University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Cybersecurity

An Anomaly Detection System For Subscriber Activities In Private Networks

Supervisor
Prof. Simone Milani
University of Padova

Co-supervisor
Antonio Fin

Master Candidate
Francesco Martini

Student ID
2026830

Academic Year
2022-2023

Ai miei genitori
A mio fratello Davide
A tutti i miei parenti ed amici
Al nonno Carlo
Grazie dal profondo del mio cuore.

Abstract

This master thesis project contains the work carried out during the internship at Athonet, a Hewlett Packhard Enterprise acquisition. The primary objective of this work is to build a machine learning-based anomaly detection mechanism for the private 4G networks that the company deploys worldwide. Anomaly detection is a critical and important task that finds application in different domains, such as fraud detection, cybersecurity, system monitoring, bug discovery and many others. IT companies are investing in this field to improve the performances and the security of the systems that are selling to the customers. However, an anomaly detection system cannot be realized without the help of machine learning, due to the huge amount of traffic data coming from the different devices connected to the company's private networks. Machine learning is a branch of artificial intelligence and focuses on developing algorithms and models that allow computers to learn from data and improve their performance over time. With this indispensable tool, it is possible to automatically find anomalies and frauds in the networks, discover brand new bugs and inconsistencies in the data retrieved from the subscribers activities. This project presents and describes the processes of collection and storage of the Charging Data Records generated from the subscribers connected to a private networks. The work shows the strategies adopted to extract a dataset with significant features and presents different machine learning models whose results were evaluated using the charging data records of an Athonet's client.

Contents

Listing of figures

Listing of tables

Listing of acronyms

M2M ..... Machine To Machine

NPN ..... Non-Public Network

3GPP ..... Third Generation Partnership Project

CDR ..... Charging Data Record

NG-RAN ..... Next Generation Radio Access Network

USIM ..... Universal Subscriber Identity Module

UE ..... User Equipment

IMEI ..... International Mobile Equipment Identity

IMSI ..... International Mobile Subscriber Identity

MCC ..... Mobile Country Code

MNC ..... Mobile Network Code

MSIN ..... Mobile subscriber identification number

MIMO ..... Multiple Input Multiple Output

RAN ..... Radio Access Network

E-UTRAN ..... Evolved Universal Terrestrial Radio Access Network

eNodeB ..... Evolved NodeB

EPC ..... Evolved Packet Core

MME ..... Mobility Management Entity

S-GW ..... Serving Gateway

P-GW ..... Packet Data Network (PDN) Gateway

HSS ..... Home Subscriber Server

GUTI . . . . . Globally Unique Temporary ID

GUMMEI . . . . . Globally Unique MME ID

M-TMSI . . . . . MME-TMSI

EPS . . . . . Evolved Packet System

EEA . . . . . EPS Encryption Algorithm

EIA . . . . . EPS Integrity Algorithm

APN . . . . . Access Point Network

PCRF . . . . . Policy and Charging Rules Function

PCEF . . . . . Policy and Charging Enforcement Function

QoS . . . . . Quality Of Service

RAT . . . . . Radio Access Technology

NAS . . . . . Non Access Stratum

RRC . . . . . Radio Resource Control

CTF . . . . . Charging Trigger Function

CDF . . . . . Charging Data Function

CGF . . . . . Charging Gateway Function

BD . . . . . Billing Domain

IP-CAN . . . . . IP Connectivity Access Network

ASN.1 . . . . . Abstract Syntax Notation One

AMF . . . . . Access Mobility Function

SMF . . . . . Session Management Function

AUSF . . . . . Authentication Server Function

UDM . . . . . Unified Data Management

PCF . . . . . Policy Control Function

NRF . . . . . Network Repository Function

NSSF . . . . . Network Slice Selection Function

NEF . . . . . Network Exposure Function

UPF . . . . . User Plane Function

BER . . . . . Basic Encoding Rules

1 Introduction

Mobile communication has witnessed impressive advancements in last decades, changing the way we connect, communicate and access information. From the early days to today's 4G and 5G technologies, each generation has played a big role in our lives and brought important new innovations. The improved data speeds and low latency of 4G networks had a profound impact on the mobile application ecosystem, meeting the needs of billions of users in a fast-growing market. The introduction of improved performance and the provision of user-centric services such as real-time video streaming, online shopping, and high-quality video calls marked 4G as an important technological milestone [1]. 5G is the next step in the evolution of mobile communications. It is designed to provide greater rates of data transfer, lower latency, and improved reliability compared to previous generations. Moreover, 5G is not only designed for customer applications like in 4G, but also for industrial purposes such as M2M (machine to machine) communication [2]. With 5G, new innovations and opportunities in a variety of market verticals such as healthcare, manufacturing and transportation are becoming reality. According to the World Economic Forum[1], the number of mobile subscriptions are more than the global population. This overtake has already took place in 2022, where 8.59 billion of subscriptions were used worldwide, against the 7.95 billion of people (figure 1.1). This trend will continue and probably grow in the next years. Considering that the amount of people that have access to the internet is rapidly growing every year [2] (figure 1.2), it is possible to forecast the importance

[1] https://www.weforum.org/agenda/2023/04/charted-there-are-more-phones-than-people-in-the-world/
[2] https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

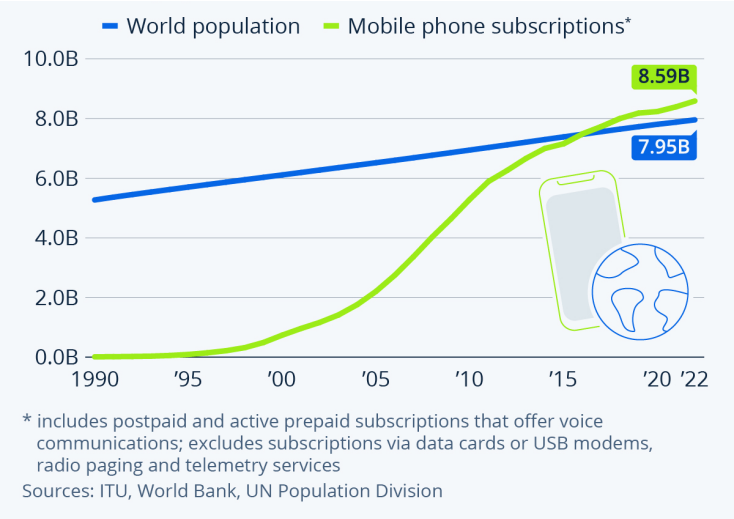


Figure 1.1: Graph representing the growth of global mobile subscriptions.

of mobile networks in the years to come.

With the increasing demand for fast and reliable mobile networks, more and more companies are turning to private mobile networks. Standardized by the Third Generation Partnership Project (3GPP) and referred as non-public networks (NPNs) [3], private networks are attracting attention in both the academic and industrial research communities. NPNs aim at providing the technologies developed for public networks, such as 5G, to private entities by restricting network access only to authorized terminals [3]. The main factors that drive new customers to choose an NPN are the simplified network management and the customization of their network resources. Private mobile networks are designed and deployed to optimize and redefine business processes that are either impractical or not possible within the limitations of wired and Wi-Fi networks. For industrial users, the ability to design mobile networks to meet the coverage, performance, and security requirements of production-critical applications is fundamental in the new era of the industry 4.0 [4].

Private networks offer several advantages to companies, but they also come with some disadvantages that must be considered. Due to the high amount of data generated by the subscribers or devices inside a network, there could be different types of anomalies that could compromise network functionalities. Anomalies can be defined as patterns in data that do not conform to a well defined notion of normal behavior. They are generated by many different reasons such as malicious activities of the network subscribers, breakdown of a radio infrastructure or soft-

[3] https://www.3gpp.org/technologies/npn

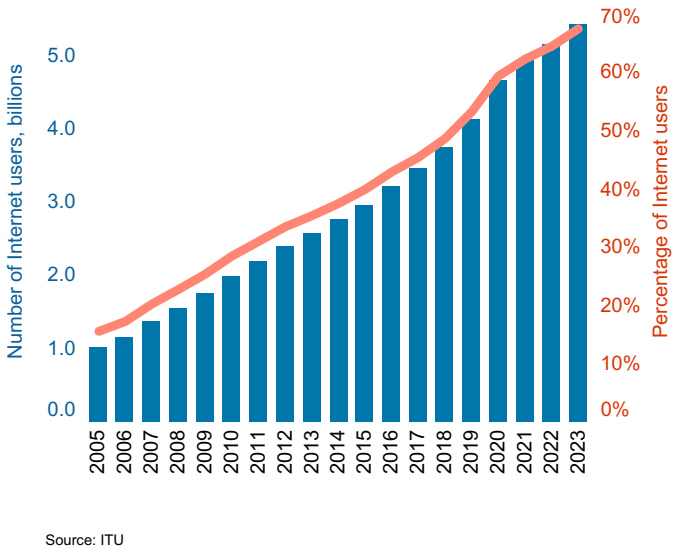


Figure 1.2: Trend of users that access the internet.

ware bugs. The term ”anomaly detection” or also called ”novelty detection” refers to the process of detecting patterns in data that significantly deviate from the majority of data instances. It has been an active research area for several decades and, due to the increasing demand and applications in broad domains, it plays an increasingly-important role in data mining, machine learning, computer vision and statistics [5]. With the exponential growth of data exchanged through a network, the manual identification of anomalous behaviors has become an infeasible task. But, thanks to the recent studies in the machine learning field, it is possible to automatically analyze and identify patterns that can highlight the presence of an anomaly. Anomaly detection algorithms are now used to enhance the traditional security measures and they can be generally applied in many domains, such as [6]:

- Intrusion Detection : Potential intrusion attempts and exploits at network and host levels. These are called respectively Network-based intrusion detection systems (NIDS) and Host-based intrusion detection systems (HIDS).
- Fraud Detection : Log data is analyzed to detect misuses and suspicious events, indicating frauds in financial transactions. In this project's scenario for example, there could be frauds related to the charging procedure of a subscriber.
- Medical Applications : Anomaly detection can help doctors and nurses to identify some critical situations in patient's health conditions. For example, identify some anomalous

signal coming from body sensors and detect possibly life-threatening situations.

However, during the development of an anomaly detection system, the are some issues that need to be taken into account. Most of the times, anomaly detection tasks involve unlabeled data which makes the learning process of the models more difficult. Furthermore, in some cases, the boundary between normal and abnormal behavior is not very defined, so the machine learning algorithms could produce many false positives and false negatives. Developing an anomaly detection system is challenging because of these factors.
This work has been carried out during an internship at Athonet, a Hewlett Packard Enterprise acquisition. The scope of this project is to build a machine learning-based anomaly detection mechanism able to identify erroneous patterns in the charging data records (CDRs) produced in their private networks. CDRs are defined as a formatted collection of information about a chargeable telecommunication event and are generated for billing purposes, enabling the mobile service providers to charge users for their services. The presence of anomalies in these files may indicate frauds, malicious activities from a specific subscriber or even possible bugs of the underlying system. The first step is the acquisition and conversion of the encoded CDRs in a human-readable format. Then, all the parameters available in the records are stored in a database created referring to the 3GPP's official documentation. After a detailed feature analysis on the most relevant parameters, two different datasets have been taken into consideration for training four different machine learning models. The pipeline has been created using artificially-crafted CDRs but then the overall system has been evaluated on the data of a real client. This work has been carried on using CDRs coming from private 4G networks, due to the low amount of 5G data available at the moment. However we are confident that with some small adjustments is possible to make the same system work also with 5G CDRs.

1.1 About Athonet

Athonet [4] is an italian company with more than 18 years of expertise in delivering reliable, secure, and flexible private mobile networks solutions. It was founded by Gianluca Verin and Karim El Malki in 2005 after working for several years at Ericsson, one of the most prestigious telecommunication multinationals. And since then, Athonet and its team of experts have been helping companies, governments and organizations by providing fully virtualized mobile core networks and several related solutions. Nowdays Athonet counts more than one

[4] https://athonet.com

hundred employees spread across different locations around the world, with hundreds of mobile cores deployed across Europe, Asia, Africa, Australia and Americas. Athonet also looks for new projects and growth opportunities, promoting the activities of its R&D team.
Indeed, the company's primary focus is now directed towards the innovative 5G technologies, which brought a performance and structural revolution with respect to the previous generations. While the commonly used 4G Long-Term-Evolution (LTE) has been developed to provide reliable voice services and broadband data traffic for users, the fifth-generation (5G) mobile network technology is also designed around services. Because of this philosophy, 5G allows to expand the business not only by offering connectivity to telephone operators but also by offering value-added services to verticals, which could be particularly beneficial to the business of private networks.
Together with this new paradigm on mobile networks, 5G comes with a performance improvement never seen in previous generations, including :

- high connectivity and bandwidth;
- more reliability and security;
- reduced latency and energy consumptions.

All these factors are bringing companies to invest in private 5G networks. To respond to the needs of the market and a growing demand from companies, Athonet developed their 5G core network, a software that provides the functionality of a mobile network but with the usability and manageability of a wifi network . The 5G core is the "brain" of the mobile network that manages the PDU session and controls all the network functions that are presented in section 2.3. The main advantages of the 5G core network are the following:

- It supports network slicing (explained in section 2.3), deploying different possible services on shared architectures to meet the needs of different types of customers.
- Virtualization enables different deployment options:
  - Fully on cloud : All the components that constiture the 5G core are deployed in cloud using AWS (Amazon Web Services).
  - Hybrid : User plane is stored at the client's site and the control plane is on cloud.

  - On Premise : User plane and control plane are installed at the client's premises, but the management services are located on cloud.
  - Standalone : Everything is installed in the client's infrastructure.

- Monitoring, analytics and other services on top of the core network.

On top of these services, Athonet has also developed a dashboard[5] for better management of the private mobile networks deployed across the globe. It allows to handle multiple network functions and the relative provisioning of subscriber data, including credentials, secret keys, USIM data profiles and associated quality of service (QoS). Moreover they can keep track of all the users' operations. Athonet over the years made its way in the telecommunication field, becoming a point of reference in the Italian and international market. It works in partnership with important brands such as Nokia, Ericsson, AWS, Vodafone, Samsung and many others. By working with the excellence in the telecommunication and information technology fields, Athonet offers a quality service to the many customers that are looking for their products. And thanks to its commitment towards constant innovation through the newest technologies, Athonet is becoming a leader in the sector, earning projects for major companies and government agencies. One of its most recent projects, was the deployment of private 4G and 5G in the airport of Paris, to guarantee a more secure,fast and reliable connectivity for all the services in the airport area. Its affirmed success didn't go unnoticed, in fact it has been recently acquired by the American multinational corporation, Hewlett Packard Enterprise (HPE). This is an important milestone for the company, because now it has access to more resources that can be invested in new technologies and to the new customers that will choose Athonet in the expanding market of private networks.

