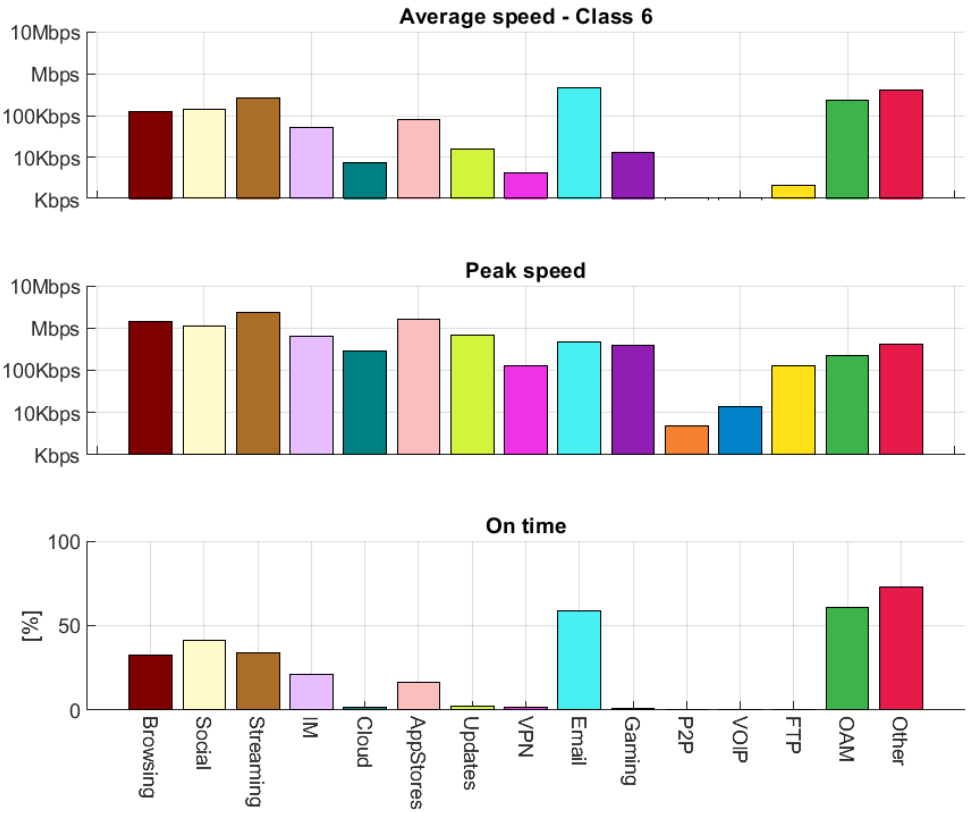

Figure 4.4.6 - User Features Class 6

This user class contained people performing mostly Streaming, Browsing, visiting Social Media and updated their app using the Appstore. Their behavior was not so unique, since they performed on average all protocols with nominal average speed and peak speed. The OnTime of each protocol was on average, a part of the usual Email, OAM and Other that always spikes up above the remaining ones. Their numerosity was relevant since they were 174 over 7655. We could identify in this category the most heterogenous users that performed every type of traffic during the day.

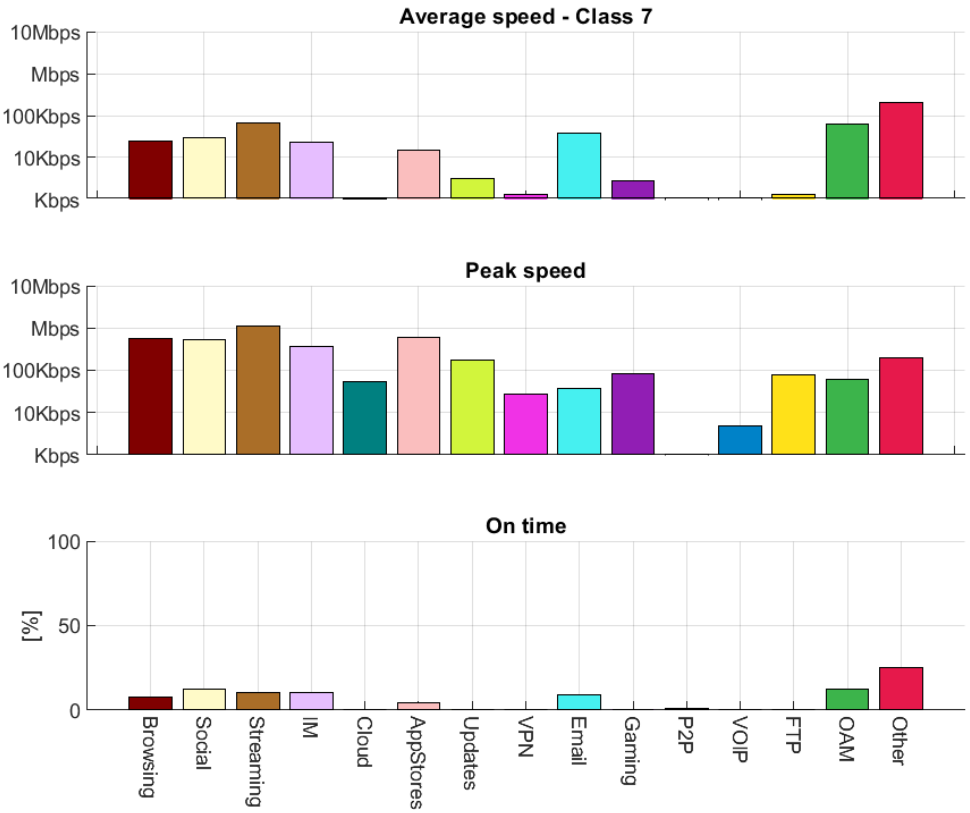

Figure 4.4.7 – User Features Class 7

The seventh class contained all the user that used their connection less time during the day. In fact, their OnTime are significantly low (<13 % max) and, differently from all the previous classes, also the Email, OAM and Other protocol ONTimes were pretty low. Probably, these users kept their apparatus turned off and just turned them on when they needed. By the way, their performed traffic when they were on was pretty nominal, where it can be noticed the usual peak speed supremacy of the Streaming, followed by the AppUpdates, the Browsing and Social Media consulting. Moreover, very small Cloud synchronization were present, very fast and with low peak speed. Furthermore, some very light network using Gaming was performed and some VPN session. The most noteworthy note that can be made on this class is on its numerosity, in fact 763 users out of 7655 were identified in this category, making this the second most populous class in our analysis.

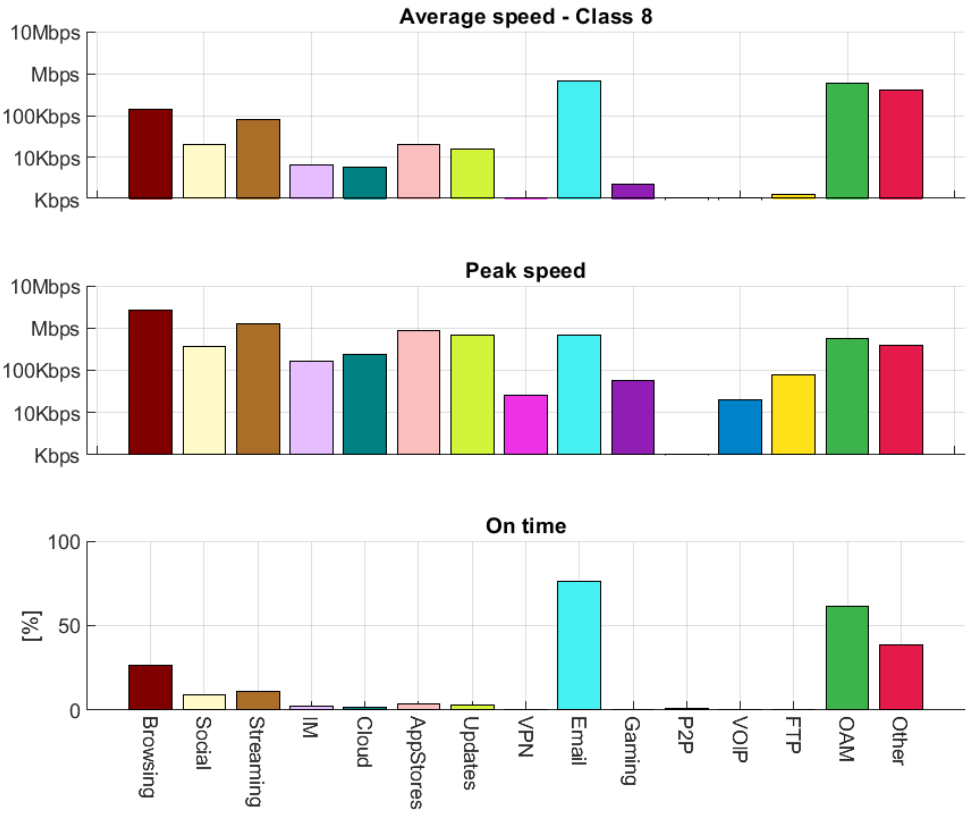

Figure 4.4.8 - User Features Class 8

The highest peak speed of this class belongs to the Browsing category, where users performed some very network intensive browsing experience from the network speed point of view. In fact, we can notice peak of around 6 Mbps for this protocol, that are pretty high if we consider that usually browser just download the HTML file and their attachment. The other mostly used protocols are the always present Email, OAM and Others. No relevant P2P connections were registered (since their OnTime was different from zero and the speeds were below 1 Kbps), some very short VPN sessions and a moderate use of every other protocols. The cardinality of this class is pretty big, since 405 of 7655 were identified as belonging to this category.

