Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA

IN INGEGNERIA ELETTRONICA,TELECOMUNICAZIONI E TECNOLOGIE DELL'INFORMAZIONE

Ciclo 35

Settore Concorsuale: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico Disciplinare: ING-INF/03 - TELECOMUNICAZIONI

RADIO FREQUENCY COMMUNICATION AND FAULT DETECTION FOR RAILWAY SIGNALLING

Presentata da: Pasquale Manuele Grilli

Coordinatore Dottorato

Aldo Romani

Supervisore

Walter Cerroni

Co- supervisore

Alessandra Costanzo

Esame finale anno 2023

Surround Yourself with people whose eyes
light up when they see you coming
Slowly is the fastest way to get
to where You want to be
The top of one mountain is the bottom
of the next, so keep climbing.

True progress is that which places
technology in everyone's hands.
(H. Ford).

Acknowledgment

Completing this challenging three year project presented in this thesis would not have been possible without the help of numerous people who supported me during my PhD studies. I am grateful to my family, friends, and colleagues, both long-time and new, those here in Italy, those a bit further away, and those on the other side of the world for their encouragement and support during moments of discouragement when achieving this objective seemed impossible.

Returning to study after a 13 year break and managing daily job commitments was not an easy decision, and I often wondered where I found the strength to continue. Over the past few years, I faced several difficult moments in my personal and professional life, but this journey taught me the importance of never giving up on my dreams. Despite the challenges, I am grateful for this experience because it helped me overcome the darkest times and taught me that commitment and determination, even in the face of adversity, can lead to achieving what one is striving for.

Now that I have finally reached the summit, I am satisfied and proud of the journey taken, and I am looking forward enjoying.

GRILLI PASQUALE MANUELE

Contents

Acknowledgment 4

List of Figures 9

List of Tables 1

Abstract 3

1 Chapter one: Introduction 4

2 Chapter two: Railway system over wired private network 7

2.1 Introduction 7

2.2 Current architecture description 7

2.3 Wired private network details 9

2.4 Signalling Communication 13

2.4.1 FSFB/2 protocol brief description 13

2.4.2 PVS protocol brief description 15

2.4.3 Safe Response time analysis 15

3 Chapter three: Railway system over public network 18

3.1 First topic of the research 18

3.2 Railway architecture based on LTE description 20

3.2.1 Proprietary signalling protocols for communication over LTE 25

3.2.2 Distributed architecture over LTE network 30

3.2.3 Coverage level and resource allocation 32

3.2.4 Cybersecurity feature handling on a public network 40

3.2.5 LTE devices for radio frequency network access 50

3.2.6 Telecom Service Level Agreement 53

3.2.7 Trackside equipment handled through LTE existing infrastructure 56

3.3 Research and Design activity results 59

4 Chapter four: Machine Learning world 62

4.1 Machine Learning introduction 62

4.1.1 Unsupervised Learning 65

4.1.2 Supervised Learning 66

4.1.3 Reinforcement Learning 67

4.2 Dimensionality reduction techniques in ML 70

4.2.1 PCA: Dimensionality reduction in details 71
4.2.2 T-SNE: Dimensionality reduction in details 78
4.3 Clustering technique 85
4.3.1 Hierarchical clustering algorithm 87
4.3.2 Fuzzy clustering algorithm 88
4.3.3 K-means: Partitional clustering algorithm 89
4.4 Machine Learning process mindset 92
5 Chapter five: Detection for Railway equipment 96
5.1 Second topic of the research 96
5.2 ML in literature for Railway application field 97
5.3 STDS-AF: Description and working principle 102
5.4 STDS-AF: Railway equipment to be diagnosed 108
5.5 Fault detection algorithm workflow 112
5.6 Fault detection algorithm results 115
5.6.1 State Detection Use Case 1 116
5.6.2 State Detection Use Case 2 118
5.6.3 State Detection Use Case 3 120
5.6.4 State Detection Use Case 4 123
5.6.5 One Fault Detection Use Case 1 126
5.6.6 One Fault Detection Use Case 2 128
5.6.7 One Fault Detection Use Case 3 131
5.6.8 One Fault Detection Use Case 4 134
5.6.9 Two Faults Detection Use Case 1 137
5.6.10 Two Faults Detection Use Case 2 139
5.6.11 Two Faults Detection Use Case 3 142
5.6.12 Two Faults Detection Use Case 4 145
6 Chapter six: Conclusion and future developments 148
Bibliography 151
Appendix 154