1.2 Thesis Outline

This document is organized as follows. Chapter 2 presents a technical description of the technologies involved in the project, including 4G and 5G technologies, as well as the structure of the CDRs generated during the charging procedures. In chapter 3 is presented the pipeline that comprehends the following steps:

[5] https://athonet.com/products/athonet-dashboard

- Acquisition and automatic conversion of the encoded CDRs via python script.
- Creation of a customized database with PostgreSql that is complaint with the 3GPP documentation.
- Python script for automatizing the insertion of the converted CDR's parameters inside the database.

The content of section 4 is focused on machine learning, more specifically all the techniques adopted to extract meaningful datasets from the database and a complete overview of all the models involved for the anomaly detection task. The results achieved by each model and dataset are presented in chapter 5, with also some performance comparisons between the algorithms. Finally, with chapter 6, the project is concluded by presenting some considerations about the achieved results and some ideas for future developments. The latter include insights to lower the time complexity of the processes in the pipeline and the changes to put in place for the transition from 4G to 5G of the anomaly detection system.

2

Theoretical Background

2.1 4G Technology

Inside a 4G network there are several entities that work jointly to ensure a reliable and fast connection to various mobile services such as internet access, messaging and calls. The overall architecture of a mobile 4G network is represented in figure 1.2, where it is possible to identify the following entities:

- **User Equipment** : It is any device used directly by an end-user to perform different actions through the network. Each physical equipment is uniquely identified by an IMEI or IMEISV code. The IMEI is a 15 digits code that consists of a TAC (Type Approval Code), a FAC (Final Assembly Code), SNR (Serial Number) and a Spare Digit, while the IMEISV also includes the software version of the handset. On the other hand, the subscribers are identified by the IMSI code, like in the third and second generations. The IMSI code is generated by concatenating the country code (MCC), the mobile network code (MNC) and the mobile subscriber identification number (MSIN). The official 3GPP documentation states that exist different categories of UEs[1] and, for each one, defines the quantities of uplink, downlink and MIMO layers each class of handsets can reach.

[1] www.etsi.org TS 36.306 V13.5.0

- **E-UTRAN** : is the radio infrastructure (RAN) where user interfaces connect to access network. This part of the LTE network is composed by several eNodeB, which serve as the base stations or access points for wireless communication between the user equipment and the core network. Each eNodeB in the radio access network is identified by a code called global eNodeB ID, which is built from the MCC, MNC and the eNode ID.

- **EPC** : is the core network architecture responsible for routing, managing, and delivering data packets between the user equipment and external networks. The key components inside the EPC are: MME, HSS, S-GW, P-GW. The MME is responsible for the session management procedures in the EPC. In particular, the MME handles the authentication, mobility management, location update, bearer establishment and handover support for each UE. During the authentication of a subscriber, it exchanges authentication information between the user and the HSS, the database that contains all relevant details about a subscriber and all the authentication parameters. The mobility management function, allows the mobility of a subscriber between networks, and the location update keeps track of its position and status within the network. The last functionalities of the MME, the bearer establishment and the handover support are presented later in this chapter, respectively in the attachment and handover processes of a subscriber in a 4G network. The final main nodes of a 4G network are the serving gateway and the packet data network gateway. The S-GW component handles the IP data traffic of the user, forwarding and routing packets to and from the eNodeB and the P-GW. The latter acts as the interface between the LTE network and other packet data networks, such as the Internet, IMS networks or LANs. The data flow through the P-GW is regulated by the PCRF (Policy and Charging Rules Function), which is an entity of the core network that formulate rules regarding the communication between UE and external networks. Among the different policies for the quality of the service, it also specifies the charging rules used for billing purposes.

4G technology has not only brought important advances in performance and architecture, but also introduced the security features needed to mitigate all the attacks identified in the previous generation, together with new cryptographic algorithms and improved key structures [7]. An evolved packet system authentication and key agreement (EPS-AKA) was introduced to guarantee a robust authentication method before the packet exchange between the UEs and EPC [8]. After the UE has been authenticated, two standardized algorithms are provided to

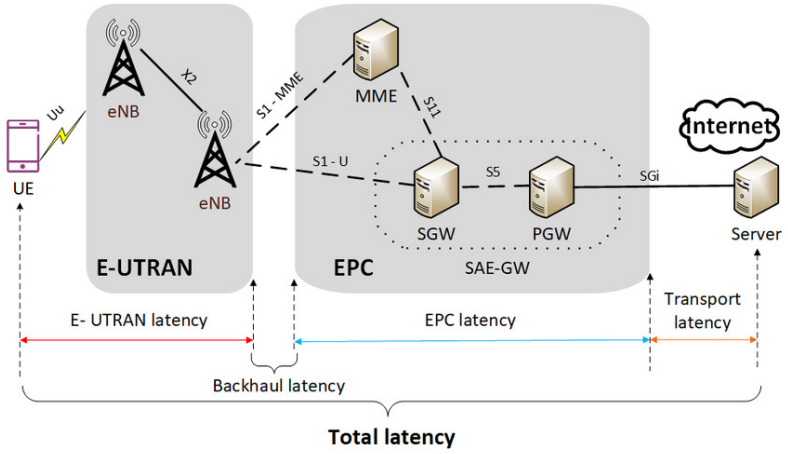


Figure 2.1: LTE Architecture

ensure data integrity and confidentiality protection via air interface named as EEA (EPS Encryption Algorithm) and EIA (EPS Integrity Algorithm). EEA and EIA use different cryptographic algorithms, depending on a 4-bit identifier:

4-bit code	version	algorithm	cipher type	key length
0000	EEA0	NULL	block	128 bit
0001	EEA1 - EIA1	SNOW 3G	stream	128 bit
0010	EEA2 - EIA2	AES	block	128 bit
0011	EEA3 - EIA3	ZUC	stream	128 bit

Table 2.1: Different versions of EEA and EIA based on the 4-bit identifier

Each version has strengths and weaknesses in terms of complexity (both space and time) and security against different classes of attacks that derive from the choice of the cryptographic algorithm [9] [10]. Both EEA and EIA are used to ensure confidentiality and integrity protection in different protocols used in the 4G network, according to the 3GPP standard. For example, 3GPP specifies that the non access stratum (NAS, protocol of the MME responsible for authentication, registration, connection/session management) and the radio resource control (RRC, the radio interface that passes the messages coming from the UE to the NAS) messages in the control plane must be secured with confidentiality and integrity protection (i.e. they must rely on the EEA and EIA). Instead other areas of the network, for instance the user plane packets between UE and eNodeB, cannot be integrity protected [11]. Despite the various security mechanisms deployed in the 4G technology, there are still vulnerabilities and threats that

need to be addressed. 4G users are still vulnerable to many attacks coming from the internet like malwares [7]. The 4G infrastructure itself can be subject to threats such as DoS and DDoS attacks, radio signals spoofing, eavesdropping and jamming, vulnerabilities in the components' operating system and many others [11][12].

2.1.1 UE Attachment

When a new user attempts a connection to a 4G network and access data or voice services, a set of processes happen in the background, forming the initial attachment procedure (shown in figure 2.2). If the process is successful, the UE is provided default connectivity, along with any charging rules that are enforced by the 4G network [11]. Firstly, the UE enstablishes radio link synchronization with an eNodeB and generates an attach request, which the eNodeB forwards to the MME. Within the request, there is the GUTI code, which is composed by the M-TMSI and GUMMEI codes. The former identifies the UE within an MME, and the latter uniquely distinguishes an MME. In addition to the GUTI, the attach request contains tracking details and other UE information. The MME subsequently sends an update location request to the HSS, before exchanging with the UE the necessary information for authentication and key agreement (AKA) to authenticate the subscriber, and to secure the messages. Once this procedure is completed, a default EPS bearer is established to enable the IP connectivity to the PDN. An EPS bearer can be thought as a tunnel connecting the UE with the P-GW, enabling the user's device to access the desired APN. The MME sends a create session request to the S-GW, that, in turn, sends it to the P-GW. Together with the PCRF, the P-GW establishes the QoS profile (in terms of performances, policies and charging rules) for the EPS bearer. Finally the UE sends an attach complete message to the MME to complete the attachment [13].

2.1.2 UE Handover

The term handover denotes the process of shifting an ongoing communication session from one cell or base station to another, ensuring a seamless and uninterrupted connection for a mobile device as it moves within the network. Handovers in 4G mobile networks are not only carried out in order to maintain a connection, but also to provide users better services and to meet individual requirements established in the QoS profile [14]. In a 4G network there are different types of handovers:

- **Intra-LTE handover**: it occurs between cells that are within the same LTE network.

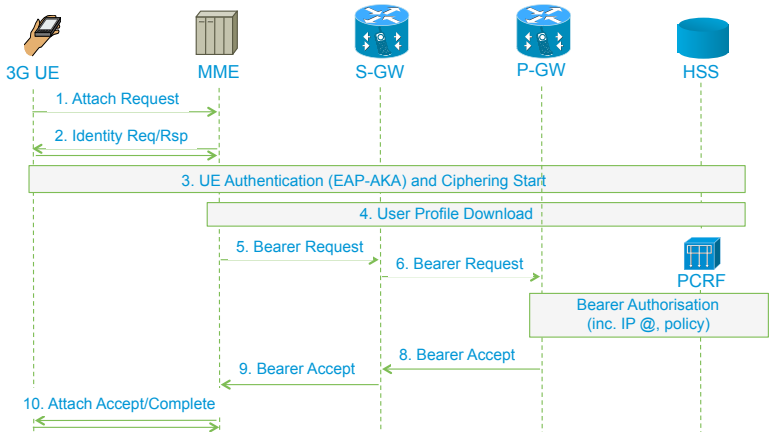


Figure 2.2: User attachment procedure. Source : Cisco -"4G Mobile Networks Long Term Evolution (LTE)" - Vladimir Settey

- **Inter-LTE handover**: it involves the transfer of an ongoing communication session between cells that operate on different LTE networks.

- **Inter-RAT handover**: it refers to the transition between different radio access technologies, such as moving from an LTE network to a 3G (UMTS) or 2G (GSM) network.

The handover is a complex procedure composed by different steps grouped into three main stages: handover preparation, execution, and completion. In the handover preparation phase, the source eNodeB will make an handover decision and send an handover request to the target eNodeB after receiving the measurement report from the UE. 3GPP defines several sets of parameters that the UE can include in the measurement report, so that eNodeB can evaluate whether to carry out the handover procedure. If the target cell accepts the request, a handover acknowledgement message will be sent to the source eNodeB through the inter-base station interface (i.e. the X2 interface shown in figure [2.1]). The message contains the configuration information of the target cell, including also the handover command. In the handover execution phase, the source eNodeB sends the handover command to the UE. The UE disconnects with the source cell immediately after receiving the handover command and starts to establish downlink synchronization with the target cell. The UE then initiates a random access process to the target cell using the random access resources configured in the handover command and reports the handover completion message after the procedure ends [15]. This is a simplified explanation of the mechanism, which is actually accomplished with much more complex procedures. The european telecommunications standards institute (ETSI) provides a more detailed

explanation, with a graphic representation of all the steps involved[2].

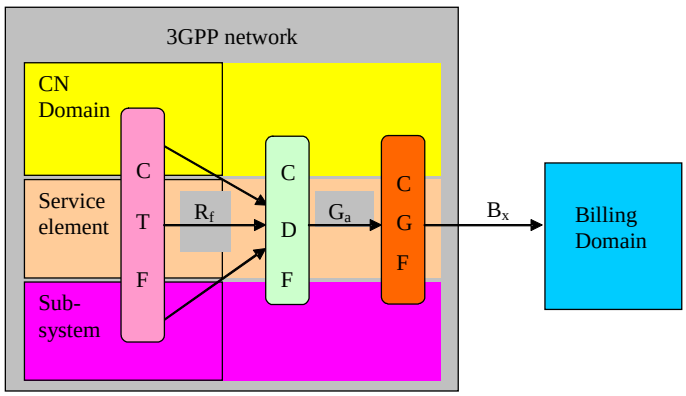
2.2 Charging Data Records

As operators invest in new infrastructure to provide more reliable services for their customers, the revenue-generating options become a key factor for the business cycle. From the EPS point of view, the system needs to enable the collection of information relating to different uses for individual users, so the operator can determine their own variant of billing as well as packaging towards end-users [16]. A charging data record is a formatted collection of information about a chargeable event (e.g. time of call set-up, duration of the call, amount of data transferred, etc.) for use in billing and accounting. 3GPP provides functions that implement offline and/or online charging mechanisms[3].

- **Offline charging**: it is a mechanism in which charging information does not impact the real-time service. The charging information for network resource usage is collected concurrently with that resource usage. The charging information is then passed through a chain of logical charging functions and, at the end of this process, CDR files are generated by the network. Finally, they are transferred to the network operator's billing domain (BD) for the purpose of subscriber billing and/or inter-operator accounting.
- **Online charging**: it is a mechanism where charging information can affect, in real time, the rendered service and therefore a direct interaction of the charging mechanism with the control of network resource usage is required. Online charging is a process where charging information are collected in the same fashion as in offline charging. However, the network needs to obtain in advance the authorization for the utilization of the network resource. The resource usage authorization may be limited in its scope (e.g. volume of data or duration), therefore the authorization may have to be renewed from time to time as long as the user's network resource usage persists.

3GPP defines different types of CDRs, based on their originating nodes and for the service that they are charging. But the thing that all these CDRs have in common is the procedure by which they are generated. It is composed by three phases that are shown in figure 2.3. The first is the charging triggering function (CTF), a mandatory component in all network elements that

[2] https://www.etsi.org - TS 36.300 V13.7.0 - p. 98
[3] https://www.etsi.org - TS 32.240 V17.10.0



CTF: **C**harging **T**rigger **F**unction
CDF: **C**harging **D**ata **F**unction
CGF: **C**harging **G**ateway **F**unction
BD: **B**illing **D**omain. This may also be a billing system/ billing mediation device.