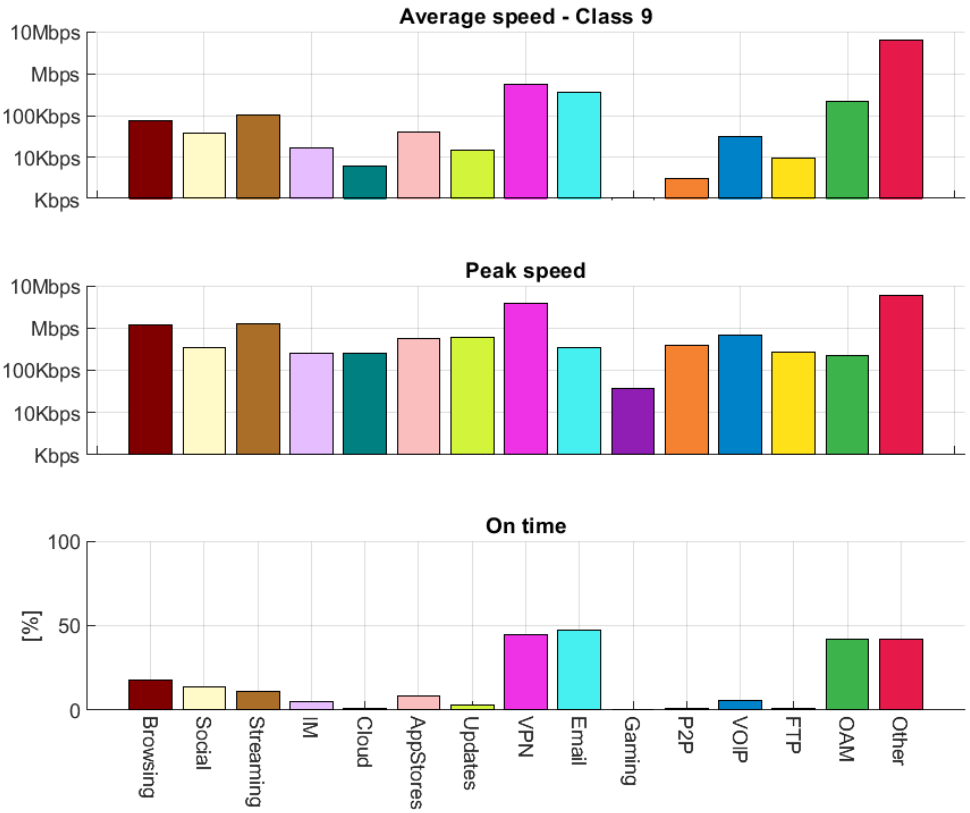

Figure 4.4.9 - User Features Class 9

The ninth category was found to have users that performed mostly VPN traffic with a very high speed. It can be identified as TeamViewer, or either any of the remote VPN program the responsible for that behavior. These programs allow a user to control a computer remotely, to connect business building far from each other creating a virtual local area network or to bypass region limitation present is some countries. This theory is validated by the fact that the VOIP, and FTP traffic is much more present in this class with respect to others, sign that confirms that the users used a VOIP call and transferred data between them also using some P2P protocol. Regarding the remaining protocols, it can be noticed the usual presence of the Streaming and Browsing as top peak speed and Email, OAM and Other where the average and the peak are very similar.
This category by the way is the least populous among the identified ones, since only 19 out of 7655 users were found adhering the average features represented in the graphs. Nevertheless, this category is considered important due to the peculiar behavior of the users that use VPNs.

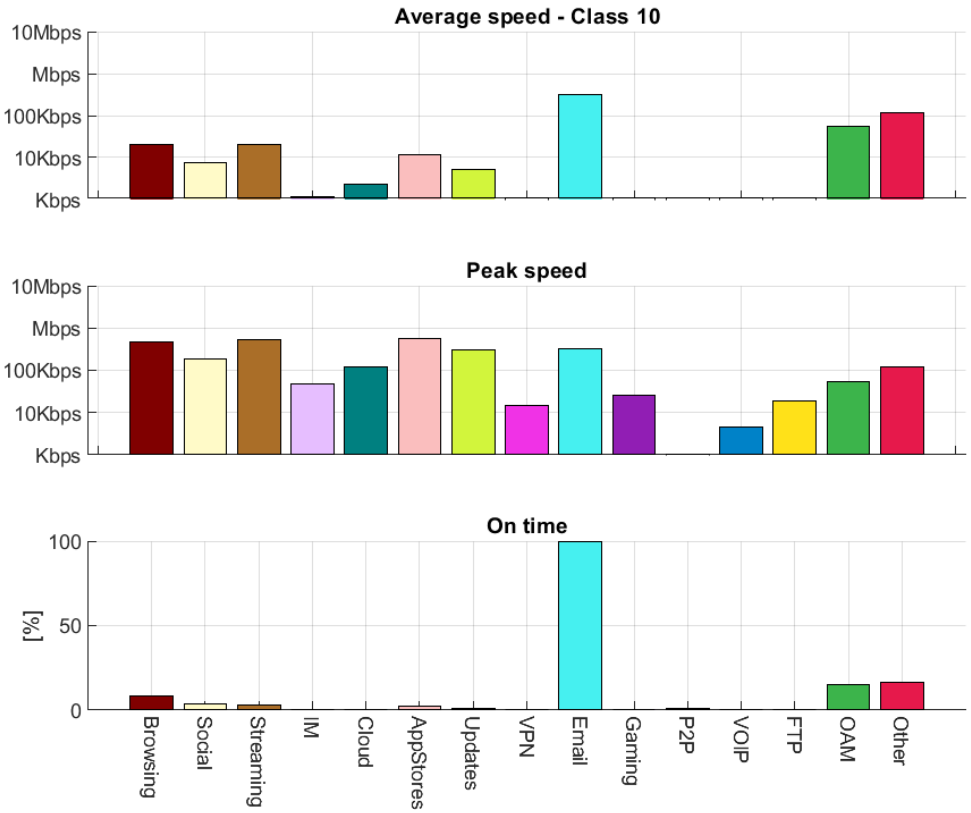

Figure 4.4.10 - User Features Class 10

The last identified category is the strangest one regarding the OnTime behavior: users belonging to this class had their Email protocols always active during the day, with a pretty high average speed that, due to the 100% OnTime, coincided (obviously) to the peak speed at around 800 Kbps. Maybe these users hosted some email servers, since the other protocols traffic was very low, even the OAM and Other. The only strange fact was that the numerosity of these users was very huge, since 1068 out of 7655 belonged to this class, making this the second most populous category. That's why the email server statement doesn't seem to be so much reasonable. By the way, it should be considered as a valid one due to its high cardinality.

4.5 User Percentage Results

After having analyzed all the categories that came out from the clustering of the features, it was decided to make another clustering but with the user percentage of type of traffic performed. In this way, it was more evident the type of traffic with respect to the amount of traffic that was performed by the average user identified in that class.

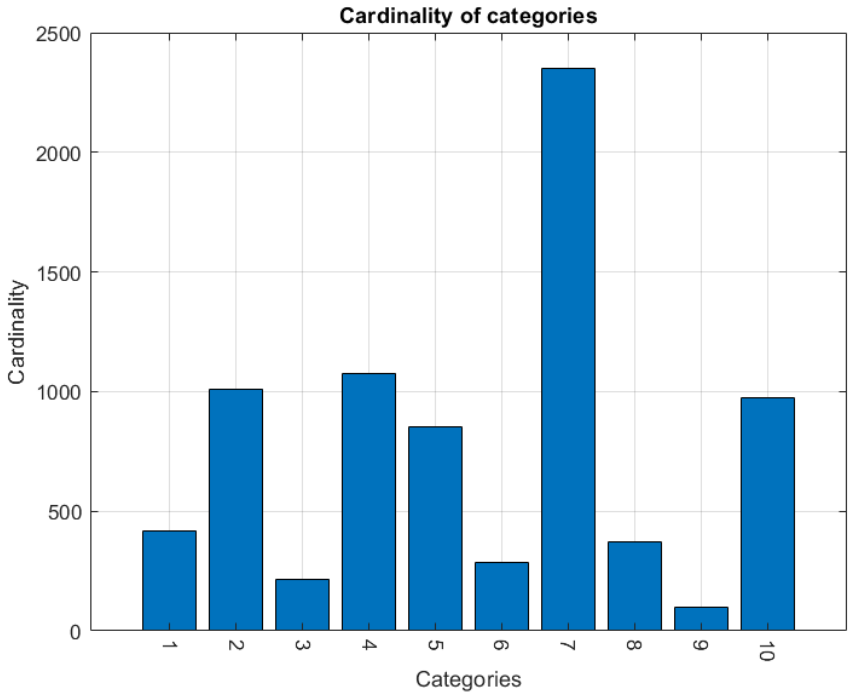

Figure 4.5.1 – User Percentage Class Histogram

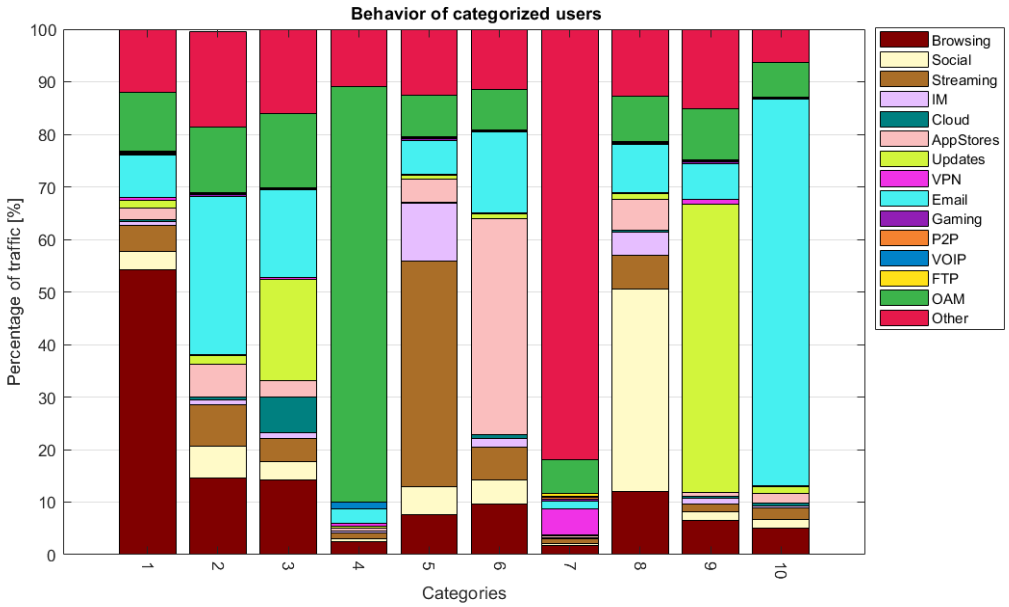

Figure 4.5.2 – User Percentage Class Division

Both the numerosity and the type of traffic performed by each category was different from our previous analysis. Therefore, each category needed a new discussion and interpretation.

Class 1 was identified as the **Browsing** one, since more than half of the traffic was performed with this protocol. The remaining slice was divided between the remaining protocol, out of each stand out the usual Email, OAM and Other.

Class 2 was identified as one of the **heterogeneous** class set, since every protocol obtained an almost equal slice of the total. Just the Email and the OAM seems to be more pronounced with respect to the others.

Class 3 discussion was more pushed toward an environment were Cloud and Updates were more present (it could be said to be an office one). The remaining part was almost equally divided by the other protocols.

Class 4 was one the **strangest**, were OAM traffic took over more than 85% of the total. Probably these were just dummy terminal used to collect statistic in a region.

Class 5 could be easily redirected to the videophile users, that performed **Streaming** protocol the most. Moreover, these users were the ones that performed more IM with respect to all the other categories.

Class 6 was the class that employed **AppStore update** the most, with the usual slice taken by the OAM, Other and Email. The VOIP, Gaming, P2P and VOIP didn't seem to be chosen by this category of users.