List of Figures

Figure 1 – Wired Private Network architecture 8
Figure 2 – Wired Private Network simplified architecture 10
Figure 3 – Wired Private Network MRP operational scheme 11
Figure 4 – Local Area Network sectorized 12
Figure 5 – FSFB2 transmission flow exchange 14
Figure 6 – Safe Response time flow computation 16
Figure 7 – Experimental track Padiglione - Nettuno 20
Figure 8 – Experimental track over radio frequency media 21
Figure 9 – LTE signals strength and quality values 23
Figure 10 – RSSI and atmospheric attenuation effect 24
Figure 11 – Railway architecture based on LTE infrastructure 25
Figure 12 – Tunnelling concept 27
Figure 13 – IPSec Tunnel 28
Figure 14 – VPN architecture implementation 30
Figure 15 – Distributed architecture over LTE 31
Figure 16 – RSRP evaluation on Padiglione Nettuno experimental track 32
Figure 17 – Test Bench 33
Figure 18 – Clustering and Reuse distance for radio resource unit 37
Figure 19 – GSM-R frequency spectrum 39
Figure 20 – FTDMA frequency allocation 39
Figure 21 – Cybersecurity CIA triad 41
Figure 22 – CSP platform fitted at control room 46
Figure 23 – 802.1x components 48
Figure 24 – 802.1x authentication flow 49
Figure 25 – LTE router installation on peripherical nodes 50
Figure 26 – RV55 frequency characterization 51
Figure 27 – RV55 additional specifications 52

Figure 28 – Antenna and gain plot 52
Figure 29 – SiSo Vs MiMo technology benefit 53
Figure 30 – MPLS infrastructure and dedicated APN 56
Figure 31 – Real railway system communication over LTE 57
Figure 32 – Real railway system communication over LTE setup 58
Figure 33 – Unsupervised Learning process overview 65
Figure 34– Supervised Learning process overview 67
Figure 35 – Reinforcement Learning cycle 69
Figure 36– Reinforcement Learning process overview 69
Figure 37 – PCA dimensionality reduction example 73
Figure 38 – PCA: Toy example input dataset 73
Figure 39 – PCA: Toy example labels removed 73
Figure 40 – PCA: Toy example 3-D to 2-D dimensionality reduction 78
Figure 41 – T-SNE: Data represented in 2D 79
Figure 42 – T-SNE: Data projected in 1D 79
Figure 43 – T-SNE: Gaussian Vs t-Student Distribution tails 81
Figure 44 – T-SNE: Data represented in 1D 82
Figure 45 – PCA: representation of Digits and MINST dataset 83
Figure 46 – T-SNE: representation of Digits and MINST dataset 83
Figure 47 – Clustering techniques 87
Figure 48 – Hierarchical Clustering 88
Figure 49 – Elbow method showing optimal k 90
Figure 50 – Machine Learning process 92
Figure 51 – Confusion Matrix for binary class 94
Figure 52 – STDS-AF environment 103
Figure 53 – Mechanical Joint 106
Figure 54 – “S” Electrical Joint 106
Figure 55 – “S” Electrical Joint - LC equivalent scheme 107
Figure 56 – “S” Electrical Joint working principle 107
Figure 57 – San Pellegrino STDS-AF installation 109

Figure 58 – STDS-AF Current Voltage measurement 109
Figure 59 – STDS-AF Current Voltage measurement shape 111
Figure 60 – Fault Detection Algorithm workflow 112
Figure 61 – Data pre-processed for constant sampling rate 113
Figure 62 – Features normalized 113
Figure 63 – Clusters expectation 114
Figure 64 – Optimal k for UC1 state detection 116
Figure 65 – Cluster representation UC1 state detection 117
Figure 66 – Accuracy UC1 state detection 117
Figure 67 – Graphic state detection for UC1 118
Figure 68 – Optimal k for UC2 state detection 118
Figure 69 – Cluster representation UC2 state detection 119
Figure 70 – Accuracy UC2 state detection 120
Figure 71 – Graphic state detection for UC2 120
Figure 72 – Optimal k for UC3 state detection 121
Figure 73 – Cluster representation UC3 state detection 121
Figure 74 – Accuracy UC3 state detection 122
Figure 75 – Graphic state detection for UC3 122
Figure 76 – Optimal k for UC4 state detection 123
Figure 77 – Cluster representation UC4 state detection 124
Figure 78 – Accuracy UC4 state detection 124
Figure 79 – Graphic state detection for UC4 125
Figure 80 – Optimal k for UC1 one fault detection 126
Figure 81 – Cluster representation UC1 one fault detection 127
Figure 82 – Accuracy UC1 one fault detection 127
Figure 83 – Graphic one fault detection for UC1 128
Figure 84 – Optimal k for UC2 one fault detection 129
Figure 85 – Cluster representation UC2 one fault detection 129
Figure 86 – Accuracy UC2 one fault detection 130
Figure 87 – Graphic one fault detection for UC2 131

Figure 88 – Optimal k for UC3 one fault detection 131
Figure 89 – Cluster representation UC3 one fault detection 132
Figure 90 – Accuracy UC3 one fault detection 133
Figure 91 – Graphic one fault detection for UC3 133
Figure 92 – Optimal k for UC4 one fault detection 134
Figure 93 – Cluster representation UC4 one fault detection 135
Figure 94 – Accuracy UC4 one fault detection 135
Figure 95 – Graphic one fault detection for UC4 136
Figure 96 – Optimal k for UC1 two faults detection 137
Figure 97 – Cluster representation UC1 two fault detection 138
Figure 98 – Accuracy UC1 two faults detection 139
Figure 99 – Graphic two faults detection for UC1 139
Figure 100 – Optimal k for UC2 two faults detection 140
Figure 101 – Cluster representation UC2 two faults detection 140
Figure 102 – Accuracy UC2 two faults detection 141
Figure 103 – Graphic two faults detection for UC2 142
Figure 104 – Optimal k for UC4 one fault detection 142
Figure 105 – Cluster representation UC3 two faults detection 143
Figure 106 – Accuracy UC3 two faults detection 144
Figure 107 – Graphic two faults detection for UC3 145
Figure 108 – Optimal k for UC4 two faults detection 145
Figure 109 – Cluster representation UC4 two faults detection 146
Figure 110 – Accuracy UC4 two faults detection 147
Figure 111 – Graphic two faults detection for UC4 147