Figure 2.3: Offline Charging Architecture Source : ETSI - TS 32.240 V. 17.10.0

provide offline charging functionalities. Its role is to collect information related to chargeable events within the network element and create matching charging events that it sends towards the charging data function (CDF). The latter uses the information contained in the charging events to construct CDRs. The format and the content of the CDRs depend on the interface that triggered the charging event (i.e. CN domain, service, subsystem shown in figure 2.3). The CDR generated by the CDF is then transferred to the charging gateway function (CGF) which acts as a gateway between the network and the BD.

Apart from the CDR's parameters, that depend on the type of network node that created it and the mobile operator, the structure of a CDR consists of:

- CDR header: it contains information related to the amount of data contained, the timestamp of generation and other parameters that can be seen in figure 2.4.
- List of CDRs : it is the sequence of records collected over a period of time (which is designated by the mobile operator). Each record contains a set of parameters about the charging event depending on its type.

2.2.1 P-GW CDR

For this project we take into consideration only the P-GW CDRs which, as the name suggests, are the records generated by the P-GW node in the EPC of a 4G network. All PGW-CDRs

Bits								
Octets	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1..4	File length							
5..8	Header length							
9	High Release Identifier			High Version Identifier				
10	Low Release Identifier			Low Version Identifier				
11..14	File opening timestamp							
15..18	Timestamp when last CDR was appended to file							
19..22	Number of CDRs in file							
23..26	File sequence number							
27	File Closure Trigger Reason							
28..47	IP Address of Node that generated file							
48	Lost CDR indicator							
49..50	Length of CDR routeing filter							
51..xy	CDR routeing filter							
xy+1..xy+2	Length of Private Extension							
xy+3..n	Private Extension							
n+1	High Release Identifier extension							
n+2	Low Release Identifier extension							

Figure 2.4: Table of CDR header fields : ETSI - TS 32.297 V. 17.2.0

are encoded using the ASN.1 format and are sent to the charging gateway function (CGF) using the GPRS Tunneling Protocol Prime (GTPP) as defined in the 3GPP standards. The ASN.1 (Abstract Syntax Notation One) encoding is a standard and flexible method for representing structured data, often used in telecommunications and network protocols. It defines a set of rules for encoding data structures in a way that can be understood by different computer systems, allowing for interoperability between diverse platforms and applications. The trigger event for a P-GW CDR generation is the opening or modification of an IP-CAN bearer, which is the connection that provides the transport and delivery of IP (Internet Protocol) traffic between the user's device and the internet or external networks. When there is an IP-CAN bearer activation, the CDF creates a brand new CDR and if there are some changes in the charging conditions of the session, the modified parameters are added in the same CDR by adding a new container inside the "list of service data" parameter. The triggers that lead to the CDR closure are the end of the IP-CAN bearer or the arising of at least one of the following conditions:

- Reaching of data volume limit;
- Reaching of time (duration) limit;
- maximum number of charging condition changes;
- management intervention;
- MS time zone change;

- PLMN change;
- radio access technology change (RAT Type);
- MO exception data counter receipt.

Table 2.2 shows the complete list of parameters inside the P-GW CDR, according to the 3GPP standard. A parameter can belong to one of four possible categories:

- **M**: The parameter is mandatory.
- **C**: The parameter has to be present if certain conditions are met.
- $\mathbf{O_m}$: If the operator provides the parameter, then it must be included.
- $\mathbf{O_c}$: If the operator provides the parameter and certain conditions are met, then it has to be included.

PARAMETER	CATEGORY	DESCRIPTION
Record Type	M	P-GW IP-CAN bearer record.
Retransmission	O_C	If present, indicates that information from retransmitted Charging Data Request has been used in this CDR.
Served IMSI	C	IMSI of the served party, if available.
IMSI Unauth. Flag	O_C	Indicates that the IMSI is not authenticated (emergency bearer service situation)
Served IMEI	O_C	The IMEI or IMEISV of the ME, if available. It is used for identifying the user in case Served IMSI is not present during emergency bearer service.
Served 3GPP2 MEID	O_C	MEID of the served party's terminal equipment for 3GPP2 access.
Served MN NAI	O_C	Mobile Node Identifier in NAI format (based on IMSI), if available.
P-GW Address used	M	The control plane IP address of the P-GW used.
P-GW Address IPv6	O_C	The control plane IPv6 address

Charging ID	M	IP-CAN bearer Charging identifier used to identify this IP-CAN bearer in different records created by PCNs. When charging per IP-CAN session is active, this field holds the Charging Id of the EPS default bearer.
IP-CAN Session Indicator	O_C	Indicates whether charging per IP-CAN session is active. If this field is not present, charging per IP-CAN session is not active.
PDN Connection Charging Id	O_M	Holds the explicit Charging Id for the PDN connection. This field is used to identify different records belonging to same PDN connection.
Serving Node Address	M	List of control plane IP addresses.
Serving Node IPv6 Address	O_C	List of control plane IPv6 addresses, in case of IPv4v6 dual stack, used during this record.
Serving node Type	M	List of serving node types in control plane (SGSN, SGW, etc). Node types map the addresses in Serving node Address field
PGW PLMN Identifier	O_C	PLMN identifier (MCC MNC) of the P-GW
APN Network Identifier	O_M	The logical name of the connected AP to the external PDN
PDP/PDN Type	O_M	Packet Data Protocol and Packet Data network type
PDP/PDN Type extension	O_C	Indicates Non-IP PDN type.
SCS/AS Address	O_C	Address of SCS/AS for tunnelling
Served PDP/PDN Address	O_C	IP address allocated for the PDP context PDN connection.
Served PDP/PDN Address prefix length	O_C	PDP/PDN Address prefix length of an IPv6 typed Served PDP address. The field needs not available for prefix length of 64 bits.
Served PDP/PDN Address extension	O_C	Holds IPv4 address of the served IMSI, if available, when PDP/PDN type is IPv4v6.

Dynamic Address Flag	O_c	This field is missing if IPv4 address is static when PDN Type is IPv4, or if IPv6 address is static when PDN Type is IPv6 or IPv4v6.
Dynamic Address Flag extension	O_C	This field is missing if PDP / PDN IPv4 address is static.
List of Traffic Data Volumes	O_c	A list of changes in charging conditions for all bearers within the IPCAN session. Each change is time stamped. Charging conditions are used to categorize traffic volumes, such as per tariff period. Initial and subsequently changed QoS and corresponding data values are also listed. Applicable only for IP-CAN bearer charging when IP-CAN session charging is active.
Record Opening Time	M	Time stamp when IP-CAN bearer is activated in this P-GW or record opening time on subsequent partial records.
Last MS Time Zone	O_C	This field contains the UE Time Zone the UE is located at IP-CAN bearer deactivation, or at default bearer deactivation when IP-CAN session charging is active, when available.
Cause for Record Closing	M	The cause for the release of record from this P-GW.
MS Time Zone	O_C	It contains, if available, the time zone where the Mobile Station is currently located.
Duration	M	Duration in seconds of this record in the P-GW.

List of Service Data	O_M	A list of changes in charging conditions for all service data flows within this IP-CAN bearer, categorized per rating group or per combination of the rating group. Each change is time stamped. Charging conditions are used to categorize traffic volumes, elapsed time and number of events, such as per tariff period. Changed QoS and corresponding data values are also listed. Online charging information (PS Furnish Charging Information) may be added per each service data flow container when it is sent by the OCS. External charging identifiers may be added per each service data flow container in case sent by the PCRF.
Diagnostics	O_M	A more detailed cause for the release of the IP-CAN bearer, when a single cause is applicable.
Enhanced Diagnostics	O_C	A more detailed reason for the release of the IP-CAN bearer, when a set of causes are applicable.
Record Sequence Number	C	Partial record sequence number, only present in case of partial records.
Node ID	O_M	Name of the recording entity.
Local Record Sequence Number	O_M	Consecutive record number created by this node. The number is allocated sequentially including all CDR types.
APN Selection Mode	O_M	An index indicating how the APN was selected.
Served MSISDN	O_C	The mobile station international subscriber directory number, if available.
User Location Information	O_C	This field contains the user location information of the MS when the CDR is opened.

Last User Location Information	O_C	This field contains the user location information of the UE at IP-CAN bearer deactivation, or at default bearer deactivation when IP-CAN session charging is active, when available.
User Location Information Time	O_C	This field contains the time at which the user location information was acquired, if available.
User CSG information	O_C	It contains the User CSG Information of the UE, if available.
3GPP2 User Location information	O_C	It contains the user location information of the MS when the CDR is opened as defined for 3GPP2 access.
TWAN User Location Information	O_C	It contains the UE location in a Trusted WLAN Access Network (TWAN), if available.
UWAN User Location Information	O_C	It contains the UE location in an Untrusted Wireless Access Network (UWAN) which includes the UE local IP address and optionally UDP source port number.
Presence Reporting Area Information	O_C	It contains the Presence Reporting Area Identifier for the UE PDN connection.
Charging Characteristics	M	The Charging Characteristics applied to the IP-CAN bearer.
Charging Characteristics Selection Mode	O_M	It holds information about how Charging Characteristics were selected.
IMS Signalling Context	O_C	It is included if the IP-CAN default bearer IM-CN Subsystem Signalling Flag is set.
Serving node PLMN Identifier	O_M	This fields holds the PLMN Identifier (MCC and MNC) serving the UE during this record.

CN Operator Selection Entity	O_C	It indicates whether the Serving Network identified by the ”Serving Node PLMN Identifier” has been selected by the UE or by the network, if available.
PS Furnish Charging Information	O_C	Online charging session specific information.
CAMEL Information	O_C	The set of CAMEL (Customized Applications for Mobile network Enhanced Logic) information related to IP-CAN bearer, if available.
RAT Type	O_C	It indicates the Radio Access Technology (RAT) type currently used by the Mobile Station, when available.
Start Time	O_C	Time when User IP-CAN session starts, available in the CDR for the first bearer in an IP-CAN session.
Stop Time	O_C	Time when User IP-CAN session is terminated, available in the CDR for the last bearer in an IP-CAN session.
Low Priority Indicator	O_C	This field indicates if this IP-CAN session has a low priority, i.e. for Machine Type Communication.
QoS Information	O_C	This field indicates the APN-AMBR (Access Point Network- Aggregate Maximum Bit Rate) uplink and downlink information for the IP-CAN session.
NBIFOM Support	O_C	NBIFOM (Network-based IP flow mobility) was requested by the UE, supported and accepted by the network for the IP-CAN session.
NBIFOM Mode	O_C	NBIFOM mode selected for the PDN connection, when NBIFOM is supported for the IP-CAN session.

UNI PDU CP Only Flag	O_C	It indicates whether this PDN connection is applied i.e. transfer only using Control Plane NAS PDU.
Serving PLMN Rate Control	O_C	It holds the Serving PLMN Rate Control used by the MME during this record.
APN Rate Control	O_C	It holds the APN Rate Controls enforced in the P-GW during this record.
MO exception data counter	O_C	Data counter value with the timestamp indicating the time at which the counter value increased from 0 to 1
Record Extensions	O_C	A set of network operator/manufacturer specific extensions to the record. Conditioned upon the existence of an extension.
3GPP PS Data Off Status	O_C	3GPP PS Data Off Status.
List of RAN Secondary RAT Usage Reports	O_C	A list of containers with volumes reported from RAN for secondary RAT usage, each container is time stamped.
VoLTE Information	O_C	This field holds the caller and callee information of VoLTE (Voice over LTE).

Table 2.2: List of parameters in a P-GW CDR.

Even if many of the parameters in the CDR are considered optional and operator-dependent, all of them have been included in the CDR database described in section 3.2. Many of them were not considered as dataset features (see section 4.1) and in general do not provide direct details about a possible anomaly in the subscriber's activity. But if the anomaly detection system finds a possible anomaly among all the CDRs, it is crucial to have as many details as possible, in order to fully understand the structure of the malicious action. Furthermore, it is worth noting that subscribers may belong to different mobile operators, which can result in variations in the structure of the collected CDRs, in terms of included parameters. Therefore, it is important to take this into account in order for the database to be able to store and manage CDRs without losing potentially important information.

2.3 5G Technology Introduction

As introduced at the beginning of this document, the drivers for 5G technology can be summarized in a mixture of both technological requirements and demands coming from new businesses and industrial companies of different vertical sectors. The possibilities for both a significant cost saving and new revenue enablers has created a large interest in 5G, not only among the usual mobile service providers and users [17]. Therefore, 5G can play a significant role in general industry digitalization for the years to come. The architecture of the 5G technology may resemble the 4G structure presented in section 2.1, however there are new mechanisms and technologies. Figure 2.5 shows the 5G architecture and the new network functions that compose the 5G core network. The next generation RAN (NG-RAN) infrastructure results from a mix of very different requirements for both 4G and 5G such as cooperation and resource sharing with existing 4G networks, migration strategies from 4G to 5G, etc. [18]. It also takes care of establishing, maintaining and releasing the parts of the PDU sessions that cross the radio interface. The 5G core network is the heart of a mobile network. It manages the remaining parts of the PDU sessions and takes care of all processes not related to radio access. This includes functionalities for securing the session and forwarding user data by providing connectivity. Behind all these processes there are different network functions, each one with a specific role:

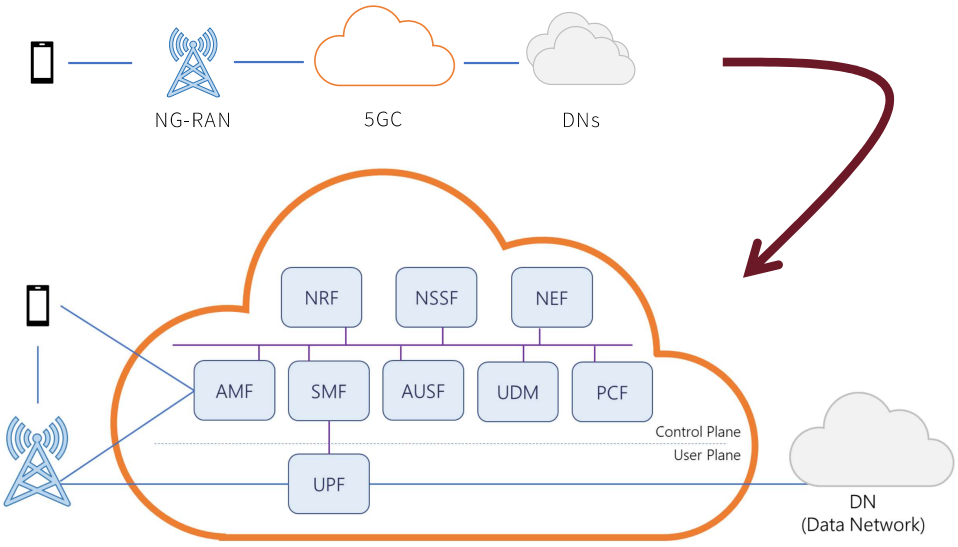


Figure 2.5: 5G network architecture.