Class 7 was the one already identified in the class 4 of the previous subchapter. Here the data are contaminated by telemetry data not filtered out correctly by the DPI, that make this class appears to be just performing Other protocol. Some VPN traffic could also be spotted (5%) and very limited amount of the remaining protocols.

Class 8 was composed by users that used **Social Media app** and Browsing the most. Also Browsing, Streaming and AppStores were pretty evident, sign that these were the user that use mostly their phones (probably the tourist theory already described for class 9 of the previous analysis.

Class 9 was full of users that performed **Update** such as Windows or MacOS updates, since they occupied more than 55% of the total. The remaining part where just Browsing and minor contributions of the remaining protocol. Probably these are some office or school that just use the terminals for teaching or working.

Class10, the last one was the clear clone of class 10 of the previous analysis, **just Email** and very few other protocols used. Even using different type of input features their cardinality was relevant (around 1000 users).

In the following table it is possible to observe a summary of the evaluated categories obtained using as input the set of 45 features previously described.

Category	Feature	Browsing	Social	Streaming	IM	Cloud	App Store	Update	VPN	Email	Gaming	P2P	VoIP	FTP	OAM	Other
1	A	5,2	4,5	5,0	4,0	5,5	4,8	4,8	3,9	6,3	3,2	2,9	2,8	3,1	6,9	5,7
	P	6,3	5,6	5,9	5,4	6,7	6,2	6,2	5,0	6,3	4,8	3,6	3,9	5,0	6,9	5,7
	O	29,8	10,3	14,6	3,6	18,6	5,8	6,8	2,6	62,5	0,5	0,0	0,1	0,5	83,3	58,3
2	A	4,7	4,4	5,6	3,8	3,9	5,1	3,6	3,2	6,2	4,5	1,8	2,3	3,4	5,0	5,3
	P	6,0	5,6	6,7	5,2	5,5	6,7	5,3	4,4	6,2	5,9	2,8	4,0	5,2	5,0	5,3
	O	14,2	9,8	20,2	3,2	1,2	8,2	0,8	0,4	68,1	1,0	0,0	0,0	0,1	20,7	40,4
3	A	4,8	4,3	4,7	4,0	3,6	4,4	5,0	3,5	6,0	3,4	4,1	2,9	3,6	5,8	6,3
	P	6,0	5,5	6,0	5,3	5,3	5,9	6,5	4,7	6,0	5,0	6,0	4,1	5,3	5,8	6,3
	O	19,1	8,7	8,8	4,0	1,3	4,5	9,9	1,3	57,6	0,5	7,5	0,3	0,7	43,5	46,7
4	A	3,7	3,1	3,7	2,6	2,3	3,3	3,1	2,8	3,6	2,7	1,6	1,6	2,2	4,2	5,8
	P	5,2	4,6	5,1	4,1	4,1	5,0	4,9	4,0	3,6	4,1	2,7	3,1	3,8	4,2	5,8
	O	1,7	0,7	0,9	0,2	0,0	0,5	0,2	0,1	0,0	0,0	0,0	0,0	0,1	3,1	13,5
5	A	5,2	5,0	5,4	5,6	3,7	4,9	4,1	3,7	5,4	5,1	1,6	2,7	4,9	5,3	5,7
	P	6,1	5,8	6,3	6,2	5,0	6,0	5,3	5,1	5,4	5,8	2,9	4,1	5,9	5,3	5,7
	O	47,4	41,9	53,6	70,5	2,8	34,8	2,8	1,6	13,2	6,6	0,0	0,3	3,6	76,5	89,7
6	A	5,1	5,2	5,4	4,7	3,9	4,9	4,2	3,6	5,7	4,1	2,8	2,7	3,3	5,4	5,6
	P	6,2	6,1	6,4	5,8	5,5	6,2	5,8	5,1	5,7	5,6	3,7	4,1	5,1	5,4	5,6
	O	32,7	41,2	33,8	21,3	1,4	16,2	2,4	1,3	58,6	1,2	0,2	0,1	0,4	60,9	73,0
7	A	4,4	4,5	4,8	4,4	3,0	4,2	3,5	3,1	4,6	3,5	1,8	2,2	3,1	4,8	5,3
	P	5,7	5,7	6,0	5,6	4,7	5,8	5,2	4,4	4,6	4,9	3,0	3,7	4,9	4,8	5,3
	O	8,0	12,3	10,7	10,5	0,3	4,1	0,7	0,3	9,2	0,4	0,0	0,0	0,1	12,6	25,2
8	A	5,2	4,3	4,9	3,8	3,8	4,3	4,2	3,0	5,8	3,4	1,9	2,8	3,1	5,8	5,6
	P	6,4	5,6	6,1	5,2	5,4	5,9	5,8	4,4	5,8	4,8	2,7	4,3	4,9	5,8	5,6
	O	26,3	9,3	10,9	2,6	1,5	3,5	2,8	0,4	76,3	0,3	0,0	0,1	0,2	61,2	38,8
9	A	4,9	4,6	5,0	4,2	3,8	4,6	4,2	5,7	5,5	3,0	3,5	4,5	4,0	5,4	6,8
	P	6,1	5,5	6,1	5,4	5,4	5,8	5,8	6,6	5,5	4,6	5,6	5,8	5,4	5,4	6,8
	O	17,7	13,9	10,9	5,2	0,8	8,3	3,1	44,4	47,4	0,4	1,1	5,5	0,7	42,1	42,1
10	A	4,3	3,9	4,3	3,1	3,4	4,1	3,7	2,8	5,5	2,7	1,8	2,3	2,5	4,7	5,1
	P	5,7	5,3	5,7	4,7	5,1	5,8	5,5	4,2	5,5	4,4	2,9	3,7	4,3	4,7	5,1
	O	8,5	3,7	3,2	0,5	0,5	2,0	1,0	0,2	99,6	0,1	0,0	0,0	0,1	15,1	16,2

Table 4.4.11 - User features categories summary

Legend

- A: Average speed
  value obtained as $\log_{10}$(speed) (e.g. 3 ~ Kbps, 5 ~ 100Kps, 6 ~ Mbps)
- P: Peak speed
  value obtained as $\log_{10}$(speed) (e.g. 3 ~ Kbps, 5 ~ 100Kps, 6 ~ Mbps)
- O: ON Time
  percentage of time the speed was above 30 Kbps

In this table, the summary of the categories obtained using the percentages of type of traffic is displayed.

Category	Browsing	Social	Streaming	IM	Cloud	App Store	Update	VPN	Email	Gaming	P2P	VoIP	FTP	OAM	Other
1	54,3	3,4	5,1	0,7	0,3	2,2	1,5	0,4	8,2	0,2	0,1	0,2	0,1	11,2	12,0
2	14,6	6,0	7,8	1,0	0,6	6,1	1,8	0,2	30,0	0,3	0,0	0,2	0,1	12,5	18,2
3	14,1	3,5	4,4	1,1	6,8	3,2	19,3	0,3	16,6	0,1	0,0	0,2	0,2	14,0	16,1
4	2,5	0,5	1,1	0,3	0,1	0,5	0,4	0,6	2,7	0,0	0,0	1,2	0,0	79,1	10,9
5	7,6	5,4	42,9	10,9	0,3	4,4	0,8	0,2	6,3	0,5	0,0	0,1	0,2	7,9	12,5
6	9,6	4,5	6,3	1,6	0,8	41,2	0,7	0,3	15,3	0,3	0,0	0,1	0,1	7,7	11,4
7	1,8	0,4	0,8	0,2	0,0	0,3	0,2	5,0	1,4	0,4	0,4	0,1	0,7	6,4	81,9
8	12,0	38,5	6,5	4,5	0,3	5,8	1,1	0,2	9,1	0,2	0,0	0,2	0,1	8,6	12,8
9	6,5	1,7	1,5	1,0	0,4	0,7	54,9	1,0	6,8	0,3	0,0	0,1	0,2	9,7	15,1
10	5,1	1,6	2,2	0,4	0,4	1,8	1,4	0,1	73,7	0,1	0,0	0,1	0,1	6,6	6,4

Table 4.5.3 - User percentage categories summary

Chapter 5
Identified categories

In this chapter the previously found categorizations were applied to different level of grouping of the users. At first, the users were grouped by the different products that they had activated, considering all the data and speed caps that characterized a particular product. Then, they were aggregated under different population, that is the way in which user belonging to different vendors were tagged. Finally, the analysis was extended to beam grouping, considering indistinctively users that belong to a certain geographical area of the satellite coverage.

5.1 Used Metric

The metrics used to evaluate which was the class that mostly resembled that specific grouping behavior was the simple, yet effective, Euclidean Distance. The previously found categories was assumed as centroids in the N^{th} dimensional space (15 dimensions for the percentages analysis and 45 dimensions for the features analysis) and each candidate group Euclidean Distance with respect to every centroid (that were 10, as the number of categories) was evaluated. To write it in a more formal way:

$$D_{i,c} = \sum_{d=1}^{N} \sqrt{(x_{i,d} - C_{c,d})^2}$$

Where

- $D_{i,c}$ is the Euclidean Distance between group i and the centroid of the category c
- $x_{i,d}$ is the component along the d^{th} dimension of group i
- $C_{c,d}$ is the component along the d^{th} dimension of the centroid of the category c
- N is the number of dimensions

5.2 Products analyses

The first considered level of grouping was based on the product activated by each of the considered users. Each product was characterized by specific monthly data caps, Committed Information Rate (the guaranteed minimum speed provided) and Peak Information Rate (the maximum speed allowed). Due to company restriction and privacy it was possible to show just a product ID, and not all the other product information. By the way, it is possible to appreciate the categorization in action for this task. For each product, different features were extracted and compared to the previously found categories.

The first analysis concerned the percentage of type of traffic that users having a product performed. The total number of products considered was 182.