List of Tables

Table 1- VPN parameters chosen 29
Table 2- IEC 62443 security levels 43
Table 3- Impact criteria 45
Table 4- Machine Learning Properties 65
Table 5- STDS-AF Track circuit Layouts 104
Table 6- STDS AF Electrical Joints type 105
Table 7- Machine Learning Properties 111
Table 8- List of the algorithm test 116
Table 9- Accuracy of the Fault detection model 149

Abstract

The continuous and swift progression of both wireless and wired communication technologies in today's world owes its success to the foundational systems established earlier. These systems serve as the building blocks that enable the enhancement of services to cater to evolving requirements. Studying the vulnerabilities of previously designed systems and their current usage leads to the development of new communication technologies replacing the old ones such as GSM-R in the railway field. The current industrial research has a specific focus on finding an appropriate telecommunication solution for railway communications that will replace the GSM-R standard which will be switched off in the next years.

Various standardization organizations are currently exploring and designing a radiofrequency technology based standard solution to serve railway communications in the form of FRMCS (Future Railway Mobile Communication System) to substitute the current GSM-R. Bearing on this topic, the primary strategic objective of the research is to assess the feasibility to leverage on the current public network technologies such as LTE to cater to mission and safety critical communication for low density lines. The research aims to identify the constraints, define a service level agreement with telecom operators, and establish the necessary implementations to make the system as reliable as possible over an open and public network, while considering safety and cybersecurity aspects.

The LTE infrastructure would be utilized to transmit the vital data for the communication of a railway system and to gather and transmit all the field measurements to the control room for maintenance purposes. Given the significance of maintenance activities in the railway sector, the ongoing research includes the implementation of a machine learning algorithm to detect railway equipment faults, reducing time and human analysis errors due to the large volume of measurements from the field.

Keywords: *Railway, LTE, Public Network, Machine Learning, Fault detection algorithm*

1 Chapter one: Introduction

Interactive conversation between two individuals or the integration of different devices relies on the main operating principle of communication and its transmission medium. Telecommunications have come a long way since the use of smoke signals by Native Americans, and Guglielmo Marconi's first radio communication in 1895 marked a significant turning point. Over the years, telecommunications have evolved in the radio frequency environment, enabling the transmission of an increasing amount of data, including voice, video, and data. The great success of telecommunications was achieved by enabling real time communication in various fields of application among users or devices situated in one or multiple geographical locations, surpassing the architectural, physical, and cost limitations of wired infrastructures.

In the railway environment, reliable and real time communication is a crucial area of interest, given the complex system that integrates trackside and on-board subsystems which must operate continuously, efficiently and safely. Nowadays, railway communications heavily rely on wired infrastructure for exchanging data among trackside equipment, while communication between on board and trackside equipment utilized GSM-R (Global System for Mobile communication - Railway) combined with GPRS (General Packet Radio Service) though packet switching communication. As the need to introduce new services while maintaining system interoperability continues to grow, GSM-R is set to be decommissioned by 2030. Consequently, discussions are underway to identify the technological successor in the railway industry, and the International Union of Railways (UIC) is collaborating with the European Union Agency for Railways to design the Future Rail Mobile Communication System (FRMCS), which will based most likely on 5G Radio Frequency technology.

Since 5G is still a hybrid technology that relies on the old 4G, Alstom's research and development department have been involved in a European project that assesses the feasibility of utilizing current public networks to cater to the mission and safety critical signalling communications requirements of low-density railway lines that are equipped with the ERTMS (European Rail Traffic Management System) standard. To ensure a good quality of service, a Service Level Agreement was requested from telecommunications operators, despite the legislative restrictions that prevent the allocation of dedicated resources such as frequencies or radio resource blocks, except for military purposes with the 4G solution. The slicing feature, which will be

introduced in 5G, may resolve this issue by enabling the differentiation of services based on the user's specific requirements.

In a railway environment, the reliability of subsystems is just as critical as communication since a failure could result in the normalization of communication protocols, leading to the interruption of railway services. Public network must be leveraged to realize the communication between the control room and the peripherical nodes for signalling purpose and maintenance as well. Consequently, the logs obtained from on field measurements must be utilized for maintenance activities to identify the normal and anomalous states of railway equipment. This maintenance activity represents another area of research that has been automated to minimize human errors. On this topic, an algorithm based on machine learning techniques has been developed in Python code, which is included in the appendix, to detect faults experienced on the railway equipment chosen to be diagnosed.