- **access and mobility management function (AMF)**: It is the interface that communicates with the radio interface and the devices. It is involved in the signaling call flows in the 5G network. It supports encrypted signaling and authentication of the devices, so they can safely move between radio cells.

- **session management function (SMF)**: This unit manages the end user sessions, including the establishment, modification and release phases and the allocation of the IP address. This network function also controls the UPF, by configuring the traffic of the user session and the charging functionalities.

- **authentication server function (AUSF)**: It provides the device authentication service, using the credentials generated in the UDM. In addition, it generates cryptographic components for secure update of roaming information.

- **unified data management function(UDM)**: It acts as a front-end for the data saved in the Unified Data Repository (UDR), which stores the subscriber profile information, policy, structured and application data.

- **policy control function (PCF)**: It supports the unified policy framework that dictates the behavior of the network. It also provides and enforces policy rules to control plane functions.

- **network repository function (NRF)**: It acts as a registration center for all the core network components and provides different functionalities depending on the network function it is communicating with.

- **network slice selection function (NSSF)**: Function used by the AMF to assist in the selection of the network slice (see 2.3.2) instances that will serve a particular device.

- **network exposure function (NEF)**: It provides means for different network capabilities of application functions (AF), including location services, data collection and analytics, device triggering and QoS control.

- **user plane function (UPF)**: Its main role is to forward the user data, since its management is handled by the SMF. It connects to external IP networks and performs different processing on the data. It can apply packet inspection, enforce policies and QoS levels.

Together with all these new features, the release 15 of 3GPP introduced also a new charging system for 5G. The online and offline charging procedures of the 4G technology are merged into one converged charging system (CCS), which acts as a bridge between other network functions collecting charging data and the billing domain [19]. The component that communicates directly with the network functions is the charging function (CHF), which is supported by the following functions to gather all the information and transfer them to the billing domain.

- Rating Function (RF): It determines the value of the network resource usage.
- Account Balance Management Function (ABMF) : It contains the location of the subscriber's account balance.
- Charging Gateway Function (CGF) : Interface that communicates with the billing domain.

2.3.1 Virtualization

Traditional core networks (like for example the 4G core network described in section 2.1) are distributed applications that run on dedicated hardware, providing a non-optimal solution in terms of flexibility, scalability and adaptability. Moreover, the space and costs to accommodate this equipment as technology and services innovation accelerate, are significantly increasing. This is why the 5G technology moved towards the network virtualization. It involves the implementation of network functions and additional services in software that can run on a range of industry standard server hardware. Network functions can be placed in various locations of the network as required, without the need for installation of new equipment [20]. The goal is to decouple a functionality from the hardware it runs on by migrating them to virtualized resources, such as virtual machines and containers [17]. NFV (network function virtualization) has been evolving since the beginning of 5G, passing from the virtual machine based virtualization to the cloud deployments based on OS container virtualization. Cloud-native strategies allow service providers to develop and deploy new services in a faster and more organized way. Moreover, the introduction of containers helped the latter to optimize the hardware resources, by running different virtual OS instances in a single server [21].

2.3.2 Network Slicing

Another important advancement introduced in the 5G technology is network slicing. Customers coming from different industry segments may have different requirements in term of ultra-reliable services, ultra-high bandwidth or low latency. 5G has been designed to offer a different mix of capabilities to meet these different requirements at the same time. A network slice is a logical network that serves a well defined business purpose, utilizing shared physical network resources and capitalizing on cloud computing and virtualization technologies [22]. In this way the 5G network is able to adapt to the needs of the customers rather than the way around [23].

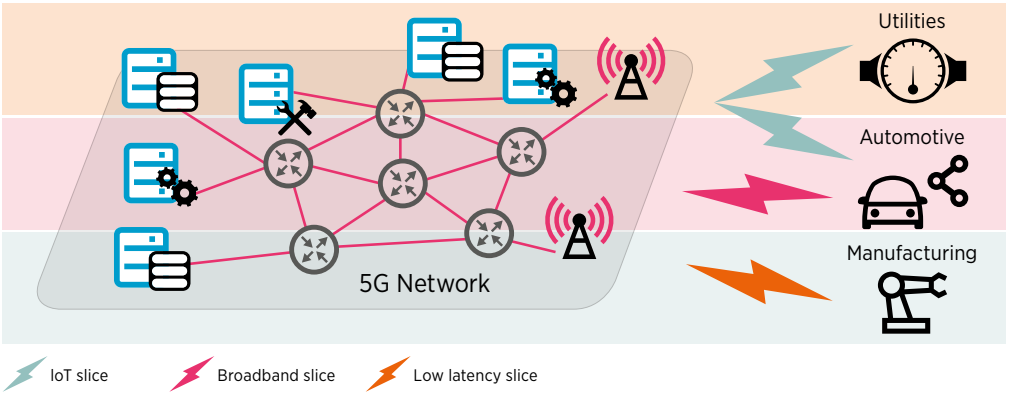


Figure 2.6: 5G network slicing representation. Source: GSMA White Paper - "An introduction to network slicing"

3

Pipeline

In this chapter is presented the pipeline that has been developed to collect, convert and store the P-GW CDRs (see figure 3.1). The encoded CDRs obtained from the 4G network are translated in a human readable format using a python script, together with a tool called CDR-Dump. This first process will create the json version of the CDRs that are stored inside a custom database. The latter has been created according to the list of parameters inside the P-GW CDR, shown in table 2.2. The CDR database is a central point for the overall anomaly detection system, because it allows to extract the features that will compose the datasets used for training the machine learning models.

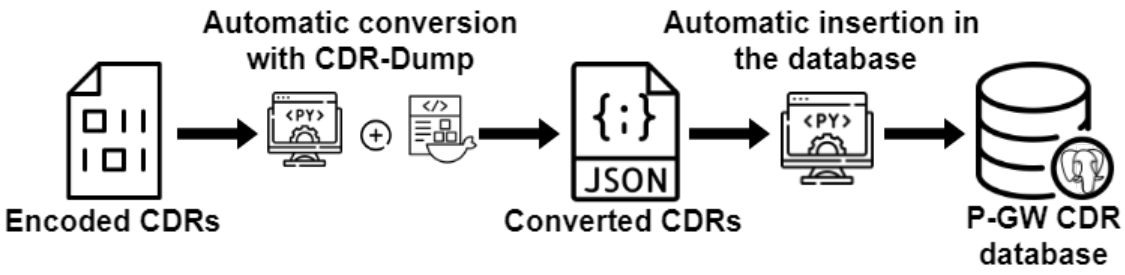


Figure 3.1: Graphical representation of the processes involved in the pipeline.

3.1 P-GW CDR Conversion

As already introduced in section 2.2, charging data records get generated in an encoded format based on ASN.1. This abstract syntax is a fundamental tool in telecommunication, since it allows to describe the complex data structures of protocol messages. In principle, there can be several different ways to encode the data values and usually gets decided at the beginning of the communication [24]. Among the different standards used to encode the ASN.1 data structures of the CDR, the most common one is the basic encondig rules (BER). Even if BER is not the most efficient encoding protocol like PER (Packed Encode Rules) and XER (XML Encode Rules), BER is often adopted because provides a very flexible framework for coding in an unambiguous way the ASN.1 values [25]. Before the transmission starts, the data is converted from the local representation of the sending system to an equivalent BER byte stream [26]. This sequence of bytes is what is found inside the encoded CDR file.
The CDR-Dump is an Erlang[1] program wrapped in a docker volume that performs the decoding process of the CDR binary file. Erlang is a programming language commonly used in the telecommunication field which allows the creation of distributed, reliable and concurrent systems. It was invented in the late '80s in the Ericsson's science laboratory, with the goal of creating a language to program the telecom systems in a much easier way. Moreover, Erlang comes with OTP (Open Telecom Project), an open source collection of libraries and tools specifically designed to enrich the capabilities of Erlang. The CDR-Dump algorithm performs the following operations:

- Opens the file path provided as command line argument and performs some checks on the binary file.
- Decodes the header of the encoded file, according to the 3GPP documentation [2].
- Extracts every single CDR contained in the file and decodes the BER byte sequence using the Erlang ASN.1 standard library [3]. The latter can be found inside the Erlang/OTP distribution.
- The decoded header and CDRs get formatted into a json file, returned as output.

[1] www.erlang.org/
[2] www.etsi.org TS 32.297
[3] www.erlang.org/doc/apps/asn1/

Algorithm 3.1 Pseudocode of the CDR-Dump tool

```
file ← open("/../CDRDIRECTORY")
check_binary_file (file)
header ← decode_header_payload (file)
cdr_payload ← get_CDRs (header)

//Use Erlang ASN.1 library
asn1_cdrs ← decode_ASN.1 (cdr_payload)
format_header (header)
format_cdrs (asn1_cdrs)
json_file ← convert_to_json (header, asn1_cdrs)
print_output (json_file)
```

For the anomaly detection task, is important to have a large quantity of data, because anomalies are present in a small percentage. This means that the mechanism needs to convert many CDRs. The CDR-Dump is capable of converting a single CDR at a time, therefore is important to make this process more fast. For this reason it has been implemented a python script that creates a subprocess which executes the command to run the CDR-Dump docker volume, feeding each time a different binary file.

Algorithm 3.2 Pseudocode of the Python script for automatic CDR conversion

```
path ← "/../CDRDIRECTORY"
for file_path in path :
    command ← call_CDRDump_command (file_path)
    subprocess.run (command)
end for
```

3.2 CDR Database Creation And Population

The CDR database has been created using a docker container with PostgreSQL, the most advanced open source ORDBMS (Object-Relational Database Management System) [27]. It was selected among the other DBMS options for its reliability and compatibility with Python, thanks to the Psycopg package. Psycopg[4] is a popular open-source PostgreSQL adapter for Python, that allows developers to connect with databases and handle data retrieval and manip-

[4]https://www.psycopg.org/docs/

ulation. It simplifies the process of working with databases and provides a convenient way to integrate Python applications with PostgreSQL. With this tool it is possible to easily automate the process of populating the database and also the extraction of parameters for the creation of the datasets.
The database has been created according to the list of parameters inside the P-GW CDR (see Table 2.2) as follows:

- The first step is the creation of the ER (Entity Relationship) model, that graphically represents the entities, their attributes, and the relationships between them. According to the 3GPP documentation, some of the parameters shown in table 2.2 are composed by several attributes, each of which has a different data type (sequence of characters, timestamp, boolean, integer, etc). For this reason, it is required to create a dedicated table for each of these parameters and link them to the "PGWRecord" table, which acts as the central point of the database. It contains all the parameters that don't need a dedicated table and the foreign key constraints to properly link to the other tables.

- The second step is the model translation into the corresponding SQL (Structured Query Language) code, defining all the tables and the parameter and key constraints.

- The final step is to populate the database by inserting the parameters of the CDR in json format. Since the number of json files is the same as the amount of encoded CDR provided by the network, is necessary to automate this process via a Python script (see the pseudocode 3.3). Some of the parameters inside the CDRs have a data type that is different from the one in the official documentation. For this reason some lookup table files have been created to correctly map the modified values.

Algorithm 3.3 Pseudocode for automatic insertion of parameters in the database

```
connection ← connect_Psycopg_DB("host","database_name"," username"," password")

// Get all the table names and corresponding parameters in a dictionary
db_tables ← get_db_tables(connection, query)
for table in db_tables :
    parameters ← get_all_parameters(connection, table)
    db_dict[table] ← parameters
end for

// Insertion Procedure
for JSON_CDR_FILE in JSON_CDR_DIRECTORY :
    // Extract CDR fields in a dictionary
    for CDR in JSON_CDR_FILE :
        cdr_dict ← extract_data(CDR)
    end for
    if cdr_dict  needs to be normalized:
        cdr_dict[field] ← normalize(field, lookup_table)
    end if

    // Fill a "foreign" table
    for cdr_dict.item() that is list or dictionary:
        sql ← prepare_sql_statement(db_dict[cdr_dict.key()], cdr_dict.value())
        insert_foreign_table(sql)
        // Append the foreign keys in a list
        foreign_keys.add(foreign_key)
    end for

    // Fill the PGWRecord table
    for cdr_dict.item()  that is NOT list or dictionary:
        pgwrecord_elements.add(cdr_dict.value())
    end for
    sql ← prepare_sql_statement_pgwrecord(pgwrecord_elements, foreign_keys)
    primary_key ← insert_pgw_record(sql)
    // Fill the linking tables
    insert_linking_table(primary_key, foreign_keys)
end for
connection.close()
```

4

Machine Learning

Machine learning is a research field that blends together artificial intelligence, computer science and statistics and its main goal is to create models that learn from some data source by optimizing a performance criterion. Machine learning uses the theory of statistics in building mathematical models able to make inferences from a sample. The role of computer science is twofold. First, store and process the dataset used to train efficient algorithms to solve an optimization problem. Second, once a model is learned, its representation and algorithmic solution for inference need to be efficient as well. Therefore, the efficiency of the learning or inference algorithm in terms of space and time complexity, may be as important as its predictive accuracy [28]. A machine learning problems can fall into one of the following categories, depending on the type of processed data:

- **Supervised Learning**. It is the most important methodology in machine learning and it also has a central importance in many classification problems [29]. These algorithms automate the decision making task by generalizing from known data. Therefore, its goal is to infer a function that maps the input into the output data. The type of the output depends on the problem: as an example, it can be a real value for regression and discrete for classification problems. The model during the training phase tries to learn this mapping by providing a dataset with labels, while in the test phase it is validated on new data not-seen before. In this way is possible to derive the performances of the model in terms of loss and errors [30].

- **Unsupervised Learning**. In this setting the machine learning model receives the input data but does not obtain the relative target labels. So in unsupervised learning the goal is to infer some underlying structure or patterns in data [31]. For example, in unsupervised clustering the goal is to infer a mapping from the given feature vectors to groups such that similar inputs are mapped to the same group.

- **Semi-Supervised Learning**. In this type of problem, the learning task is performed by models that use both labeled and unlabeled data. Semi-supervised algorithm are designed to improve other supervised and unsupervised tasks. For example, the introduction of unlabeled data in a classification problem or the introduction of labeled data in a clustering problem may lead to improved performances [32].