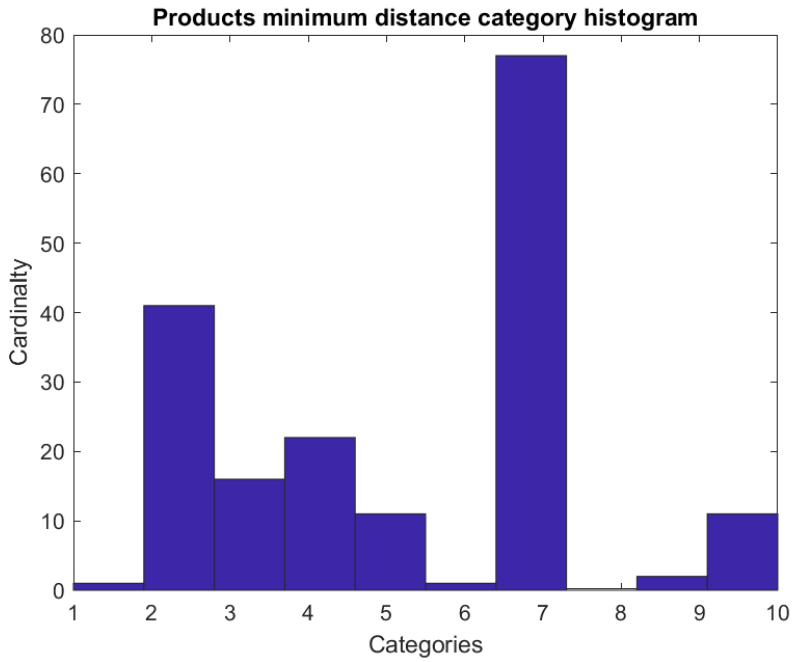

Figure 5.2.1 – Product Found Categories Histogram (Percentage)

By this first graph, it can be noticed how many products were classified for each category. In the previous chapter it was clear that the **7-th category was the most popular** among the users, and here the same behavior is confirmed even looking at a grouping of them. On the other hand, the opposite behavior is present, since category 1, 6 and 8 are almost empty, sign that those category behaviors were far from the products one, or that specific category didn't describe a specific behavior but just a general one.

Looking at some examples, the related behavior between minimum distance category and product average behavior can be perfectly appreciated.

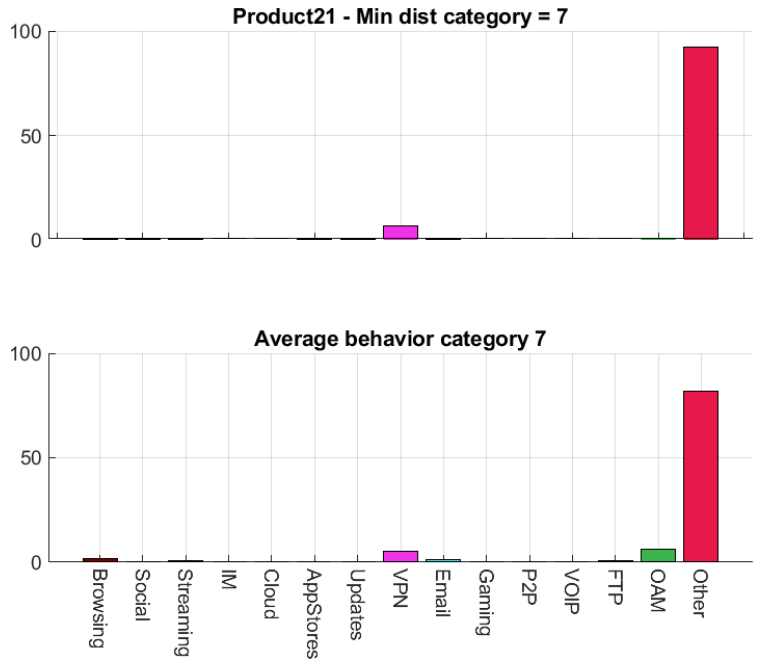

Figure 5.2.2 – Product 21 and Category 7 behavior comparison

As an example, product 21 minimum distance category was the 7-th one. In fact, their average percentage of traffic behavior are surprisingly similar.

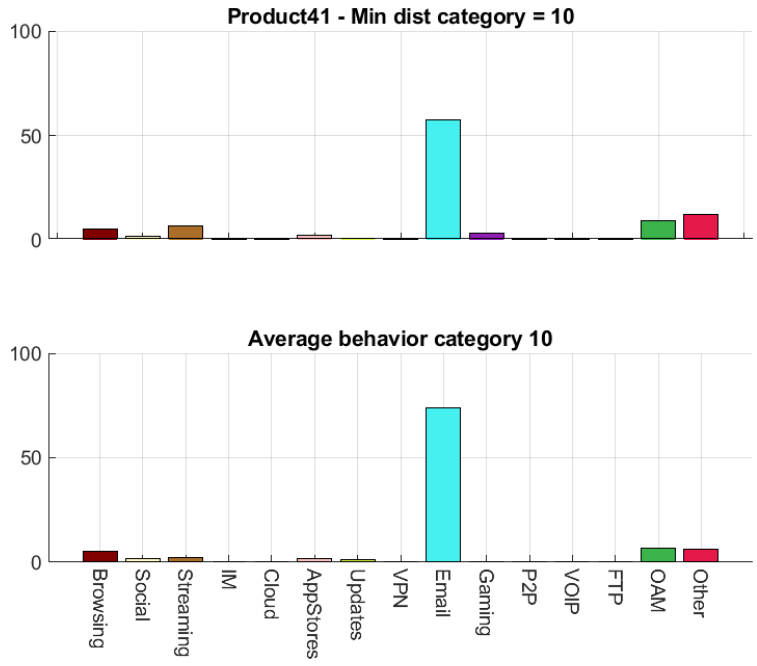

Figure 5.2.3 – Product 41 and Category 10 behavior comparison

The same reasoning applies to product 41, which was recognized belonging to category 10. Their general behaviors are similar, even if they slightly differ for some small active protocols (like Streaming) and the Email protocol percentage is some tenth different.

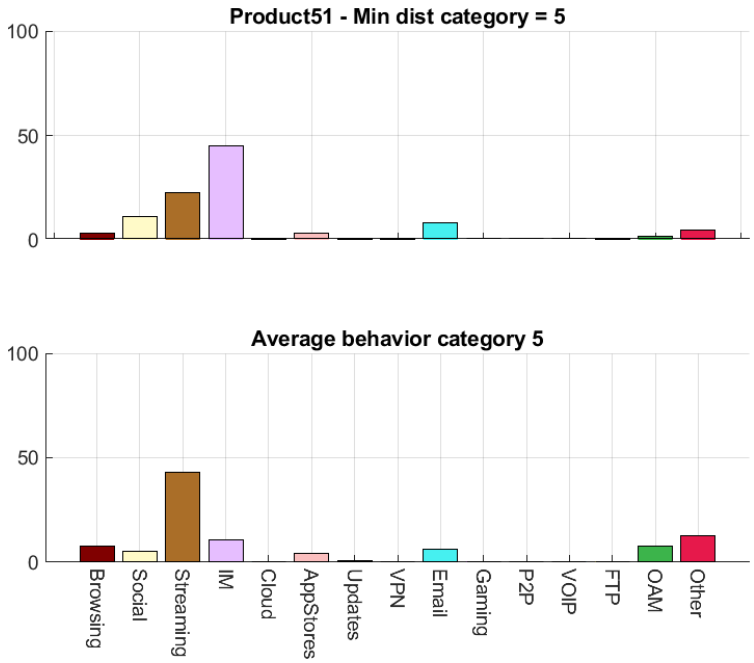

Figure 5.2.4 – Product 51 and Category 5 behavior comparison

Sometimes also strange behavior appeared, were user were attributed to specific category that actually performed different main type of traffic. This can be explained by the fact that a category is just an average representation and it is still very unlikely to have the exact same behavior when we compared the product with a limited number of categories.

The second analysis performed consisted in using the previously described set of features (AVG, PEAK and ONTime) to characterize a single product and evaluate the distances between the feature's centroids of the category and the ones found in the previous chapter.

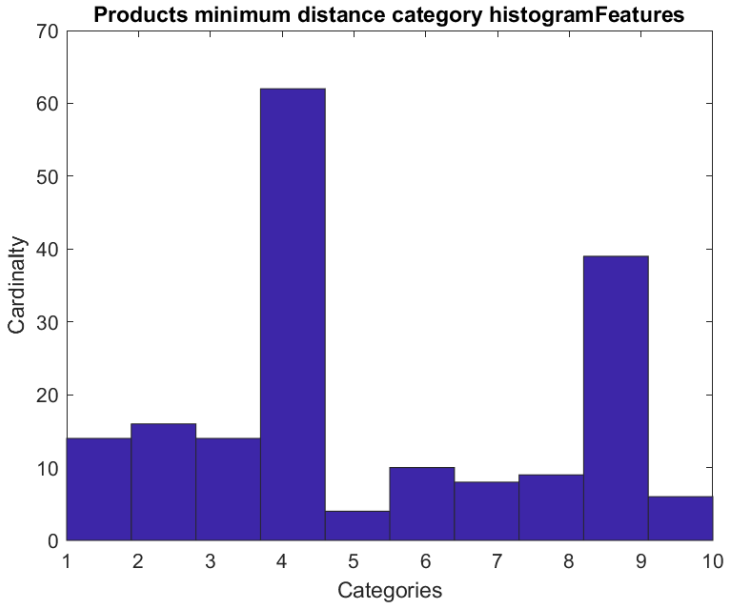

Figure 5.2.5 – Product Found Categories Histogram (AVG, PEAK, ONTime)

At a first glance, a different behavior with respect to percentage analysis can immediately be seen. In fact, **the most crowded category is the 4th one**, (exact same behavior as previous chapter category analysis). Furthermore, no so empty categories can be spotted. Instead it seems like all the products, apart from category 4, 5 and 9 are equally distributed among the remaining categories.

Looking at some samples, it is very difficult to find product average features that exactly resemble the features of the found category.

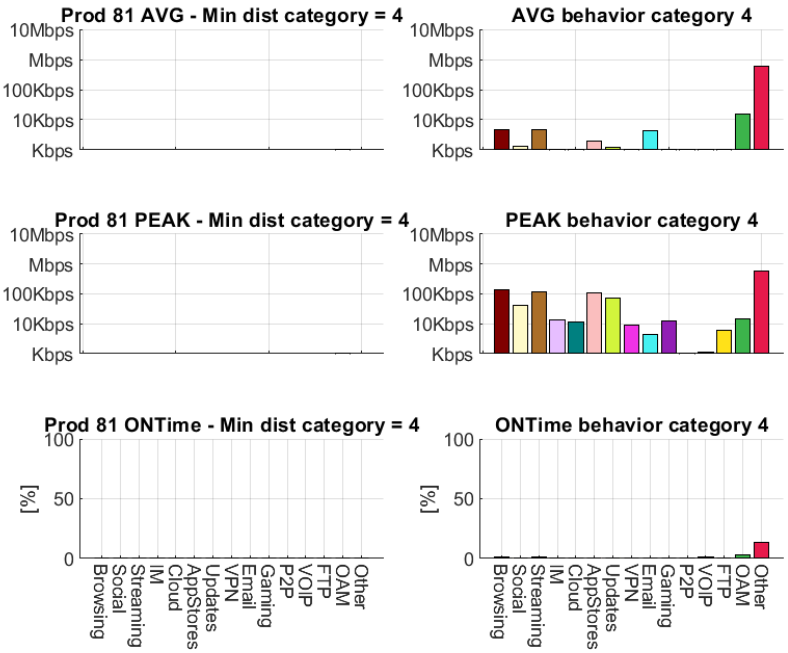

Figure 5.2.6 – Product 81 and Category 4 behavior comparison

In fact, some mistakes appeared for peculiar product who wasn't active at all, since in our previous category decision no always-silent category was envisioned. Actually, the Euclidean Distances from this product and all the centroids were pretty high and almost equal. The algorithm just selected the smallest one without looking at the difference with respect all the other distances.