Based on what expressed above for the time being, the current industrial research work has focused the attention on two main aspects of significant interest at an industrial level such as the real feasibility of exploiting the current public networks for railway communications and automating the analyses for maintenance purposes through the implementation of machine learning algorithm and techniques which are described in the introductory chapter of the thesis.

To comprehend the modifications required with the introduction of a new telecommunication technology, it is essential to understand the functioning of the current railway system in terms of architecture, integration of different devices, and communication protocols for data exchange. Therefore, chapter 2 outlines the architecture that governs a railway system, which is primarily based on a wired and redundant infrastructure. This infrastructure is represented by the backbone constructed using fibre to establish a connection between devices located at the central side and those situated along the line in peripheral geographical areas.

Chapter 3 describes the first topic of the industrial research, focusing on replacing the redundant fibre architecture with an LTE based architecture to ensure continuous communication between the central side and peripheral nodes. This chapter outlines the necessary modifications in terms of architecture, communication protocol constraints, and security measures to ensure the confidentiality, integrity, and availability of exchanged data. The definition of a service level agreement with the telecom provider is also required. Tests conducted with MISE (Ministero dello Sviluppo Economico) have yielded positive preliminary results in leveraging the LTE infrastructure for railway communication if there are low interferences and

adequate signal power coverage. A command sent from a control room to a peripheral node to operate railway equipment was successfully executed between Bologna and Florence using the public LTE infrastructure without any significant impact on latency or the overall safety system response time. In conclusion, the first topic of the industrial research confirms the real feasibility of adopting the LTE network with the introduction and implementation of appropriate cyber policies and encryption techniques to ensure the confidentiality, integrity, and availability of exchanged data.

The second argument aims to utilize the same air channel to collect all the logs and measurements from on field equipment for railway maintenance. This is accomplished by creating a model based on machine learning techniques, which is implemented through Python code provided in the Appendix. The model helps in detecting the normal and anomalous state of railway equipment by analysing the on-field measurements. This is done to save time in detecting anomalies due to the large amount of data gathered from all the railway equipment at the central site and to reduce human errors in maintenance activities. Chapter 4 is a detailed exploration of machine learning, covering various learning techniques based on the type of problem to be addressed, such as classification, regression, clustering, and data dimensionality reduction having to face with large quantities of measurements received from the field.

After gaining a deeper understanding of machine learning and participating in the Summer School of Information Engineering "Silvano Pupolin" in July 2021, where machine learning techniques were discussed in various application fields, Chapter 5 explains the techniques adopted based on the available data structure. The goal is to detect the normal and anomalous status from the measurements of a selected railway component installed in the field to manage the occupancy of the track circuits. The process involves developing a Python code attached in the Appendix section and following it through all the phases, including data acquisition, pre-processing, and feeding it to the algorithm aiming the identification of the optimal parameters to obtain the high accuracy in the fault detection.

Chapter 6 summarizes the positive results and feasibility of the two main arguments presented in this industrial research. It also identifies the improvements that need to be made to achieve a more reliable system based on communication over a public LTE infrastructure and to develop an algorithm capable of predicting faults in railway equipment.

2 Chapter two: Railway system over wired private network

2.1 Introduction

The research has been conducted in a railway scenario to leverage on current Public Network based on radio frequency technology instead of a wired Private Network to serve mission and safety critical communication and to reduce infrastructure cost, as per interest of the industrial PhD. The objective of this chapter is to illustrate a generic railway architecture as nowadays implemented, to deep knowledge which are the network elements impacting the transition from a private network based on wired infrastructure to a public network characterized by radio frequency technology.

2.2 Current architecture description

Nowadays, the architecture of a typical railway system is based on a wired infrastructure where communication between the control room and the peripheral nodes occurs through the installation of a redundant backbone realized in fibre. This backbone is used to manage trackside equipment such as level crossings, turnouts, and track circuits, and is designed to overcome single network failures and create a more reliable infrastructure. In this architecture the intelligence of the system is located at Interlocking (IXL) side fitted at central location, while the peripherical nodes serve as the actuators of the command. As a result, each communication must pass through the central side for sending commands to move field equipment and receive controls from the peripherical nodes as depicted on the below picture. The centralized architecture might be commuted in distributed architecture by including an advanced object controller (e-SMIO), where part of the logic is allocated to the peripherical node to speed up the response time in the management of trackside equipment.