The datasets presented in this chapter are generated by extracting different features of the P-GW CDRs, and are not equipped with labels. By analyzing the pattern of the feature vectors is possible to identify the normal subscriber behaviour and the anomalies, represented by outlier points or by small clusters. However, defining performance measures in unsupervised models is more difficult than the more standard supervised learning. In fact, with supervised problems is possible to define different accuracy metrics starting from the amount of errors made during the test phase. With tasks that involve clean-data like this one, the measure of success is not always well defined. The strategy adopted to evaluate the correctness of the proposed models is the following. Each model is provided with a metric that gives a numerical evaluation of the likelihood that a given data point of the dataset is an anomaly. But, considering that the number of outliers is theoretically small, a manual inspection of the corresponding CDR can confirm the presence of an anomaly, hence proving the effectiveness of the model.

4.1 The Datasets

Once the database is created and filled with the parameters of several CDRs, the next step is to analyze and extract the most interesting ones, in order to obtain a dataset with meaningful information on the subscriber's activities. In fact, every machine learning model learns a task using the information contained in the dataset and, if the latter contains redundant or irrelevant data, we can expect poor results. Therefore, different pre-processing strategies are usually adopted to change the structure of the dataset, in a way that significantly improve the performances of the machine learning models:

- **Dimensionality reduction**. When machine learning algorithms are applied on high-dimensional data, they suffer from the curse of dimensionality, i.e., data are projected into a high-dimensional space becoming sparser and making the learning process more difficult. Moreover, with a large number of features, the models overfit, as well as increasing the memory and computational requirements for the training [33]. From the CDR database is possible to extract a large pool of parameters, which are often redundant or related other fields (hence they do not bring an additional information into the dataset). Therefore, the first strategy is to analyze the available features and only consider a small subset of independent and meaningful information. Nevertheless, some of the implemented algorithm still struggle especially when dealing with very big datasets. In these cases, to furher reduce the dataset's complexity, is necessary to resort to the PCA (Principal Component Analysis) feature extraction technique [34]. This algorithm simplifies the structure of the dataset but at the same time preserves the most important information. PCA calculates the principal components, which are a linear combination of the original variables. The components must meet two requirements, which are having high variance and to be orthogonal to each other. The observations of the original dataset are finally projected in the principal components, obtaining a new variable having a smaller dimension.

- **Feature normalization and scaling**. This is a process that involves the transformation of features in a common range so that greater numeric feature values cannot dominate the smaller numeric features values [35]. It is widely used in clustering algorithms, where the distance between data points plays a crucial role in determine the objective function. An example of why feature normalization is important for this work is the following. Consider the uplink and downlink parameters in the CDR, which are the amounts of bytes exchanged during the session. A normal user can easily create data traffic in the order of megabytes and gigabytes, so the expected value for these fields of the CDR is very high, compared with other parameters such as the session duration. So the strategy adopted is to use the min-max normalization [1], which linearly scales the features in a chosen range without reducing the meaningful traits that identify the anomalies.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

4.1.1 Subscribers Dataset

This dataset has been created based on the IMSI code inside each row of the CDR. The idea is to identify and group up all the charging sessions performed by a subscriber in a single dataset record. In this way, it is possible to have a general idea on the standard behavior of each device inside the network. This dataset contains:

- Number of charging sessions (i.e. charging IDs) registered in the CDRs: excessive charging session openings by a user can raise questions about their behavior.
- Total duration (in seconds) of all the charging sessions: while each has a variable duration, an excessively small or large total duration suggests the presence of sessions with suspicious values on average.
- Total downlink and uplink: if there are peaks in the data exchanged during the subscriber activity.
- Number of RAT types per subscriber: if the device of the subscriber performs operations using different radio technologies (e.g. EUTRAN, GERAN, UTRAN, etc.). Together with the number of sessions, this feature could indicate network connectivity problems.

When studying the features to be included in the dataset, other parameters have been taken into consideration, for example the time span in which the subscriber creates the charging events. At first glance it seems a valid feature, because it can show the subscribers that perform many operations in a small time interval. However, after implementing a first version of the dataset with this feature included, the results show that it only increases the number of false positives and does not highlight any type of anomaly. The same approach was also dedicated to the total number of IMEI per single user. Even if this feature could indicate the presence of anomalies (i.e. subscribers that have multiple IMEIs mean that they use different devices) this feature has been discarded for the lack of significant information. In fact, every subscriber had only one IMEI, hence the feature did not bring any additional information to the dataset.

4.1.2 Sessions Dataset

This dataset is created to analyze more in detail each charging session. Unlike the subscribers dataset, this one contains more records, since each subscriber can activate different charging sessions. This dataset contains:

- Duration of the session.
- Number of blocks the session was divided into.
- Total downlink and uplink.
- Number of changes detected during the session.
- Number of RAT type involved in the session.

The number of changes in the session is related to the ”list of service data” field inside the P-GW record. In fact, each time there is a modification of the session (see table 2.2) a new container in the service data list gets added. If there are too many elements in the list, it means that the session has been modified several times, which may indicate a suspicious behaviour. However, after analyzing the feature it shows that there are no records with a list with more than one element. So this parameter has been excluded since it brings redundancy into the dataset. The RAT type instead, indicates if the session has changed the radio technology during the session. Usually a change in the RAT type happens when there is a weak signal to the radio infrastructure of the network, so the device connects to another one with a better signal.

4.2 Machine Learning Models

In this section are presented the algorithms implemented for the anomaly detection of the P-GW CDRs. Each one of them has been implemented in Python and with the help of the Scikit-learn library, which provides the implementations for the models. Each one of them has been tested on both the datasets, to see which is the most suitable in terms of space and time complexity and overall performances. Moreover, it is interesting to cross-reference the results achieved by each algorithm, to see which are the most common anomalies, identify some false positives, etc.

4.2.1 Isolation Forest And Variations

The isolation forest (iForest) [36] is an anomaly detection technique that differs from the more classical distance and density based unsupervised algorithms. This method builds a set of binary trees structures called iTrees that isolate the anomalies from the normal data points by partitioning recursively the dataset instances. Each iTree is a data structure where a node can

either be a leaf (i.e. node without any child nodes) or internal (i.e. node with two child nodes). Each internal node consists in a split test, between a random attribute and split threshold. The result of the split test determines the two child nodes. Therefore, at the end of this recursive procedure, there are enough splits to separate each instance of the dataset (see figure 4.1).

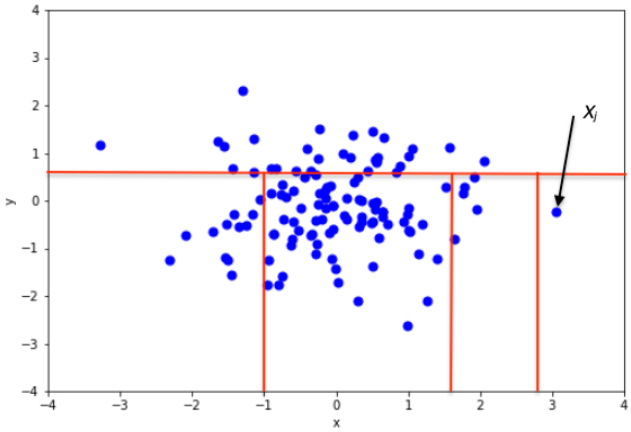
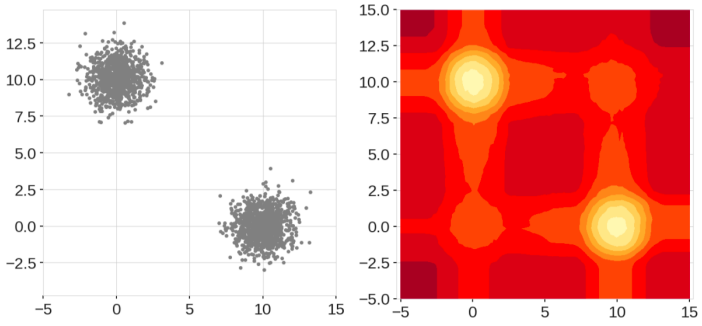


Figure 4.1: Representation of the split tests in a 2D dataset

The iForest then detects an anomaly based on the distance reached inside every iTree. The shorter the average path (i.e. the number of edges traversed to reach the leaf containing the point, starting from the root), the more likely the point is anomalous. Since this model is neither distance-based nor density-based, it does not perform computationally complex operations such as computing the distance from a centroid or other data points. Among the models proposed in this project, this is the only method that has a linar complexity , so the datasets with iForest are pre-processed by using only the feature scaling technique. Moreover an iTree when is fully developed it can have at most $2n - 1$ nodes, where n is the number of instances in the dataset, which means that the memory requirements for the algorithm are linear with respect to n. The iForest is divided in the training and test phases, in which the iTrees are generated from the training set and subsequently tested with the new data. For every sample in the test set the iForest returns the anomaly score, a value of confidence towards a sample. The higher the anomaly score, the higher the probability that a given sample is an anomaly. In this work, two different types of iForest have been implemented. The first one is the standard iForest just introduced, and the second one is the extended iForest (EIF) proposed in [37]. This model has been designed to overcome some of the problems that occurs in the standard iForest. In fact, the work presented in [37] shows how the anomaly score in the iForest is not always a reliable anomaly indicator. As shown in figure 4.2, the anomaly score is lower where the majority of the

points are located (the yellow areas in the heatmap), however there are also some artifacts introduced by the branching of the iTrees (represented in figure 4.2 as the light red and orange areas) that make the anomaly score inconsistent. The EIF implementation is similar to the standard iForest, but with some modifications that help the algorithm to compute a more robust and reliable anomaly score. Instead of making the branching parallel to the axes that define the feature space, the extended version introduces a random slope parameter (that in a n-dimensional space becomes a random vector of size n) that inclines the branch cut and the intercept parameter. The introduction of inclined hyperplanes for separating the dataset instances, help to reduce the artifacts created in the iForest algorithm, hence improving the anomaly score. Both the iForest and the EIF have been tested to see which model in the analysis of the CDR datasets is more reliable and achieves better performances.



(a) Two normally distributed clusters (b) Anomaly Score Map

Figure 4.2: Heatmap of the anomaly score that highlights the iForest problem.

4.2.2 DBSCAN

The DBSCAN (density-based spatial clustering of applications with noise) is an unsupervised algorithm that aims to find the structure of the data points, and cluster them based on common characteristics. As the name suggests, DBSCAN is a density-based model, meaning that clusters are identified as areas with high probability density and where instances are within a distance ε between each other. So the outliers, which can be seen as anomalies or noise points, lie in low-density regions, confirming the fact that are few and different with respect to normal instances [38]. The proper functioning of the algorithm is regulated by two hyperparameters:

- *epsilon*(ε) : it defines how close data points must be to each other to be considered a part of the cluster. Therefore this value is a threshold that tells if two points are considered

as neighbors.

- *min_samples* : it specifies the minimum number of instances in a close neighborhood to consider them as a cluster and a high-density region.

Unlike other clustering algotithms, DBSCAN does not require the number of clusters as hyperparameter. This is a very important property for all the unsupervised problems like the one presented in this project, where the distribution of the data and the number of classes are not known a priori. This is because the algorithm itself generates the clusters based on the values of the ε and *min_samples*. During the clustering procedure, DBSCAN defines the following types of data points:

- Core points: Points that have at least *min,amples* points in its ε neighborhood.
- Reachable points : Points that are directly reachable from the core point (i.e. the distance between them is less than ε) or it can be accessed from another reachable point.
- Outliers : All the remaining points that cannot be reached from the neighbors of a cluster.

The DBSCAN algorithm starts by choosing up a point in the dataset. If there are at least *min_samples* points within a radius of ε to the point then we consider all these points to be part of the same cluster. The clusters are then expanded by recursively repeating the neighborhood calculation for each neighboring point. This procedure continues until all the points in the feature space have been visited.

4.2.3 Local Outlier Factor (LOF)

The LOF (local outlier factor) is a density-based outlier detection algorithm which computes the local density deviation of a given data point with respect to its neighbors. To better understand how LOF works, it is necessary to introduce the following terminology [39]:

- *k - distance* : This is referred to the distance between a data point and its k nearest neighbors. In a two dimensional space, this term can be thought as the radius of a circle that includes k neighbor points.
- *k - neighbors* ($N_k(x)$) : Is the set of neighbours points that are located around point x, computed using the *k - distance*.

- Reachability distance (RD) : Is defined as the maximum value between the *k - distance* and the distance (which can be Euclidian, Manhattan, etc.. depending on the problem) between two points.

$$RD(x_i, x_j) = max(k - distance(x_j)\ ,\ d(x_i, x_j))$$

  So if the point x_j is already in the set of k neighbors of x_i, then $RD(x_i, x_j)$ will be the *k - distance*, otherwise it computes the real distance between x_i and x_j.

- Local reachability density (LRD) : Is defined as the inverse of the average rachability distance.

$$LRD(x_i) = \frac{1}{\sum_{x_j \in N_k(x_i)} \frac{RD(x_i, x_j)}{|N_k(x_i)|}}$$

  For outliers, the average distance will be high, so the LRD value will be low. On the contrary, a data point near a cluster will have a high LRD value, since the RD will most of the times be the *k - distance*.

With these values, is possible to calculate the local outlier factor (LOF) of a point:

$$LOF_k(x) = \frac{\sum_{x_j \in N_k(x)} LRD_k(x_j)}{|N_k(x)|} \cdot \frac{1}{LRD_k(x)}$$

If the LOF of a point is close or smaller than 1 it means that is close to its neighbors, hence it is a normal point that belongs to some cluster. If instead the LOF value is greater than 1, then the point is likely to be an outlier.