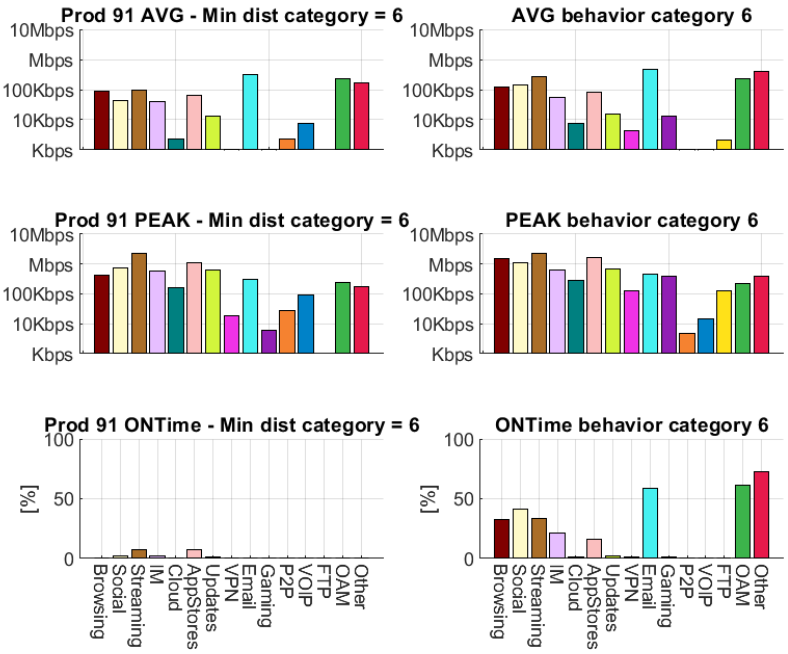

Figure 5.2.7 – Product 91 and Category 6 behavior comparison

Another example is this one, were a similar behavior in the AVG and PEAK speed for each protocol can be spotted. Just the ONTime was a bit off track, probably due to the way over which the activity time was measured.

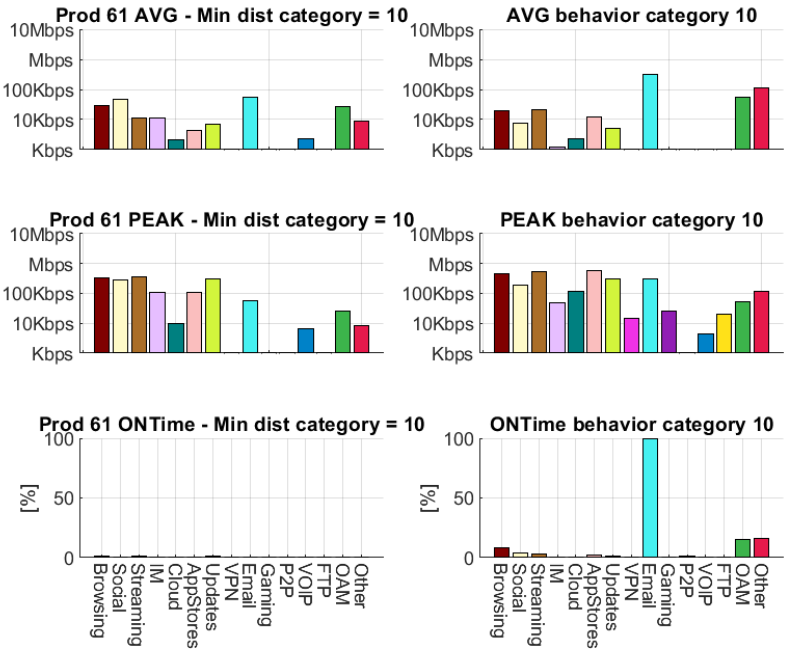

Figure 5.2.8 – Product 61 and Category 10 behavior comparison

Even this example is affected by the previously described flaw, since the AVG and PEAK speeds were comparable, but the ONTime wasn't.

5.3 Populations analyses

The second type of users grouping was made combining in a single set all the users that belonged to a specific population. A population is a group made by user that are served by the same vendor. Sometimes it can happen than a vendor host more than one population. The users grouped in a single population can be very far one from each other, and they can also be served by different beams. Due to the previously said characteristics, a medium level heterogeneity is obtained, and some analysis can be performed to spot if certain vendors serve a particular or random type of user.

The first analysis concerns the percentage of the type of traffic performed by each population. Using the previously described metric, each population was labelled with the "nearest" centroid class number. The number of populations was 124.

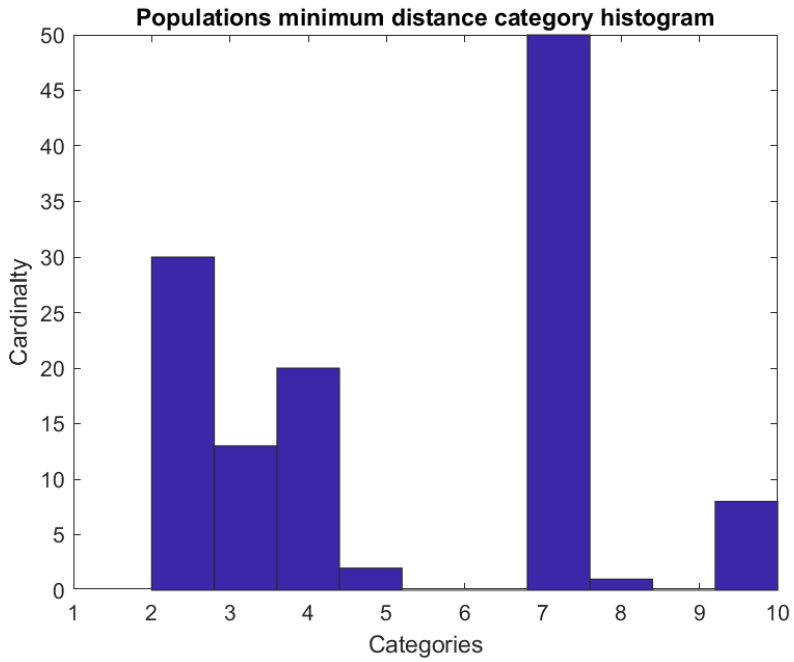

Figure 5.3.1 – Populations Found Categories Histogram (Percentage)

At a first glance, this graph seems to resemble a lot the one of the product grouping. Of course, the general behavior that sees **the 7th class** to be **most crowded** is maintained. But here some differences can be spotted, starting from the different distribution of populations among the found category, finishing to the emptiness of some category like the 1st, the 6th and the 9th. Probably these last three weren't so similar to the average behavior spotted in the populations extracted for our analysis.

In fact, in the following examples we have found some singular behavior for some populations.

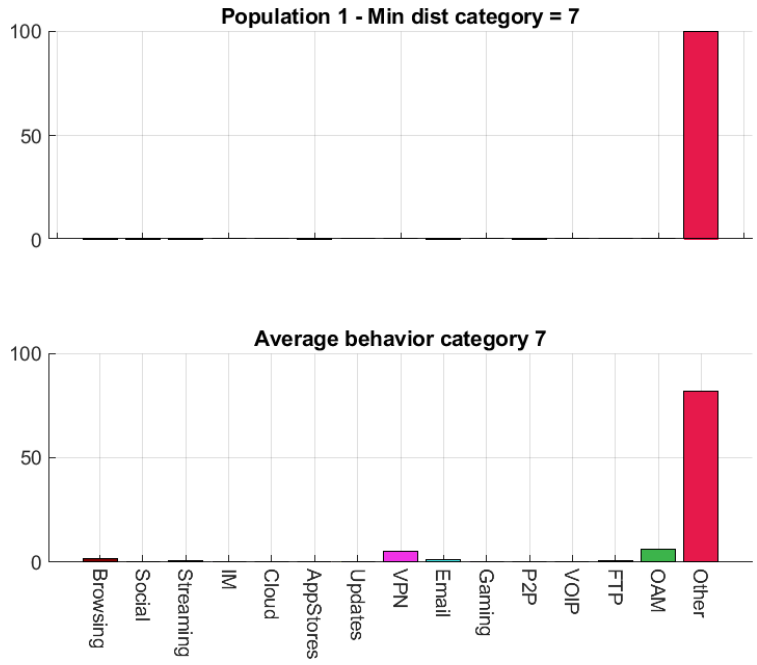

Figure 5.3.2 – Population 1 and Category 7 behavior comparison

From this graph it can be seen that the labelling can be correct, since the category and the population average behavior differs slightly in percentage for the most active protocol Other, and also for some seldom active protocol like VPN and Email.

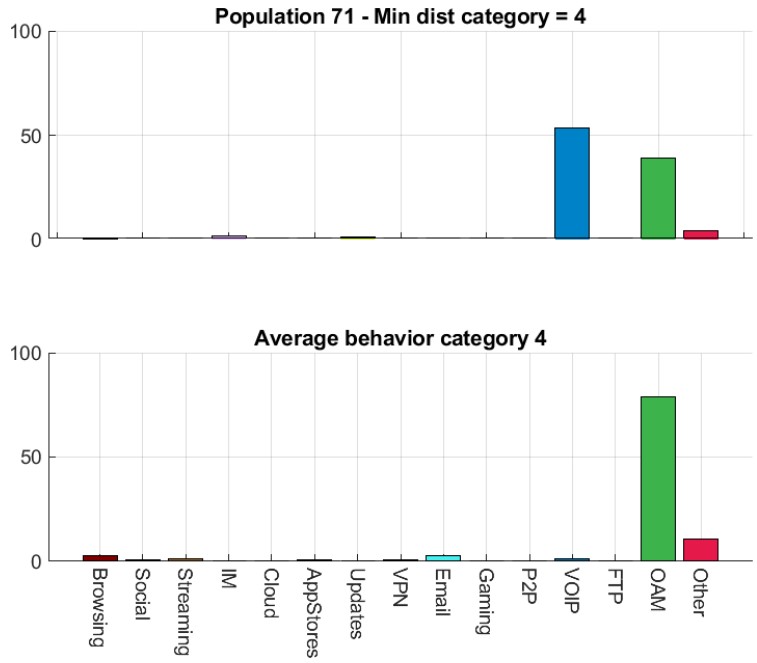

Figure 5.3.3 – Population 71 and Category 4 behavior comparison

In this case, the similarity is not perfect, but the population and category resemble each other. In fact, the OAM and Other ratio are almost respect, a part for some magnitude

differences, but the VOIP traffic is completely absent. Furthermore, the category most active protocol, is not the population most active protocol, fact that can lead to eventual misjudgments.