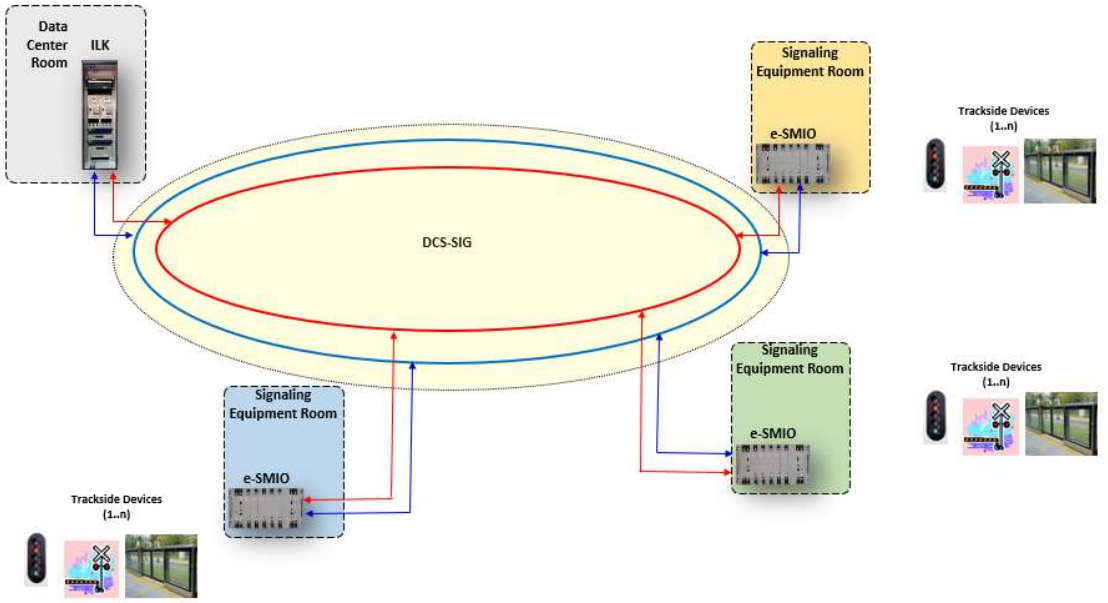


Figure 1 – Wired Private Network architecture

The main subsystems in the above architecture are:

- ✓ ILK (Interlocking): This refers to the Interlocking Kernel, which is located at the central side of the system architecture. It is a safety system that manages the signalling logic for rail traffic within a specified medium to large area.
- ✓ OC (Object Controller/e-SMIO): It represents a safety railway component that integrates the online functions of the object controller and interlocking kernel for small area application. While it can be used in larger applications, its primary use is to execute local application logic for faster response times in steering values to trackside objects, collecting detected values from trackside objects, and executing optional signalling logic.
- ✓ DCS-SIG (Data Communication System for Signalling): This refers to the network subsystem, which is characterized by a redundant configuration that enables communication among all subsystems within the architecture.

Moreover, to facilitate maintenance and centralize the visualization of the status of railway equipment, the architecture may include additional subsystems such as:

- ✓ DMS (Diagnostic Maintenance System): This is a non vital component that enables system diagnostics and performs a combination of technical and administrative functions, including supervisory and technician functions.
- ✓ S-HMI (Safety Human Machine Interface): This is a safety component that allows the operator to issue commands and view the status of trackside objects through video representation.

2.3 Wired private network details

Currently the network managing the signalling architecture of a railway system is designed as a Level 3 redundancy network topology and categorized as SIL0 according to the CENELEC international classification for the safety of railway systems EN 50128 and EN 50129 regulations, which meets the following general communication requirements:

- ✓ Topology and timing access described by the IEEE 802.1, 802.2 and 802.3 protocols.
- ✓ IPv4 Ethernet communications for different types of information transfer, such as Unicast, Multicast and Broadcast.
- ✓ Overall latency less than 50 milliseconds.
- ✓ Segregation of network traffic at layer 3 of the ISO / OSI stack.

Specifically, the network managing the signalling architecture SML400-ACCM can be functionally divided in three different physical LANs (Local Area Networks) each based on the DCS-SIG architecture and endowed by both red and blue subnetworks:

- ✓ An Operational Central Network facilitating communication among all the devices within the central side area and forwards data to external interfaces.
- ✓ A Backbone Network ensuring the communication between the devices comprised in the operational central network and those managed by the peripheral network.
- ✓ A peripheral Network which is installed on each peripherical nodes to enable connectivity between object controller located along the railway line and the trackside equipment that needs to be managed.

Each of these three types of networks operates as a Layer 2 Ethernet network, creating separate areas where broadcast communications and the effects of a local failure are contained. However, communication

between these geographically distributed networks is facilitated through Layer 3 devices that segregate traffic generated within a local network and enable transmission between different networks. This transmission is typically limited to unicast and multicast packets. The distinction between the three local area networks is illustrated on the below picture having simplified the overall network by assuming that only one peripheral station is connected to the operational central network.

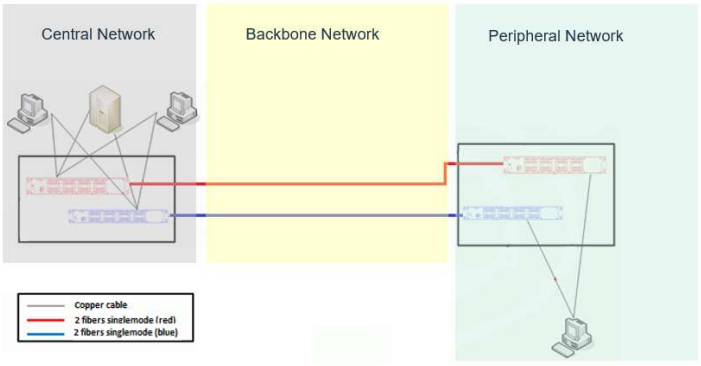


Figure 2 – Wired Private Network simplified architecture

In the above picture, and in the overall system, communication is achieved through a redundancy concept that duplicates equipment and connections at each site. This results in two distinct networks working in parallel (for example, the red network and blue network as shown in the images of this chapter) to ensure complete device reachability in the event of a single fault. The backbone network and the peripheral network are basically constructed using dedicated optical fibres consisting of two separate cables laid along the entire backbone of the system in a loop topology configuration where for instance the red network might be setup on the primary optical cable, while the blue network on the secondary optical cable.