5 Results

5.1 Athonet's Client Data Processing

For testing the machine learning models presented in section 4.2, Athonet provided real data from one of its biggest clients. They provided 9 days of client activity, in the form of encoded P-GW charging data records. All the algorithms and models presented have been executed with a laptop with 16GB of RAM and a 3GHz Intel i7 of eleventh generation.
The first thing that has been tested is the time required for converting and storing one entire day of activity, using the pipeline described in chapter 3. For each day, the P-GW of the network generates 120 encoded CDR files (one every 15 minutes plus an additional file after three minutes of every hour) which collectively have size of 1.8GB on average. This depends on the size of the network and on the number of active subscribers that perform operations subject to charging. For this particular client, every file contains around 100 thousand CDRs for each time interval. These latter are given as input to the pipeline presented in chapter 3. The algorithm for automating the conversion of CDRs (see pseudocode 3.1,3.2) takes 1 hour and 54 minutes to convert the files related to an entire day, producing in output 8.91GB of json content. Then, every individual json (which weigh around 90MB) are provided as input for the database population algorithm (see pseudocode 3.3). For inserting a single file, which correspond to 100 thousand database records, the algorithm takes on average 6 hours and 12 minutes. After running the code for three days and 15 hours, the CDR database has been populated with 1.059.019

records, related to 3 hours of subscribers charging activities.
The next step is the creation of the two datasets presented in section 4.1. A python script has been implemented for extracting and storing locally the subscriber and session datasets. Since the Psycopg library allows to interact with the database, it is possible to create the views (i.e. virtual tables that contains data from one or multiple tables) needed for returning the dataset's parameters. With this method, the feature extraction is a very fast process, because after creating and joining the views using the key constraints stated during the database creation, it is simply a matter of creating a query that prints the result. The output of this operation are a session dataset of size 466.561×5 (where each row represents the details of a single charging session) and a subscriber dataset of size 391.332×5 (where each row represents the information of the subscriber's IMSI code). Finally the two datasets have been scaled using the min-max scaling technique described in section 4.1.

5.2 Models Performances

The first model that has been tested on both datasets was the DBSCAN. Unfortunately, the latter returned a memory error, probably caused by the following reasons:

- Complexity of the algorithm. The computational complexity of DBSCAN is $O(n^2)$, where n is the number of points in the dataset [40]. So the operation of finding the main clusters on large datasets is computationally demanding.
- Curse of dimesionality. The user and session datasets have 5 columns, which is already a number of features that puts a strain on the algorithm. To fix this issue, the PCA algorithm has been applied in addition to the scaling technique (reducing the initial five-dimensional dataset into a three-dimensional dataset), to reduce the size (and the complexity) of the two datasets, without any success.
- Limited resources. The hardware capabilities for testing this model are not enough. The scikit-learn implementation of the DBSCAN pre-computes the neighbors of every point, which increases the memory usage as the number of points increase.

Even a different choice of hyperparameters did not change the outcome of this algorithm. It has been executed with several ε and *min_samples* values, but DBSCAN still terminates with

a memory error.
The second model that has been tested is the Local Outlier Factor. It has been executed and analyzed using different values of the *n_neighbors* hyperparameter and the results for each dataset are shown respectively in table 5.1 and 5.2.

Number of neighbors	Execution Time [sec]	Detected Anomalies	Percentage
20	62,84	22.755	5.8%
100	86,46	29.665	7.6%
200	111,48	30.896	7.9%
500	206,80	37.457	9.6%

Table 5.1: LOF performances on subscriber dataset

Number of neighbors	Execution Time [sec]	Detected Anomalies	Percentage
20	154,08	26.594	5.7%
100	188,06	33.126	7.1%
200	223,82	35.459	7.6%
500	335,09	44.790	9.7%

Table 5.2: LOF performances on session dataset

From tables 5.1,5.2 is possible to see that, as the number of neighbors increases, the time required to execute the algorithm increases. Moreover, as the hyperparameter grows, the number of detected anomalies increase. This makes the model less accurate in detecting anomalies, because it introduces a greater amount of data points that cannot be reconducted to anomalies. Therefore, the results from the LOF model become less consistent as the number of neighbors increase. The hypothetical number of anomalous data points are present in a very high percentage with respect to the entire datasets (see the "Percentage" column of both tables). This fact goes against the definition of anomaly, which are considered different and, more importantly, very few. Hence the most accurate setting for this algorithm is when $n\_neighbors = 20$. In figure 5.1, 5.2 are shown the feature distribution of the anomalies, respectively in the session and subscriber datasets. With this configuration the LOF model still returns many false positives, however, as described in the next section of this chapter, it is also able to identify some anomalous patterns that the other models fail to detect.

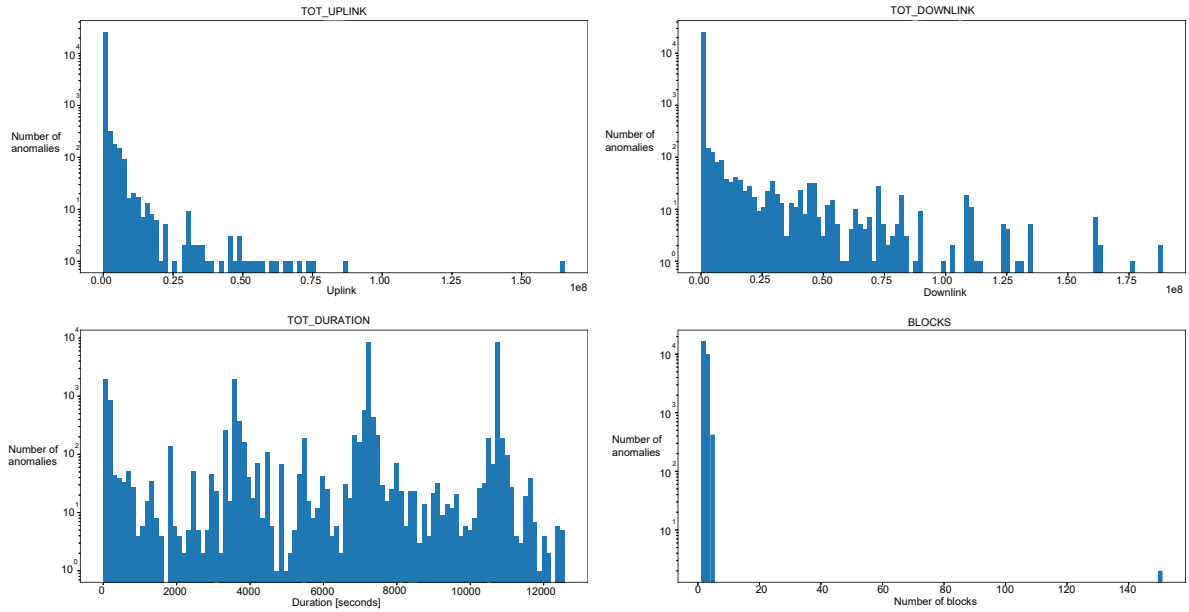


Figure 5.1: LOF with $n\_neighbors = 20$ on session dataset

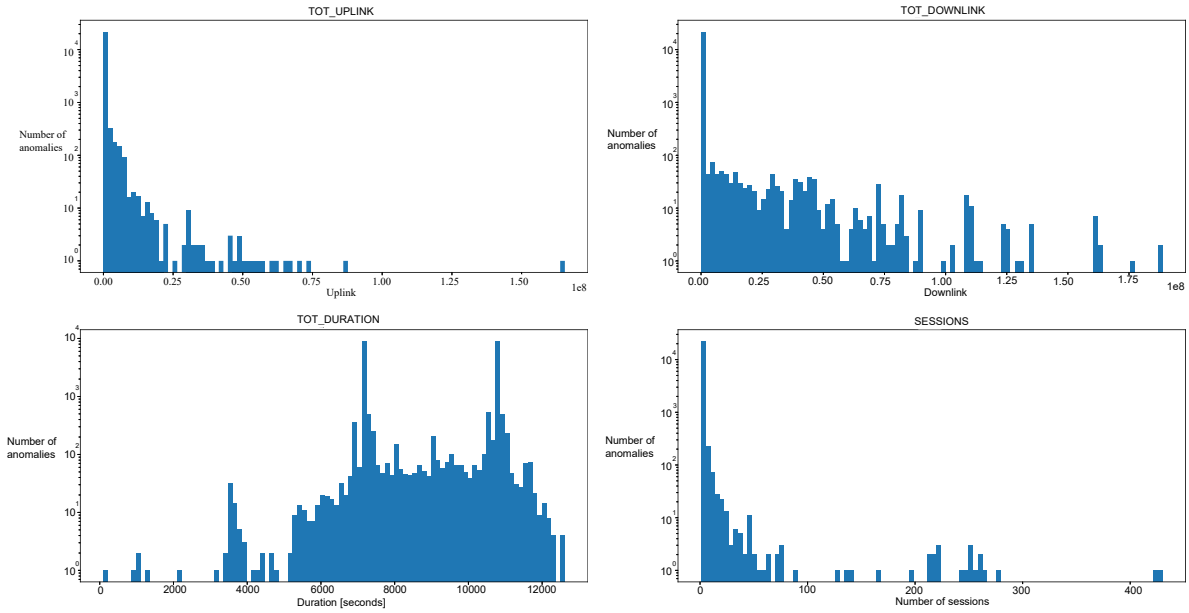


Figure 5.2: LOF with $n\_neighbors = 20$ on subscribers dataset

The next model that has been tested was the standard isolation forest. As the LOF algorithm, the IF was executed using different combinations of hyperparameters. Initially, togheter with

the number of iTrees and the number of samples used to train the latter, it was also included the contamination hyperparameter. According to the Sklearn documentation, the contamination parameter represents the proportion of outliers in the dataset. Since this data is unknown due to the nature of the problem, the model has been tested with different contamination values (0.0005,0.001, 0.005, 0.01), to see if there were significant differences in the model performances. However, results showed that the performances achieved by the model in these different settings are very similar. Therefore, the contamination value has been fixed to 0.001 for all the hyperparameter combinations.
In tables 5.3 and 5.4 are shown the results achieved by IF in both datasets.

Number of trees	Number of samples	Execution Time [sec]	Detected anomalies
256	1000	20,07	2894
500	1000	35,81	3493
1000	1000	68,99	3453
256	5000	58,98	777
500	5000	116	958
1000	5000	225,2	827
256	10.000	65,45	409
500	10.000	121,75	416
1000	10.000	242,95	398

Table 5.3: IF performances on user dataset

Number of trees	Number of samples	Execution Time [sec]	Detected anomalies
256	1000	18,63	4237
500	1000	35,45	4487
1000	1000	68,62	4424
256	5000	76,06	1563
500	5000	140,46	1647
1000	5000	278,4	1511
256	10.000	78,16	1020
500	10.000	147,77	1116
1000	10.000	302,16	1113

Table 5.4: IF performances on session dataset

The number of anomalies presented in the tables, represent only the ones that have an anomaly score that exceeds a certain limit. There is no strict rule for determine the anomaly score threshold, it is a matter to analyze the patterns and their corresponding score. Initially, this threshold was set to 0.6, but it a second moment was raised to 0.7. This because the data points with anomaly score between 0.6 and 0.7 do not present any anomalous value. This also helped in reducing the overall number of false positives among the detected anomalies. Results immediately show that isolation forest performs better than the LOF. Not only it executes in less time, but it is also more accurate. In this case it has been considered a tradeoff between execution time and detected anomalies to choose the best setting for IF with both datasets.

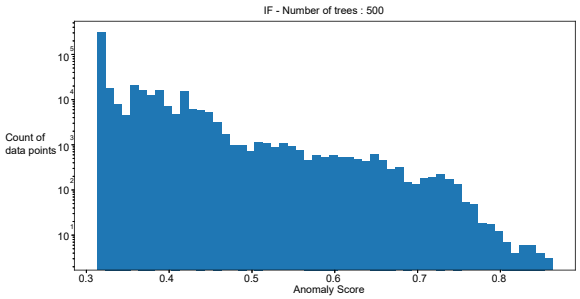


Figure 5.3: Anomaly scores of iForest on session dataset

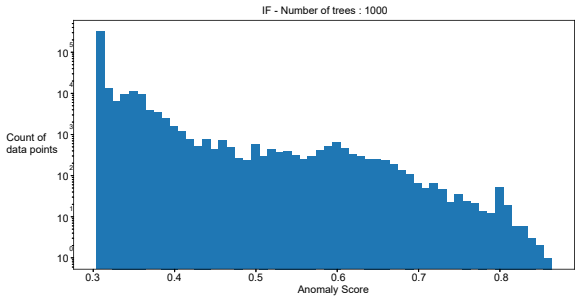


Figure 5.4: Anomaly scores of iForest on subscribers dataset

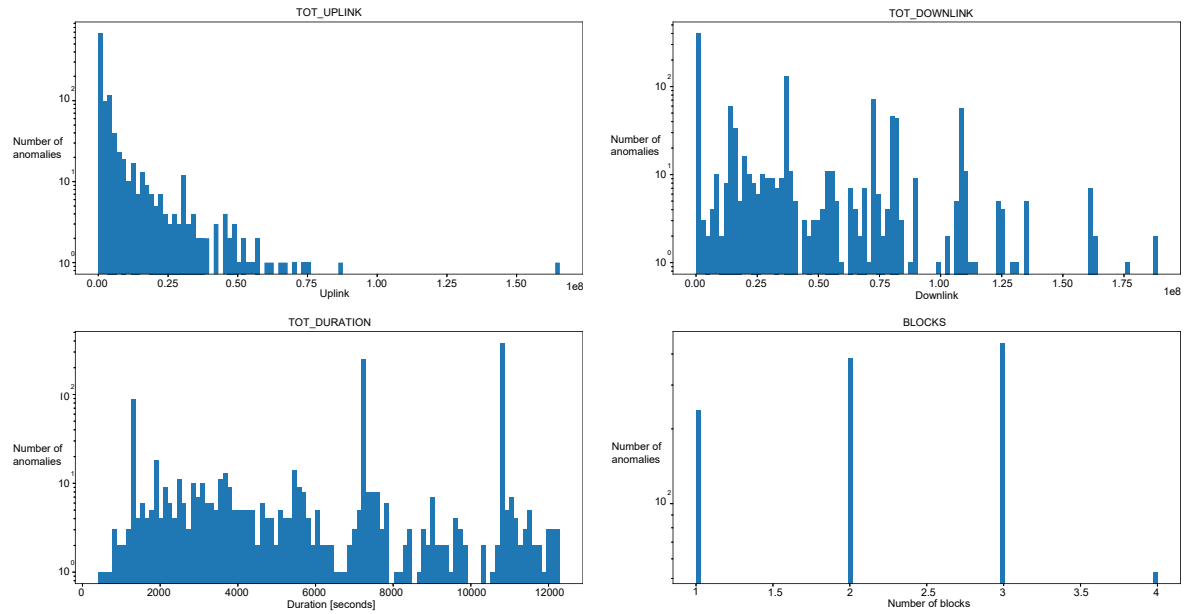


Figure 5.5: IF with $n\_trees = 500$ and $n\_samples = 10.000$ on session dataset