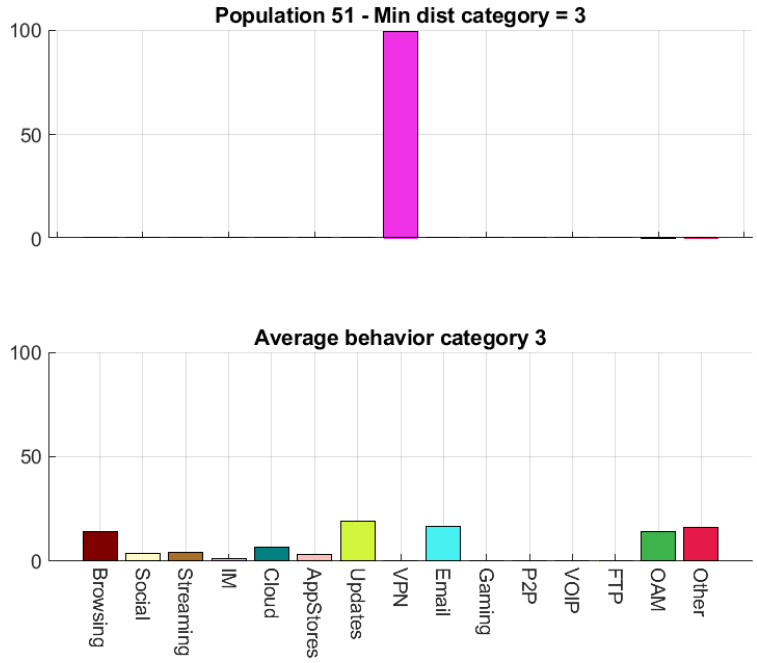

Figure 5.3.4 – Population 51 and Category 3 behavior comparison

In this last example it can be noticed that the labelling almost completely failed to give an average representation of the population. In fact, the category seems to be the opposite of the average population behavior. After some investigations it was found that the distances of this population between all the found category was always high and almost equal. Therefore, the algorithm just took the smallest one among them, even if the distance was pretty high. This is actually one of the flaws of this simple approach, since the distance should be evaluated, thresholded and, more importantly, compared to the other distance to give a level of confidence in the labelling procedure.

The second analysis was performed collecting the triplet of features AVG, PEAK and ONTime from the population and comparing it with the one of the found categories.

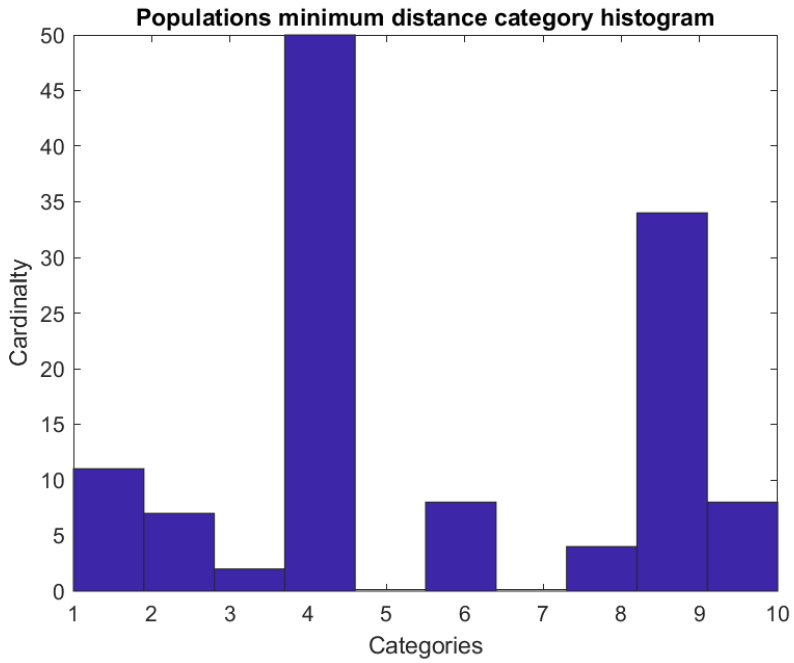

Figure 5.3.5 – Populations Found Categories Histogram (AVG, PEAK, ONTime)

From this graph the same average behavior of the product analysis can be spotted. In fact, the 5th category iw almost empty while the **4th and 9th** are the ones with the **higher number** of populations. Just the 7th here seemed to be less descriptive since less population received that label.

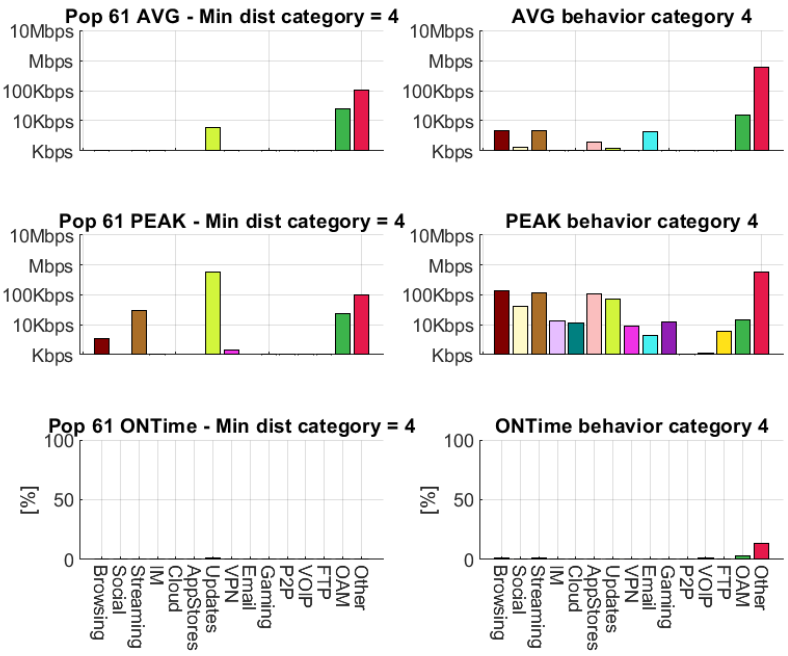

Figure 5.3.6 – Population 61 and Category 4 behavior comparison

This example shows that the population 61 showed an average behavior that wasn't so likely the category 4, but still it was the nearest one. In fact, the population most active protocol was the Updates and Streaming one, while the category focused more on Other and OAM.

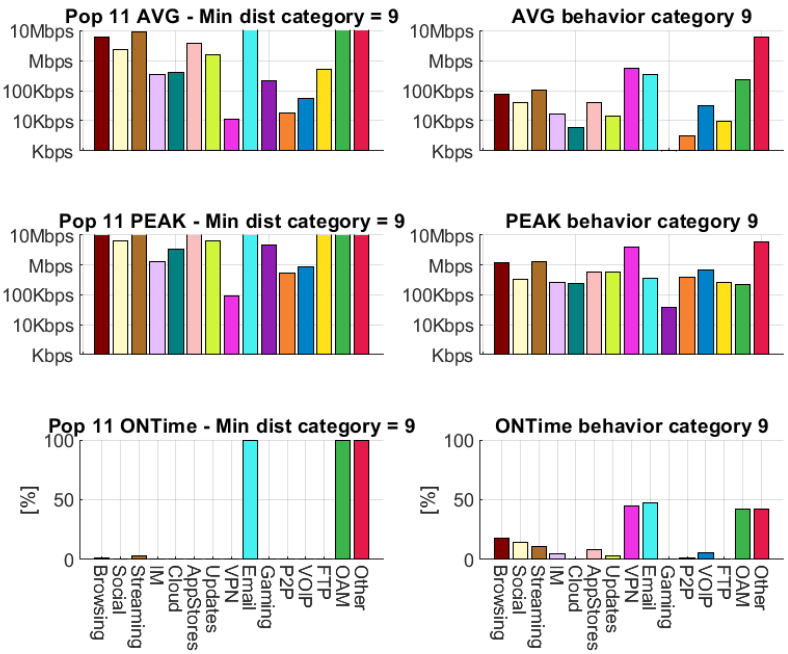

Figure 5.3.7 – Population 11 and Category 9 behavior comparison

This effect continues all along the populations, were it was difficult to find an average behavior category that looked like the average population behavior. This is also evident in this example where the population performed a lot of traffic on almost every protocol, but the percentages are not similar to the one of the categories.

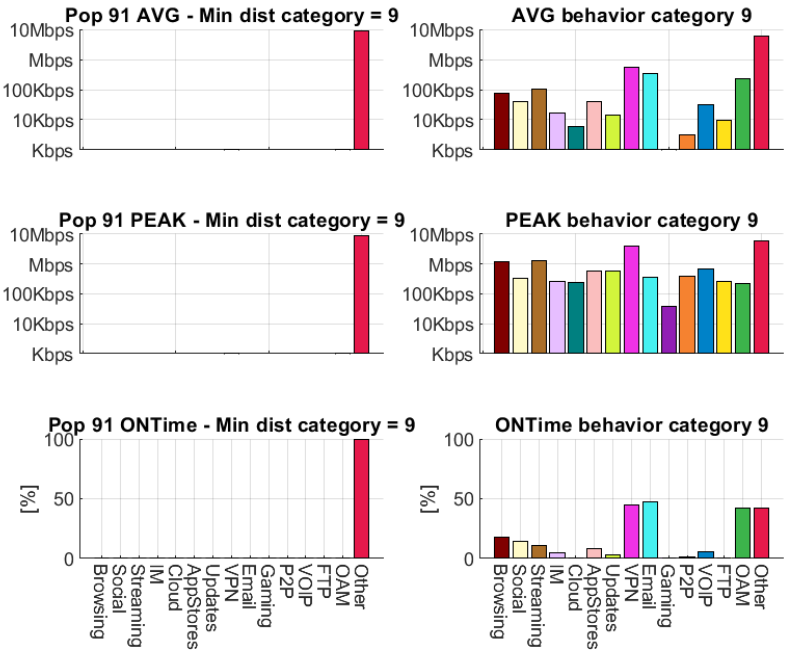

Figure 5.3.8 – Population 91 and Category 9 behavior comparison

The opposite reasoning can be made for this graph, where the exact opposite behavior happens. The population always active on just one protocol is totally different from the assigned average category behavior.

In general, it was noticed that when the set were more crowded, the average population behavior drifted away from the evaluated category. A possible explanation could have been due to the way that categories were evaluated: in fact, they were calculated basing their percentage on single users' behavior and not wide group of users. A trend is appreciable since in the following part, huge group of users were collected and labelled.

5.4 Beams analyses

In this last set of analysis, the granularity at which the users were collected reached the maximum available scale. In fact, they were grouped according to their beam that they were under to. A beam, in our analysis, is the geographical area covered by an antenna placed onto the satellite, that emits the signal around a central frequency and with a specific polarization. Under a beam, users with different products and populations are served, so the level of heterogeneity is maximum.