The loop topology is implemented such that the different peripheral sites are added to the backbone network alternately. This connection method enables the "breaking down" of long-distance sections and creates a potentially infinite and modular network consisting of intermediate sections of limited distance for interconnecting the various peripherical sites. At functional level, each loop included in the backbone network is treated as a local area network and managed by the Media Redundancy Protocol (MRP) for fault management. The Media Redundancy Protocol, published as IEC 62439 by the International Electrotechnical Commission in 2010, is a data network protocol designed specifically for networks with a loop topology. Its primary objective is to manage single faults by automatically removing any physical loop in fault triggered on the network, which would otherwise lead to the saturation of the available bandwidth.

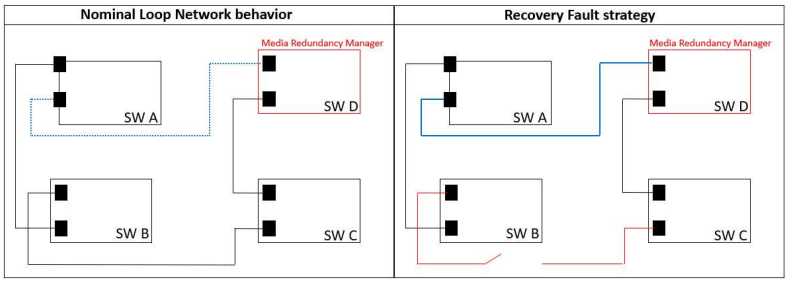


Figure 3 – Wired Private Network MRP operational scheme

In a network characterized by a loop topology where the MRP protocol is applied in case of a single fault, a Media Redundancy Manager (MRM) is defined. All other nodes in the network are considered Media Redundancy Clients (MRCs). The master switch sends MRP control packets on the first port of the loop and expects to receive them again on the second port comprised in the loop. MRM and MRC ports, within the loop under observation, support three status: disabled, blocked, and forwarding. Disabled ports drop all received frames, blocked ports drop all received frames except for the MRP control frames, and forwarding ports forward all received frames.

If the control packets sent and received by the MRP protocol are functioning properly, the redundancy manager keeps one of its network interfaces logically blocked for generic data traffic, as depicted by the blue dotted line in the previous picture. In the normal scenario, the loop can be seen as a "horseshoe" without loops where only a single path is possible between the source and destination hosts. In the event of a fault, the MRP control packets sent by the redundancy manager are no longer received. In response, the redundancy manager activates the network link that was previously blocked for data traffic (depicted as a continuous blue line in the fault recovery strategy picture).

This mechanism regenerates a functional "horseshoe" topology, enabling all network devices to be reachable again. It allows the management of loops with a high number of switches, even exceeding 100, and achieves maximum reaction times of approximately 100-200 milliseconds in response to the first fault triggered in the network. However, reaction times may vary depending on the specific network interfaces of the individual subsystems involved in the loop.

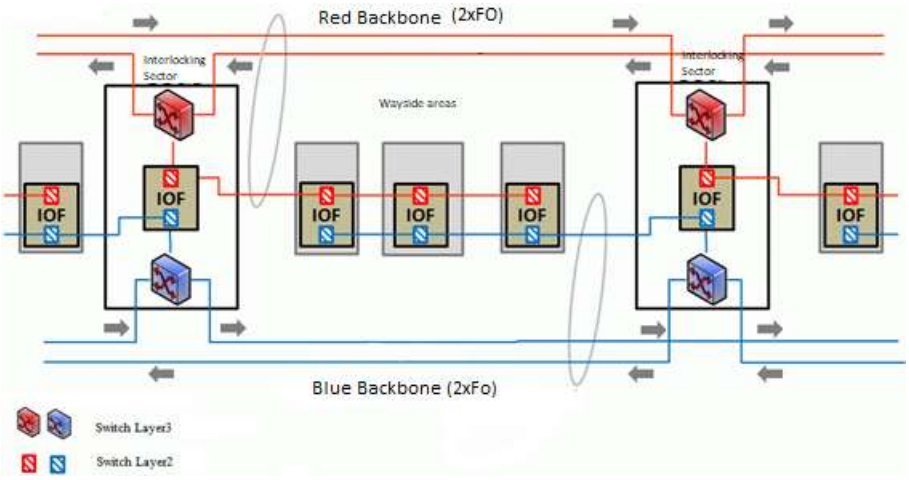


Figure 4 – Local Area Network sectorized

Functional connectivity within each loop of the overall network is established to ensure opposing paths of communication packets between the two redundant networks. This is achieved through a symmetrical configuration of the MRP protocol on the red and blue switches installed in the central side area. This strategy guarantees continuous connectivity between sites, on at least one of the two subnets, even if both the red and blue links are simultaneously interrupted at any point along the line, causing an opening in the network loop. During the MRP protocol reconfiguration transient, communications on one of the two subnets remain available without interruption at all sites. At the end of the MRP protocol reconfiguration, communications on both subnets are restored.