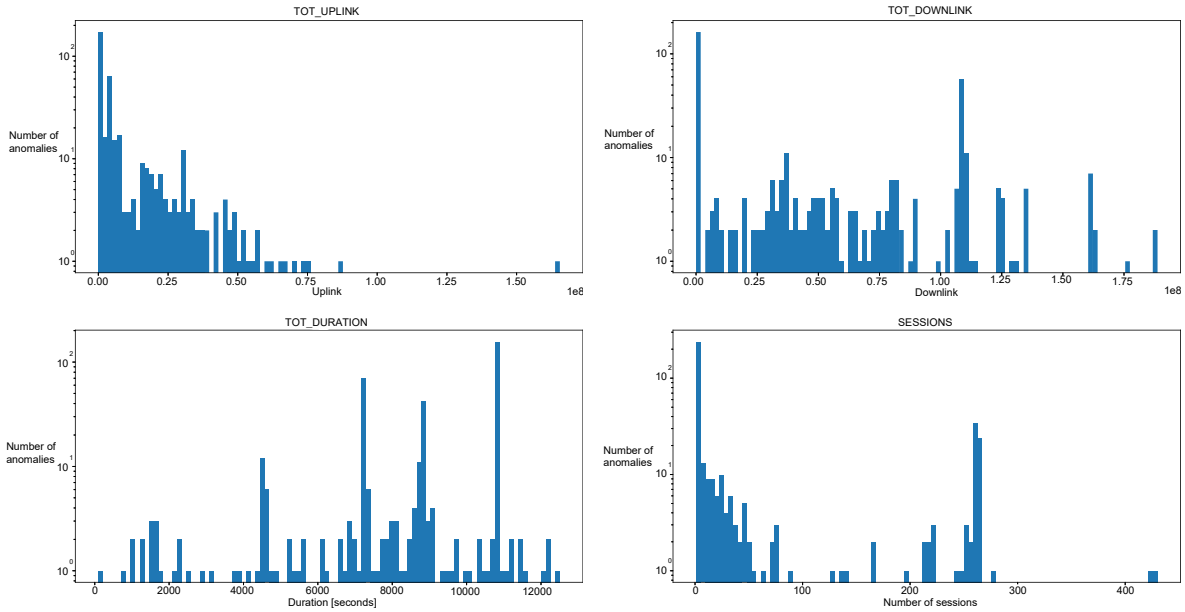


Figure 5.6: IF with $n\_trees = 1000$ and $n\_samples = 10.000$ on user dataset

Comparing the results achieved by LOF and IF (figure 5.1, 5.5 for session dataset and figure 5.2, 5.6 for subscriber dataset) is possible to see some common patterns in the detected anomalies. For example, there is a high number of charging sessions that exchange a limited amount of data packets (i.e. the spikes around the 0 value in the uplink and downlink graphs). However it is important to recall that the isolation forest retrieves several thousand fewer anomalies, confirming the fact that the LOF, togheter with possible anomalies, it returns also a lot of noise data.

Finally, the last model put to test is the extended version of the isolation forest. Like the standard IF, it has been tested using different combinations of hyperparameters (i.e. number of iTrees and number of samples) on both datasets. According to the original EIF paper [37], it is possible to choose the extension level of the n-dimensional hyperplanes. This value ranges between 0 and $n - 1$, where n is the dimension of the dataset. When set to 0, the EIF model coincides with the standard isolation forest, while the fully extended case is when the extension level is set to $n - 1$. This implementation of EIF has been studied in the fully extended case, therefore with an extension level equal to four.

Tables 5.5,5.6 show the model performances on both datasets.

Number of trees	Number of samples	Execution Time [sec]	Detected anomalies
256	1000	109,84	1578
500	1000	234,15	1384
1000	1000	783,34	1411
256	5000	464,67	1717
500	5000	1095,78	1535
1000	5000	2794,43	1612
256	10.000	529,53	702
500	10.000	1269,72	673
1000	10.000	3308,86	687

Table 5.5: EIF performances on user dataset

Number of trees	Number of samples	Execution Time [sec]	Detected anomalies
256	1000	126,64	2720
500	1000	290,09	2449
1000	1000	875,39	2569
256	5000	225,61	1293
500	5000	443,06	1213
1000	5000	1655,85	1253
256	10.000	327,45	1011
500	10.000	658,53	1045
1000	10.000	1900,46	1029

Table 5.6: EIF performances on session dataset

Note that the number of anomalies detected by EIF is related to the ones that obtained an anomaly score greater than 0.7. Results show that EIF, in many cases, detects a smaller subset of anomalies with respect to the standard IF. However, the EIF takes more time to execute with respect to the other models, probably because of the computation of the separation hyperplanes needed for training the isolation trees. As in the previous models, it has been considered the EIF with the best tradeoff between time complexity and performance in anomaly detection. The the anomalies values distributions are shown in figures 5.9,5.10 where is possible to notice many similarities with the standard IF.

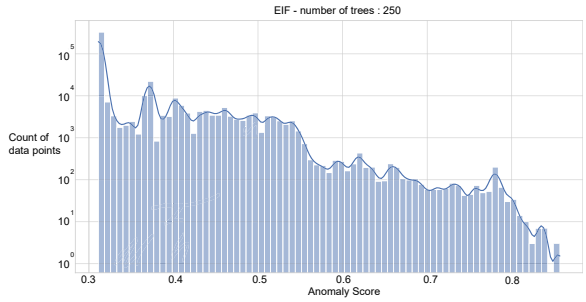


Figure 5.7: Anomaly scores of Extended Isolation Forest on session dataset

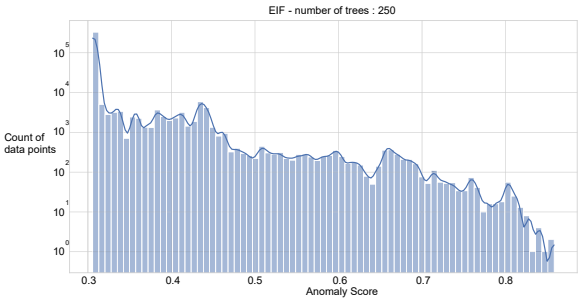


Figure 5.8: Anomaly scores of Extended Isolation Forest on user dataset

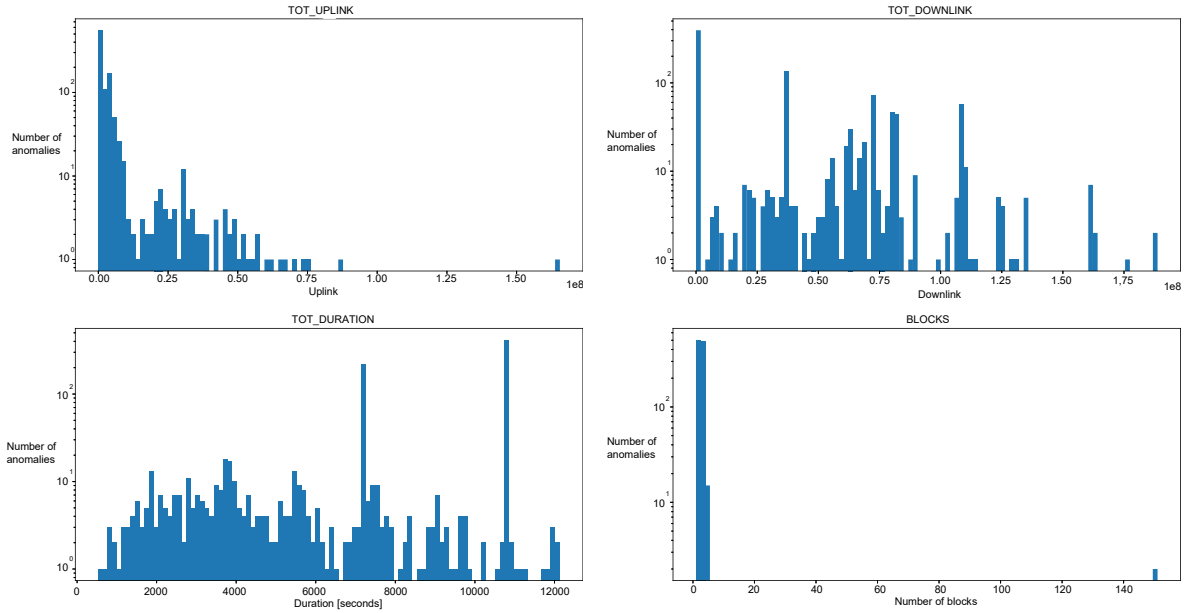


Figure 5.9: EIF with $n\_trees = 256$ and $n\_samples = 10.000$ on session dataset

5.3 Results

From the tables and the graphs shown above is possible to understand that, apart from the DBSCAN, each model returns a certain number of possible anomalies that need to be analyzed more in depth. The LOF model returns too many data points, therefore it cannot be considered as reliable as the IF and the EIF. The latter is considered more robust than IF, however it is more computationally demanding due to the additional computations introduced to improve the

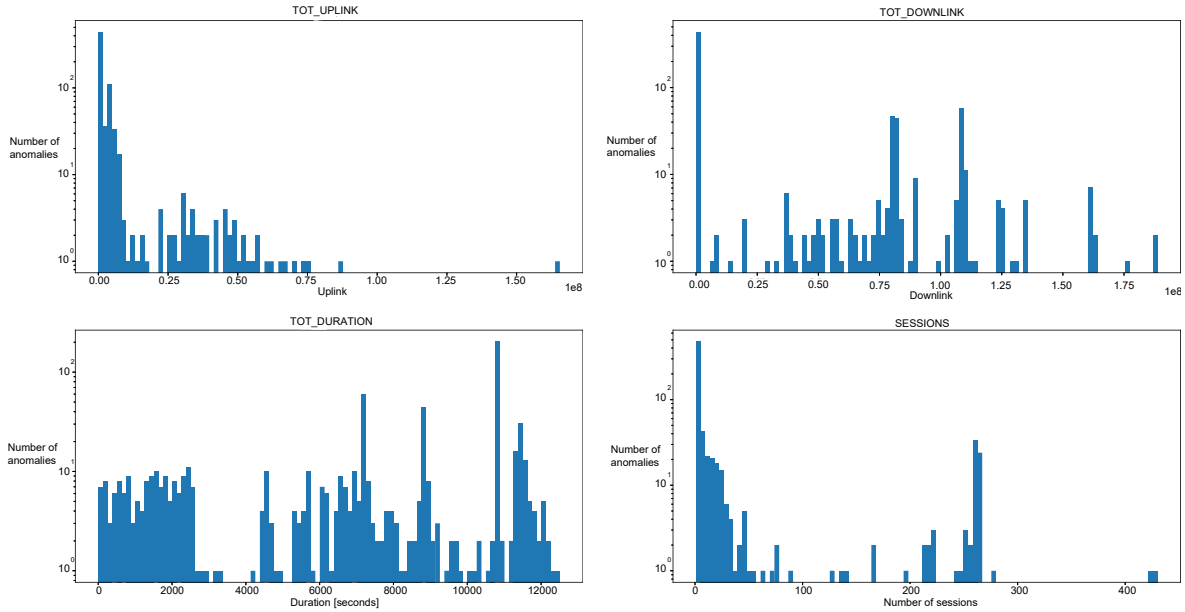


Figure 5.10: EIF with $n\_trees = 256$ and $n\_samples = 10.000$ on user dataset

standard IF. So the model that has the best trade-off between time complexity and reliability remains the standard isolation forest. The anomalies returned from the isolation forest (figure 5.5, 5.6) have been considered as a starting point for a more detailed analysis, to check if the model is capable to capture real anomalous behaviors. With the help of an Athonet's dashboard and an expert of the Athonet's dashboard team, it was possible to study some of the returned anomalies for each dataset. From the analysis emerged some interesting results:

- The anomalies returned from IF using the session dataset (i.e. finding anomalous charging events) cannot be considered as such. Unless there are some very strange feature values, it is very difficult to determine if an actual anomaly has been carried on from a single charging session. Even if many of these sessions do not exchange data packets, this cannot guarantee an anomalous activity. Moreover, the duration and RAT type features of the session dataset do not highlight any abnormal value. This has been confirmed also with the other models, with an exception in the LOF model. In the "blocks" graph in figure 5.1, is possible to notice that there is a small subset of charging sessions that are subdivided in a very high amount of time blocks. It has been discovered that these sessions, however, belong to a different access point network (APN) of the client. Since these sessions belong to another network, they have a different set of rules, i.e. the duration and volume limits change with respect to the main APN from where most of

the data has been taken. So, even if they are indeed anomalous with respect to the main APN CDRs, they still cannot be considered real anomalies.

- To identify real anomalies in the subscribers activities, the subscribers dataset is the most reliable. The most interesting results have been discovered using this dataset. By looking at the "number of sessions" graph (figure 5.6) it has been found that there are 108 IMSI (i.e. subscribers) that returned an anomalous value. In fact, these subscribers performed from 70 up to 430 charging sessions in a time span of around two and a half hours (that corresponds to the time range of the CDRs used for the experiments). By watching at their activity during the entire day with the dashboard, it has been confirmed that there is actually some anomalous activity. More specifically, there are two main categories of anomalous behaviors:

  – **Attachment and Detachment** : This type of anomaly is the most common one. The subscriber performs an attachment request, which initiates the registration of a new CDR. However, few seconds after the procedure completion, the session drops for some unknown reason. So the subscriber performs again an attachment request that gets granted but drops in a small time and so on. Some of these subscribers get stuck in this loop for many hours, and the result are a high amount of sessions (i.e. CDRs) that have a small duration and in which there is basically no data exchanged through the network (see figure 5.11).

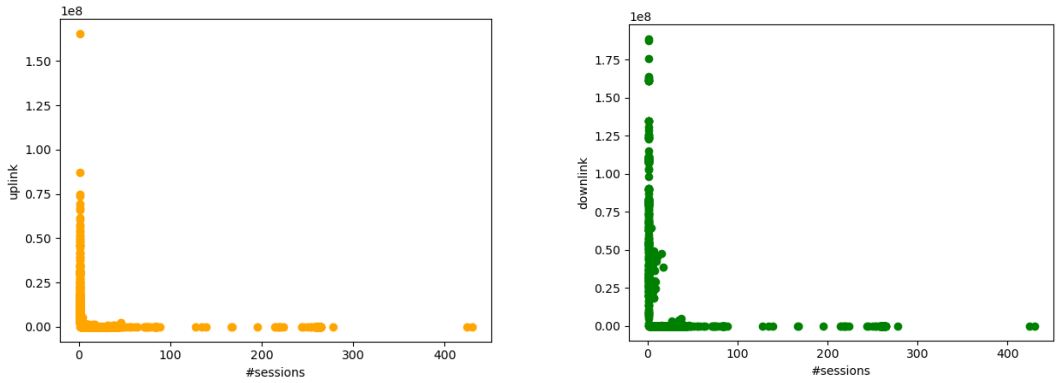


Figure 5.11: Relationship between number of sessions and uplink/downlink

This behaviour will be further investigated by Athonet's dashboard experts and the client that provided the CDRs.