The number of users for each beam could range from hundreds to thousands, depending on the amount of people in the area covered by it. This analysis covered just three of them, due to the high number of total users (7655) each one characterized by their daily profile, percentage and set of features.

The beam number is arbitrary and is uncorrelated to the beam real number due to company policies.

The first analysis performed was the usual percentage one. Due to the exiguous number of beams, in the graph below it can be seen the label assigned to each beam, instead of the previously used histogram.

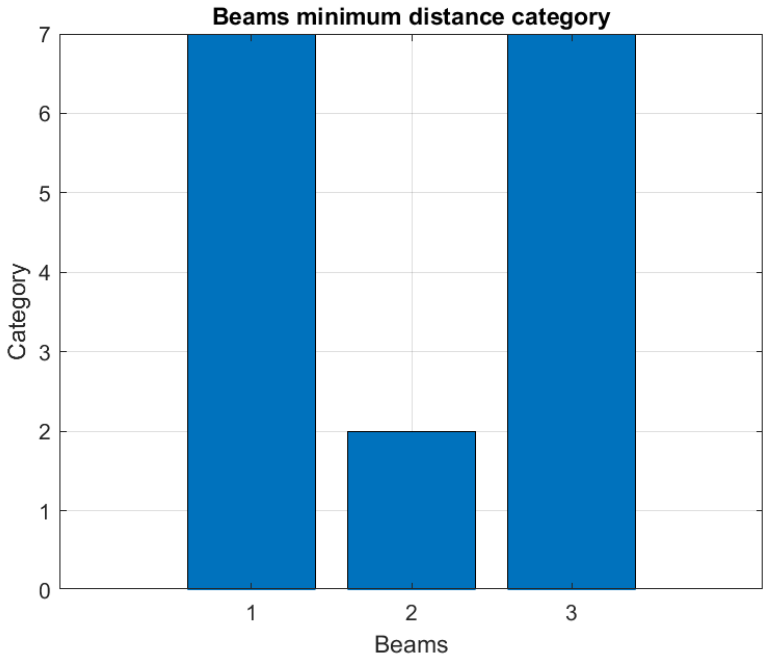

Figure 5.4.1 – Beam minimum distance category labelling (Percentage)

Few considerations can be made out it, apart from the fact that the 1st and 3rd beams falls in the usual, most populous **7th category**.

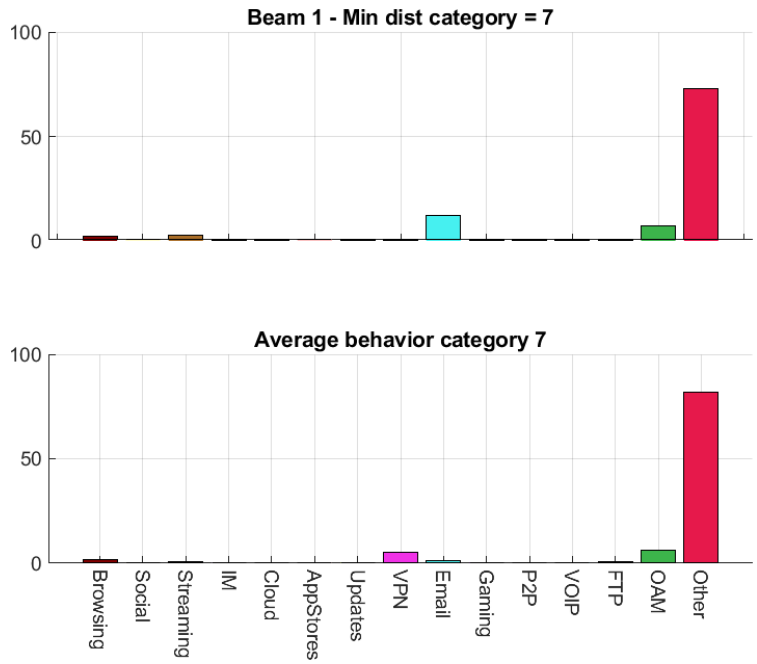

Figure 5.4.2 – Beam 1 and category 7 behavior comparison

Regarding the 1st beam, a pretty good similarity can be appreciated between its average behavior and the category average behavior. In fact, most of the traffic was performed by the Other protocol similar to the category one. Just the Email type differed to it.

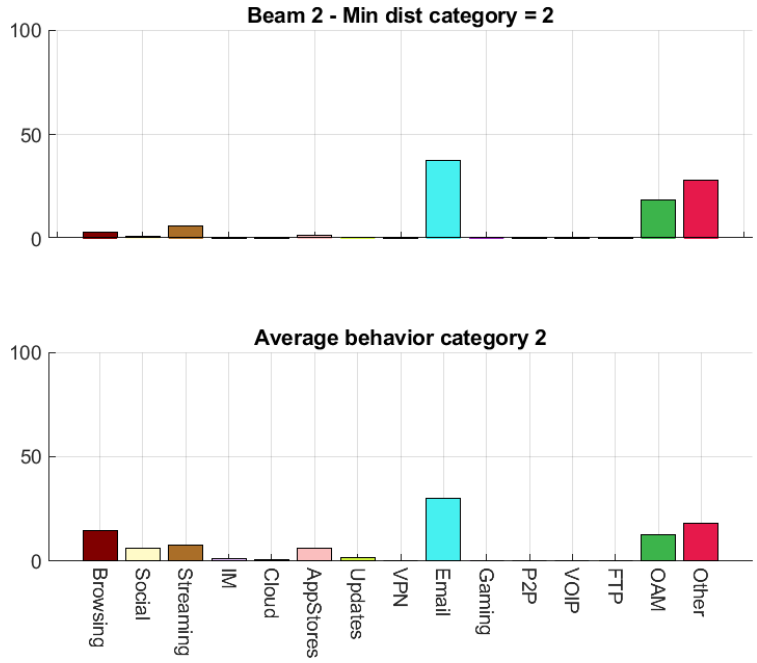

Figure 5.4.3 – Beam 2 and category 2 behavior comparison

For what concern beam number 2, the same consideration as before are valid. In fact, the Email, Other and OAM superiority is equally present in both beam average behavior and category average behavior. Almost the same is also valid for the other protocols, where it is noted that the Streaming and Browsing protocols are, let's say, the second order matter that characterizes the traffic performed under this beam.

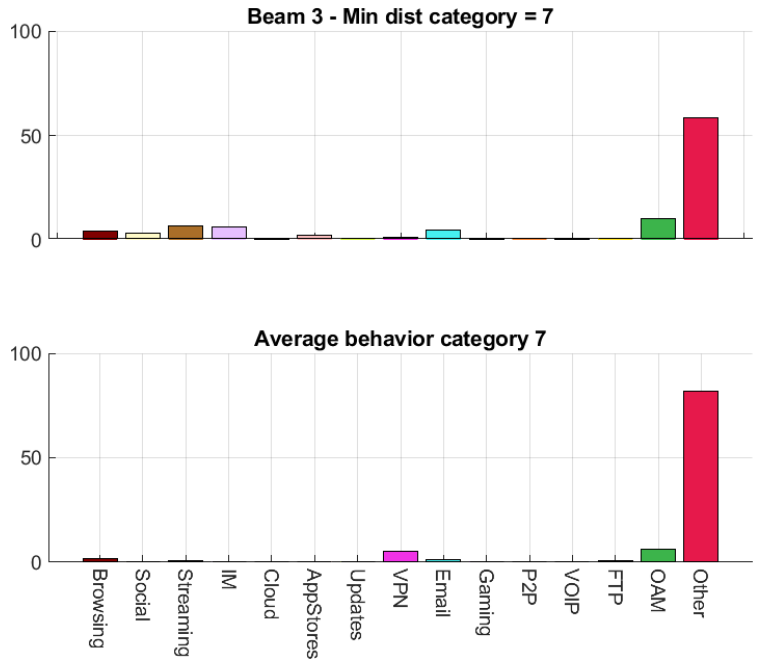

Figure 5.4.4 – Beam 3 and category 1 behavior comparison

For the last analyzed beam, some discrepancy can be detected. In fact, even if the Other and OAM protocols superiority is confirmed, the small, but still important, traffic percentage performed by the remaining protocols like Streaming, IM, Emails and Browser is not perfectly described by the average behavior of category 7.

This confirms the previously spotted trend that sees less accurate results as we move from finer granularity, like the user level, to gross granularity, like beam level.

The last analysis performed consisted in labelling the extracted triplet of features, AVG, PEAK and ONTime from the beams and evaluating the smallest distance between them and the found 10 categories set of features.

As previously said, just the final labeling and no histogram are displayed due to the exiguous amount of data to be displayed, that consisted in just three beams.

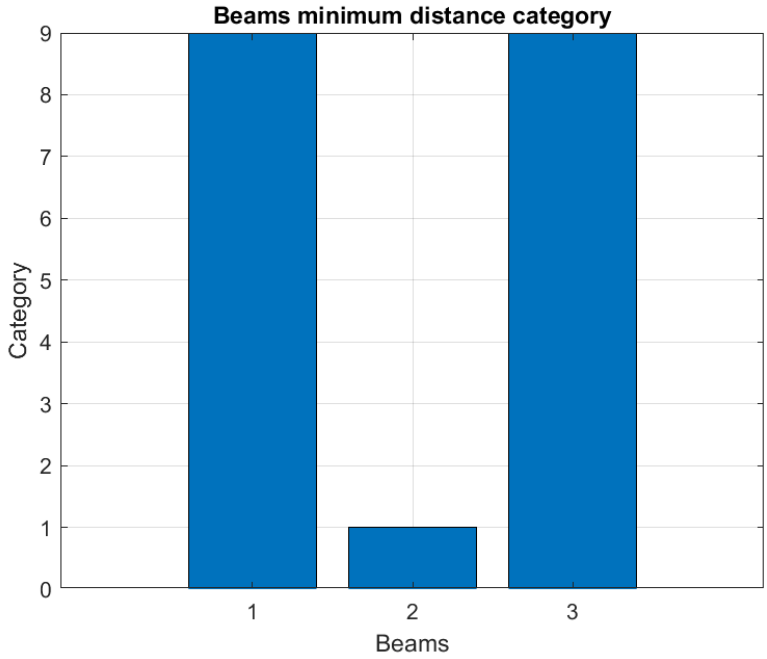

Figure 5.4.5 – Beam minimum distance category labelling (Features)

Surprisingly, this last analysis assigned to the three beams label that were not so common in the previous set of analysis. In fact, the 1st category was pretty empty, while the **9th** one was the second most more crowded. Here after, the comparison between the beam average features and assigned category average features can be appreciated.