The backbone loops are composed of Level 3 devices that enable the definition of a different local network on the same physical device for each peripherical site and providing the significant advantage of traffic segregation for each node. In the case of the signalling network, Layer 3 switches are positioned at the different interlocking sectors associated with the different stations, creating a separation between the peripheral local network at the site and the backbone network. The backbone network is extended through interconnections with switches from neighbouring sites. This separation between local networks ensures that any traffic disturbances on a network will not spread to the rest of the network but will remain confined to the local network of each interlocking sector.

The linear connections between the different interlocking sectors do not require the use of redundancy protocols for loop management. Thus, any single failure has zero impact on communications due to the presence of parallel red and blue networks. Physically, the network comprises two switches in each

interlocking sector of the railway system, one red and one blue, enabling communication with the SMIO devices for signalling management located in various peripheral nodes along the line. All switches in a section are interconnected by optical fibre within a single subnet. The network devices use dedicated fibres and can manage bandwidths up to 1 Gigabit/s. Optical connections are made using fibres within both a primary and secondary backbone optical cable. The primary backbone optical cable contains fibres dedicated to the red network, while the secondary backbone cable contains fibres dedicated to the blue network.

2.4 Signalling Communication

The communication between the central side and the peripherical nodes involves a continuous and cyclical exchange of valid information among the subsystems described in the above paragraphs. If the data sent and received by the subsystems are timely and coherent, communication based on a wired and reliable architecture using specific protocols is maintained active. This allows for the management of all the trackside equipment in terms of command and control, as well as the movement of the train along the railway line. In case the time validity of the exchanged variables exceeds a certain threshold due to packet loss, traffic congestion, or equipment faults, communication is normalized by setting up restrictive statuses. Currently, the signalling communication protocols adopted are:

- ✓ FSFB/2: Fail Safe Field Bus 2nd generation.
- ✓ PVS: Protocollo Vitale Standard.

2.4.1 FSFB/2 protocol brief description

The FSFB/2 protocol is a patented application layer protocol designed by Alstom, categorized as category 2 according to the EN 50129:2010 regulation, that defines the method of data encapsulation and exchange between applications present in two or more interconnected hosts communicating in a private network. The FSFB/2 protocol is used as the data communication system for signalling, based on a wired infrastructure architecture. The current implementation of the FSFB/2 protocol foresees to use UDP (User Datagram Protocol) at the transport level for cyclic communications. In each cycle, the state of all active command is

transmitted completely within the buffer for both unicast and multicast services. This protocol is characterized by the following types of packets:

- ✓ BSD (Boolean Safety data): It represents the flow of Boolean application data produced by each node in the system and sent in a cyclic manner at each cycle time.
- ✓ SSE (Sequence Start Enquiry): It initiates a synchronization sequence to enable every receiver to accept the data transmitted by the sender.
- ✓ SSR (Sequence Start Reply): It represents the SSE acknowledgement for the purpose of host synchronization.

Each peer on a communication stream is assigned a vital identifier SID (Source Identifier), consisting of 64 bits, while ensuring Hamming distance property. Each BSD packet is labelled with a processing cycle counter (TID time identifier) for sequence and freshness checks. Additionally, the integrity of application data is secured by two 32-bit CRCs to comply with the EN 50129:2010 standards.

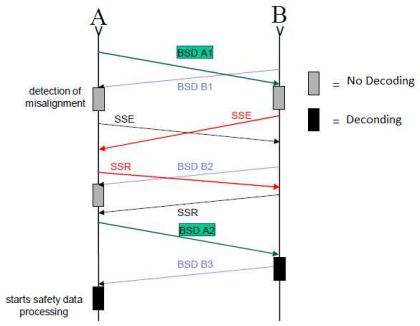


Figure 5 – FSFB2 transmission flow exchange

Essentially, communication between two nodes is established according to the sequence illustrated in the above picture, where:

- ✓ When the receiver receives the first BSD from the remote machine, it generates SSE to initiate the alignment sequence:
    a. A receives BSD-B1, but it is unable to certify it, so it sends SSE to B (black line).
    b. B receives BSD-A1, but is unable to certify it, so it sends SSE to A (red line).
- ✓ Until the receiving node receives SSR, it cannot accept data, as illustrated by the example of receiving BSD B2 before the SSR response (black line).

- ✓ Once SSR is received:
    a. A can certify the received BSD-B3.
    b. B can certify the received BSD-A3.