– **RAT type change** : This anomaly has been encountered with only few of the subscribers with many charging sessions. These users attempt to connect to a 4G network (which can provide higher connectivity) with a bad signal. Therefore, they change radio access technology (RAT) to access a 3G or 2G network that has a better signal. However, after few seconds they drop the connection and try (without success) to access again the network with high QoS and bad connectivity.

This fact proves that the isolation forest model using the subscribers dataset, is capable to identify anomalies in the users' activities.

The 108 IMSI that perform anomalous activities have been searched also among the anomalies detected in the other models. It results that the LOF algorithm (in its best setting) has all 108 IMSI, while the EIF model (in its best setting) returns 101 out of 108 anomalies. This is confirmed also by looking at the similarities in the LOF and EIF "sessions" graphs on the user dataset (figure 5.2, 5.10). This means that even the other models are capable of spotting these kind of anomalies, however they have some drawbacks in terms of time complexity and number of false positives that need to be taken to account. These are also the reasons why the standard isolation forest algorithm is the best candidate in the anomaly detection system.

6
Conclusion

In this document has been presented the creation of a machine learning based anomaly detection system for the subscribers activities in Athonet's private 4G networks. The goal of this work was finding abnormal behaviors in the charging data records, identifying possible frauds or inconsistencies inside private networks. After presenting the problem and motivating the reasons behind the realization of this work, the project starts by describing the details of the domain in which this anomaly detection system works.
Chapter 2 presents the technical background for 4G and 5G network architectures, and a detailed explanation of the charging system of 4G networks. With the help of the 3GPP doumentation, it has been presented the list of parameters and the description of the P-GW charging data records. These latter are the data source from where the anomaly detection system looks for anomalies.
Chapter 3 describes all the processes implemented to acquire, convert, store and extract meaningful data from the P-GW CDRs. In particular, it shows the pipeline that includes the python scripts that automate the CDR conversion using the CDR-Dump tool and the population of the CDR database, created according to the P-GW CDR structure stated in 3GPP.
Chapter 4 presents all the techniques to extract the user and session datasets, together with a description of all the machine learning models implemented. After testing the models, results show that DBSCAN has come issues related to its computational complexity. The Local Outlier Factor returns a high amount of anomalies, many of which cannot reconduct to an actual abnormal behaviour. The Isolation Forest model is the fastest and most precise model, even if

it still includes some false positives. Finally, results on the Extended Isolation Forest show similar performances with respect to the standard isolation forest in terms of number of detected anomalies. However, it has been shown that the algorithm in the fully extended case is slower than the standard IF. Therefore, the standard isolation forest is considered the best candidate for the anomaly detection task. It offers the best trade-off between number of false positives and time complexity, which are crucial factors when dealing with the huge amount of CDRs generated by the Athonet's client networks.
It has been shown in chapter 5 that the best dataset to identify the anomalies is the user dataset, in which were found 108 IMSIs that perform abnormal operations. This result has been found in all the models, and confirmed by analyzing the Athonet's dashboard.

6.1 Future Works

Athonet's goal for this project is to merge the anomaly detection system and their dashboard, in order to help its experts at finding erroneous behaviors inside the private networks. Even if this work already presents some interesting results, there are still some problems that need to be addressed. The first issue is the time complexity of the database population script. Considering the dimension and the generation frequency of each CDR file, one possible solution to improve the python algorithm is introducing multithreading or using directly the data stored in the dashboard database.
Another problem that emerged during the project is the complexity of the DBSCAN, that prevented the visualization of data clusters and outliers. Increasing the hardware performances of the machine that runs DBSCAN or implementing one of the many DBSCAN variants, such the ones proposed in [41], [42], are two possible solutions to this common problem.
Another goal that will make the anomaly detection system more reliable is to reduce the number of false positives. To achieve this, it is important to train the models with more data, preferably with the CDRs related to an entire day or week of activities. In this way is possible to have a wider view on the subscribers activities, especially when using the subscribers dataset. Once these aspects are refined, this anomaly detection system can become one of the services offered in the Athonet's dashboard. By merging the information of the dashboard and the outputs of the anomaly detection system, the security of Athonet's private networks will definitely increase. It will also help the experts that work with the dashboard by generating some alerts when a suspicious activity arises.
Togheter with the standard machine learning algorithm, another interesting challenge for the

future could be the implementation of the anomaly detection system using deep neural networks. Especially with the help of generative models, the anomaly detection system can benefit and improve from the fact that the network itself creates new anomalies. This counters the unavoidable problem of each anomaly detection system, that is the small amount of a anomalous patterns in the data. Another future improvement for this system is to make it compatible also with 5G CDRs. In this way, it will be able to analyze the charging operations that come from the customers that choose Athonet's private 5G network.

References

[1] S. Frattasi, H. Fathi, F. H. Fitzek, R. Prasad, and M. D. Katz, "Defining 4g technology from the users perspective," *IEEE network*, vol. 20, no. 1, pp. 35–41, 2006.

[2] C. Cox, *An introduction to 5G: the new radio, 5G network and beyond*. John Wiley & Sons, 2020.

[3] A. Bellin, M. Centenaro, N. di Pietro, A. Ishaq, D. Munaretto, D. Ronzani, A. Spinato, S. Tomasin, and F. Granelli, "Autonomous private mobile networks: State of the art and future challenges," *IEEE Communications Standards Magazine*, vol. 7, no. 2, pp. 24–31, 2023.

[4] G. Brown and A. PRINCIPAL, "Private 5g mobile networks for industrial iot," *Heavy Reading, White Paper, Qualcomm Inc*, 2019.

[5] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[6] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[7] S. Sullivan, A. Brighente, S. A. Kumar, and M. Conti, "5g security challenges and solutions: a review by osi layers," *IEEE Access*, vol. 9, pp. 116 294–116 314, 2021.

[8] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[9] A. G. Sulaiman and I. F. Al Shaikhli, "Comparative study on 4g/lte cryptographic algorithms based on different factors," *International Journal of Computer Science and Telecommunications*, vol. 5, no. 7, pp. 7–10, 2014.

[10] A. N. Bikos and N. Sklavos, "Lte/sae security issues on 4g wireless networks," *IEEE Security & Privacy*, vol. 11, no. 2, pp. 55–62, 2012.

[11] J. Cichonski, J. Franklin, and M. Bartock, "Guide to lte security," National Institute of Standards and Technology, Tech. Rep., 2016.

[12] N. Seddigh, B. Nandy, R. Makkar, and J.-F. Beaumont, "Security advances and challenges in 4g wireless networks," in *2010 Eighth International Conference on Privacy, Security and Trust*. IEEE, 2010, pp. 62–71.

[13] M. Al Shinwan, T.-D. Huy, and K. Chul-Soo, "A flat mobile core network for evolved packet core based sae mobile networks," *Journal of Computer and Communications*, vol. 5, no. 5, pp. 62–73, 2017.

[14] W. Zhang, J. Jaehnert, and K. Dolzer, "Design and evaluation of a handover decision strategy for 4th generation mobile networks," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, vol. 3. IEEE, 2003, pp. 1969–1973.

[15] H. Tang, N. Yang, Z. Zhang, Z. Du, and J. Shen, *5G NR and Enhancements: From R15 to R16*. Elsevier, 2022.

[16] M. Olsson and C. Mulligan, *EPC and 4G packet networks: driving the mobile broadband revolution*. Academic Press, 2012.

[17] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan, *5G Core Networks: Powering Digitalization*. Academic Press, 2019.

[18] X. Lin and N. Lee, "5g and beyond," *Cham, Switzerland: Springer Nature Switzerland AG*, 2021.

[19] R. Törnkvist and C. Shan, "Charging and billing architecture for 5g network," *Journal of ICT Standardization*, pp. 185–194, 2019.

[20] D. C. C. D. L. Margaret Chiosi, Ivano Guardini, "Network functions virtualisation," *White Paper, ETSI*, 2012.

[21] L. D. A. E. S. G. A. M. D. N. Y. N. J. P. J. T. X. W. B. X. H. Z. Xuhui Cai, Hui Deng, "Evolving nfv towards the next decade," *White Paper, ETSI*, 2023.

[22] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[23] GSMA, "An introduction to network slicing," *White Paper,GSMA*, 2017.

[24] N. Mitra, "Efficient encoding rules for asn. 1-based protocols," *AT&T Technical Journal*, vol. 73, no. 3, pp. 80–93, 1994.

[25] H.-A. Lin, "Estimation of the optimal performance of asn. 1/ber transfer syntax," *ACM SIGCOMM Computer Communication Review*, vol. 23, no. 3, pp. 45–58, 1993.

[26] J. Larmouth, *ASN. 1 complete*. Morgan Kaufmann, 2000.

[27] J. D. Drake and J. C. Worsley, *Practical PostgreSQL*. " O'Reilly Media, Inc.", 2002.

[28] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[29] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[30] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, 2008, pp. 21–49.

[31] Z. Ghahramani, "Unsupervised learning," in *Summer school on machine learning*. Springer, 2003, pp. 72–112.

[32] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[33] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[34] C. Labrín and F. Urdinez, "Principal component analysis," in *R for Political Data Science*. Chapman and Hall/CRC, 2020, pp. 375–393.

[35] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, 2020.

[36] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[37] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE transactions on knowledge and data engineering*, vol. 33, no. 4, pp. 1479–1489, 2019.

[38] R. J. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 2, p. e1343, 2020.

[39] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data and Cognitive Computing*, vol. 5, no. 1, p. 1, 2020.

[40] G. Wu, L. Cao, H. Tian, and W. Wang, "Hy-dbscan: A hybrid parallel dbscan clustering algorithm scalable on distributed-memory computers," *Journal of Parallel and Distributed Computing*, vol. 168, pp. 57–69, 2022.

[41] A. Bryant and K. Cios, "Rnn-dbscan: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1109–1121, 2017.

[42] N. Gholizadeh, H. Saadatfar, and N. Hanafi, "K-dbscan: An improved dbscan algorithm for big data," *The Journal of supercomputing*, vol. 77, pp. 6214–6235, 2021.

Acknowledgments

Prima di tutto vorrei ringraziare il prof. Milani per avermi introdotto ad Athonet e per avermi seguito nella realizzazione della tesi. Inoltre un ringraziamento va ad Athonet (e HPE) per avermi accolto in questi ultimi 4 mesi. Grazie ad Antonio per avermi supervisionato durante lo stage e a tutti coloro che mi hanno aiutato nel portare a termine questo progetto e che mi hanno fatto sentire parte del team dell'azienda.

Questo traguardo segna la fine di un importante capitolo della mia vita, e ciò lo devo a tutte le persone che mi sono state vicine in questi anni di università. Non nego che è stato un percorso tortuoso, segnato da momenti felici ma anche da alcuni carichi di ansia e tristezza, che difficilmente avrei superato senza il vostro aiuto. Il vostro costante affetto e supporto mi ha spronato a dare il meglio di me, facendomi diventare una persona più matura. Se stiamo festeggiando è anche merito vostro, grazie.

In particolare vorrei ringraziare i miei genitori, per i sacrifici che hanno fatto per permettermi di intraprendere il mio percorso universitario. Grazie alla mamma Alba, per essermi stata sempre vicina nei momenti di ansia e di sconforto. Tu mi conosci meglio di chiunque altro, e per questo capivi le mie giornate no appena mi salutavi al mattino. Ma bastava una parola, un abbraccio o una semplice carezza per darmi la forza necessaria ad affrontare le sfide poste di fronte a me. Grazie anche al papà Valter, per essere stato sempre al mio fianco. Ho sentito la tua vicinanza attraverso le parole di sostegno e le battute che rendevano più leggere le mie pesanti giornate da studente universitario. Voi siete stati e sarete sempre i miei punti di riferimento, vi voglio un bene.

Vorrei dedicare anche un ringraziamento al mio fratello Davide che, più di tutti, capisce tutte le difficoltà e le emozioni (belle e brutte) che ho affrontato in questi anni. Grazie alla tua esperienza sei sempre riuscito a darmi dei preziosi consigli, che mi hanno aiutato a superare le materie più complicate. Inoltre ti sono grato per avermi sostenuto in questi 6 lunghi anni di università. Te ci sei sempre stato quando ero di fronte ad un brutto voto o ad una insufficienza, facendomi ragionare su ciò che era giusto da fare in quei momenti. Sei il miglior fratello che potessi desiderare, grazie di cuore.

Voglio anche ringraziare tutti parenti che mi hanno supportato durante questo mio percorso. Grazie alla nonna Luciana e Giovanna per avermi pensato sempre, dedicandomi chissà quante

preghiere e rosari nei periodi di esami. Grazie allo zio Claudio e alla Patrizia per non essersi dimenticati di sostenermi ad ogni occasione in cui ci vediamo. Grazie alla zia Sabrina e allo zio Luca, alla zia Barbara e a tutti i miei cugini per il vostro affetto e sostegno nei miei confronti.
Un ringraziamento va ai miei compagni di magistrale Christian, Alex, Enrico ed Ismael, con la quale ho condiviso mille gioie e dolori. Nonostante le limitazioni del Covid, le lezioni a distanza e le pesanti sessioni d'esame, siamo comunque riusciti a ritargliarci dei momenti divertenti che difficilmente scorderò.
Una dedica va anche a tutti i miei amici, senza i quali sarei probabilmente già impazzito. E' grazie a voi che sono riuscito a distrarmi e a divertirmi anche nei momenti più complicati. Vi sono grato per avermi sopportato nei giorni in cui non volevo parlare con nessuno e per avermi tirato fuori di casa nei momenti in cui avrei voluto stare da solo chiuso nei miei pensieri. Grazie per i momenti di spensieratezza che mi avete regalato durante questi anni di università.
Infine vorrei rivolgere un pensiero anche al nonno Carlo. Grazie per aver sempre creduto in me, festeggiando quando arrivavano i bei voti e dicendomi di tenere duro nei momenti difficili. Inutile dire che mi manchi tantissimo, e che avrei voluto che tu fossi qui insieme a noi a celebrare questo mio importante traguardo. Mi piace pensare che, ovunque tu sia, stia festeggiando con un bicchiere di vino come avresti probabilmente fatto in questo giorno. Questo lavoro è dedicato anche a te, nonno Carlo. Ti voglio bene.