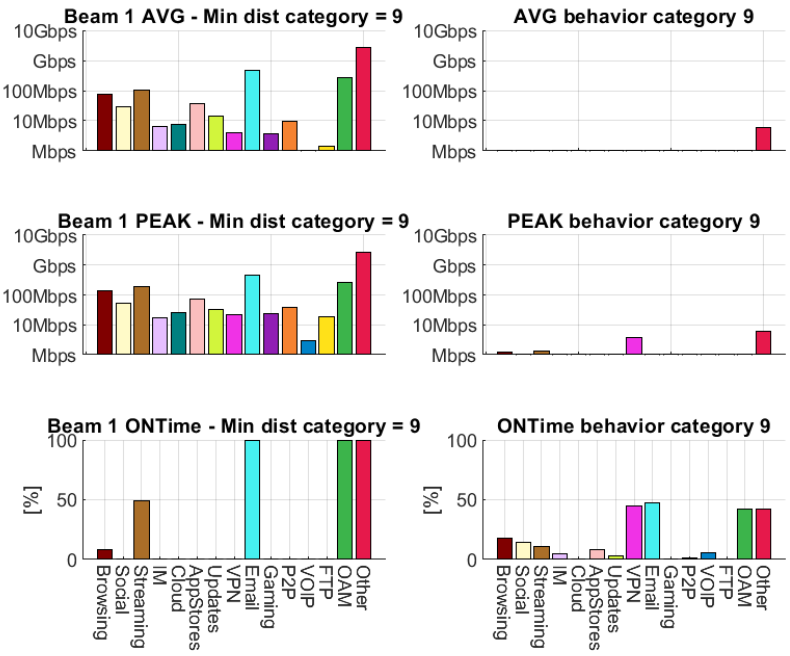

Figure 5.4.6 – Beam 1 and category 9 behavior comparison

For the 1st beam, the actual total data rate was evaluated. In fact, speed in the order of hundreds of Mbps can be appreciated. The similarity is not so evident, due to the fact that Beam most performed traffic are Other, Emails and OAM are in line with the category, while the Streaming peak is substituted by a VPN peak. Of course, it could have been possible to divide the total traffic with respect to the number of users served in that beam, but that would have resulted in just a scaled value. Moreover, the data would have lost their representativeness, because the average would have been evaluated on every user: active, silent, performing just one protocol or using all the protocols all together, with all the shades in between.

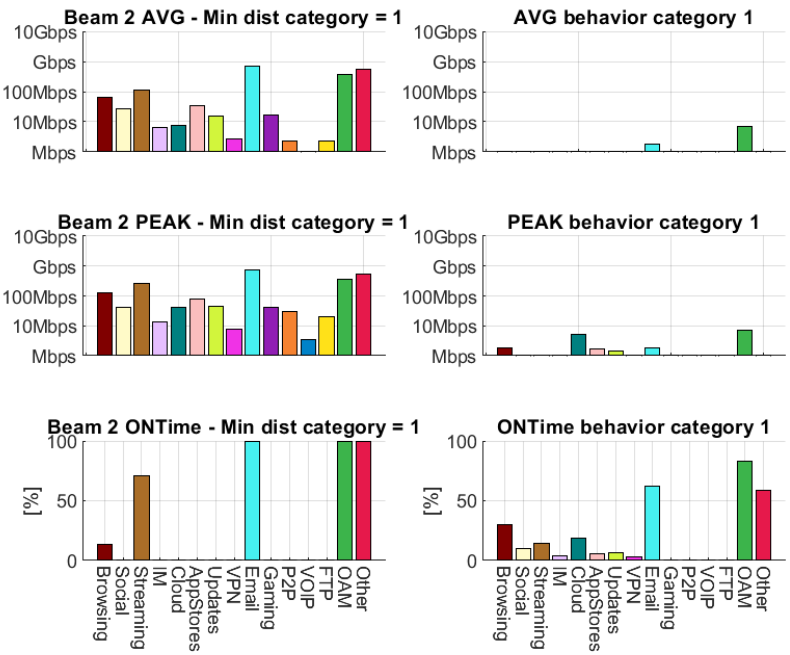

Figure 5.4.7 – Beam 2 and category 1 behavior comparison

Beam 2 ONTime seems to be very similar to beam 1 one, but it differs slightly in their values. Strangely, the assigned category differs from the 7th and is the 1st, characterized by a predominance of the usual OAM, Other and Email and also to the presence of the remaining ones. Focusing on the Browsing, that is also the fourth most used protocol in the category, so this particular aspect is present in both the set of features. As said before, the speeds are not comparable because of the total beam traffic.

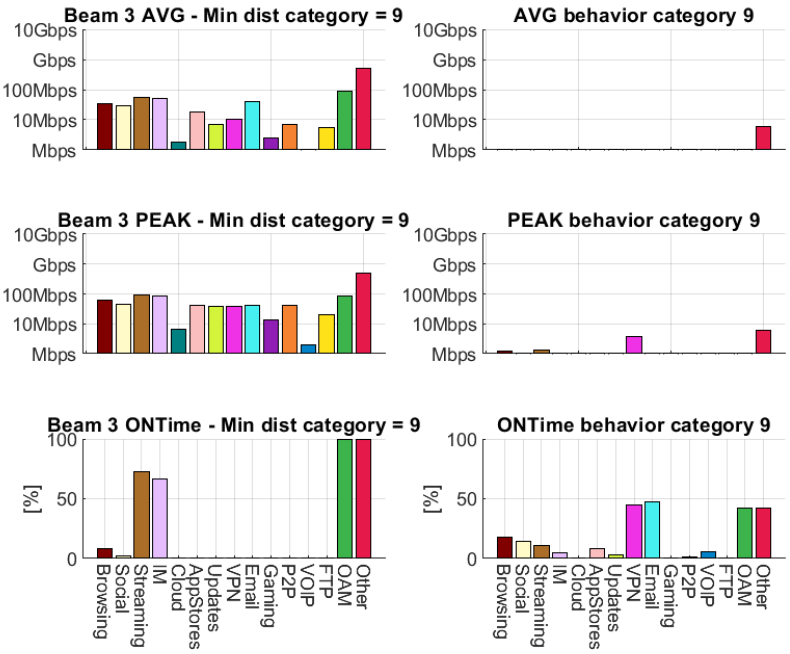

Figure 5.4.8 – Beam 3 and category 9 behavior comparison

The last beam considered in this analysis performed in a normal way, but the similarity between its behavior and the assigned category are not so much. In fact, the 9th category, even if it is the "nearest" among all the found ones, behaves in almost a completely different way: the VPN and Email are just very few active protocols in this beam, while the majority of the traffic is performed by the Streaming and IM ones. Due to this evident discrepancy it can be honestly said that this labelling was a wrong choice.

Now that the analyses are over, the trend that sees always less accurate results going from a smaller number of samples to a huge one can be confirmed.
In fact, the best results were obtained using the Percentage analysis either on the Users or grouping them by Products. The Population and Beam division still gave good results, but the assigned categories were not always so representative.
Instead, the usage of the 45 features, not only increased the computational effort to at first extract, then store and process the additional dimensions, but also performed in a worse way. Probably the 45 degree of freedom didn't quite match the just 10 found categories, even if that number was chosen following a reasonable metric. All the analyses, starting from the users, to the Products, Populations and Beams showed some isolate good results, but mostly vast inaccurate ones (like the one in Fig 5. and Fig. 5. showing respectively Pop 51 and Beam 3).

Chapter 6
Conclusion and future works

In this thesis, some possible ways of identifying the users of a satellite network has been employed. At first, we started by comparing different performances achieved by some machine learning algorithm, namely: Shallow Neural Network, Support Vector Machine, Random Forest for what concerned the supervised ones and K-means clustering for what concerned the unsupervised ones. Different ways of performing data manipulations were employed, ranging from the plain data, to the quantization of the speeds, finishing with the extraction of some features like Average Speed, Peak Speed and ON time.
Then, we enriched the input data with the information of the protocol with which the traffic was performed, obtained by a device which was performing Deep Packet Inspection. The input data were again manipulated to show on one side just the percentage of the type of traffic performed by each user, on the other side, the full range of previously evaluated features for each of the 15 aggregated protocols considered. Out of that, we decided to extract exactly 10 categories to represent all the more relevant users' behavior. This number was chosen according to the minimization of a metric that the K-means clustering algorithm provided.
Finally, this found categories were used as centroid of a multidimensional space based on the number of features that characterized the categories to measure the distances at which each group of users was, in order to assign to each of them the label of the nearest class. Different sizes and types of groups were considered, starting from collecting all the users that shared the same active product, to the users that shared the same vendors, finishing with the users that share a geographical location.

This thesis provided a wide range of results using the previously described method, but of course it was not immune to errors and misjudgments. Some hint can be given to continue this study: as an example, the data manipulations chosen in this work were somewhat arbitrary and came to the mind of the candidate due to previously attended course, like the averaging, the quantization, the three type of features presented etc. Very likely, there exist better manipulations that would have guaranteed better results for each of the analyzed algorithm.
Furthermore, the algorithm chosen were the ones provided by the MATLAB tools, that we realized were very basic and simple after working with them for 6 months. Also, no tuning was available in the wizard, and that limited a lot the refinement of them. So, considering using the discovered H2O library, available for Phyton and R, could be a good bet. More details can be found in [6].
Finally, regarding the chosen categories, it was clear that they were arbitrary evaluated by an automated algorithm, so surely not error-proof. They were considered as the reference simply because there were no other hints on how the users were expected to behave. Therefore, a deep study of them could also be beneficial to extract this kind of information.

References

1. Louis J. Ippolito, Jr. Satellite Communication System Engineering, 2008

2. J. G. Andrews, A. Ghosh, R. Muhamed, “Fundamentals of WiMAX: Understanding Broadband Wireless Networking,” Prentice Hall, 2007.

3. Kevin P. Murphy. Machine Learning: A probabilistic Perspective, 2012

4. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

5. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

6. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

7. Pennacchiotti, Marco, and Ana-Maria Popescu. "A machine learning approach to twitter user classification." Fifth International AAAI Conference on Weblogs and Social Media. 2011.

8. Moore, Andrew W., and Denis Zuev. "Internet traffic classification using bayesian analysis techniques." ACM SIGMETRICS Performance Evaluation Review. Vol. 33. No. 1. ACM, 2005.

9. Nguyen, Thuy TT, and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning." IEEE communications surveys & tutorials 10.4 (2008): 56-76.

10. Erman, Jeffrey, Martin Arlitt, and Anirban Mahanti. "Traffic classification using clustering algorithms." Proceedings of the 2006 SIGCOMM workshop on Mining network data. ACM, 2006.

11. https://satelit.web.id/what-is-a-satellite

12. https://en.wikipedia.org/wiki/Radio_spectrum

13. https://www.eutelsat.com/sites/eutelsat-internet/home/satellites/9-east.html#ka-sat

14. http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/

15. https://it.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5b6n