2.4.2 PVS protocol brief description

The PVS protocol is a cyclic and asynchronous protocol patented by Rete Ferroviaria Italiana, based on Subset-098, and delivered by the UNISIG, representing the European union group of companies for the development of telecommunications and signalling concerning the ERTMS system. It has been designed as category 2 in accordance with the EN 50129:2010 regulation and used for communications between cyclic machines. The current implementation of the PVS protocol foresees to use both UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) at the transport level for cyclic communications. Like the FSFB2 protocol, integrity aspect is verified using the same CRC protection, authenticity is verified using 64-bit identifiers that respect Hamming distance, freshness verification is based on the same pseudorandom sequences, and the estimated value for reception of the same sequence. Unlike FSFB/2, this protocol allows the sending of scalar data, such as multi-value controls, in addition to Boolean data, which does not necessarily need to be sent in different processing cycles.

2.4.3 Safe Response time analysis

The safe response time analysis is a study conducted on architectures where protocols like FSFB2 and PVS are used to assess system response, such as the actuation of commands sent from the control room where interlocking is installed to the peripherical nodes, and the acknowledgement of their actual implementation through the receipt of controls. Since both protocols perform the same function between the control room and the peripheral nodes, their communications are characterized by similar response times. The most significant factor in this evaluation analysis is the time required for information to be transmitted through a wired and private network used as the data communication system for signalling purposes.

The parameters required to compute the information exchange between the two nodes connected through the network are as follows:

- ✓ Network crossing time (T_{NET}): It is defined as network transmit time as the time between the instant when the transmitter node is prepared to send information on the network and the moment when the receiver node can utilize the received data. This time encompasses not only the time required to propagate the information on physical media but also the time needed for a node to obtain access to the media.
- ✓ Cycle time: As the equipment under analysis comprises cyclic machines, every local equipment generates information to be transmitted once T_{cl} per cycle, while every remote node consumes the information once T_{cr} per cycle.
- ✓ Survival time: It represents a configuration parameter expressed in local node cycles Y, used to maintain the validity of previously received data when no further messages have been received from the network.
- ✓ SSR-delay: Specific of the FSFB2, it takes into consideration the required synchronization time.
- ✓ Sleeping delay: This is a parameter used to countermeasure hardware fault on the network.

The definition of survival time, which obviously affects the system's response time evaluation, is not solely due to potential network gaps caused by faults but is also a characteristic of the operating principle of cyclic machines. For instance, in a communication between a transmitter operating with a cycle of 500 milliseconds and a receiver node operating at 250 milliseconds, the minimum value for survival time must be configured to enable correct behaviour in the information exchange, holding the old value for over 250 milliseconds. In more general sense and for safety purpose, the survival time, which maintains active the previous status in the local node, must be taken into consideration when timing calculations are more restrictive than the value maintained within the receiver. The steps involved in computing the safe response time in a scenario where a remote node, such an object controller (e-SMIO), needs to acquire a restrictive value to be transferred to a local node represented by an interlocking fitted in the control room, assuming the survival time is configured as one, can be outlined as below:

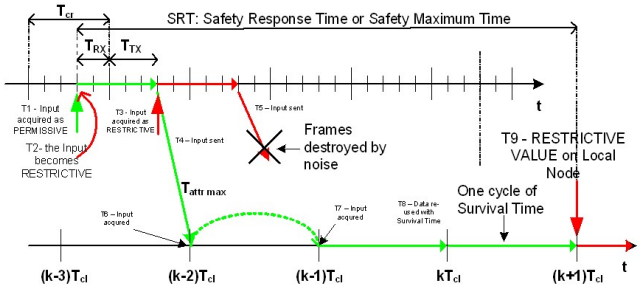


Figure 6 – Safe Response time flow computation

where:

- ✓ T1: The remote node acquires the input as permissive.
- ✓ T2: It is immediate after T1, the input becomes restrictive.
- ✓ T3: The input is acquired as restrictive by the remote node.
- ✓ T4: The input acquired in T1 (permissive) is now sent on network addressed to local node.
- ✓ T5: The input acquired in T3 (restrictive) is now sent on network addressed to local node, but it will be never received.
- ✓ T6: The input acquired in T1 by remote node is received by the local node immediately after its input acquisition, then it will be used in the next cycle (T7).
- ✓ T7: The input acquired in T1 by remote node is used by the local node.
- ✓ T8: No data has been received but being the Survival Time configured to 1 the information used in T7 are now reused.
- ✓ T9: No data received, the Survival Time has been expired, then the restrictive value is selected.

This leads to generalize the computation of the safety response time as below:

$$SRT = T_{cr} + T_{NET} = (T_{RX} + T_{TX} + T_{cr}) + T_{cross\text{-}max} + (2+Y)\, T_{cl} - T_{cr} =$$

$$= T_{RX} + T_{TX} + T_{cross\text{-}max} + (2+Y)\, T_{cl}$$

In the case of an architecture based on the DCS-SIG, which comprises two separate paths (red & blue), and redundancy management is carried out within each individual host by transmitting data on both paths, the transfer delay is independent of the protocol type used but is instead determined by the network crossing time, which is assumed to range between 10 milliseconds and 50 milliseconds.

Similar analysis will be conducted in the upcoming chapters, adopting an architecture based on public network for the sake of comparison and preliminary study to determine the service level agreement to request to the telecom provider and to implement a distributed architecture with part of the intelligence on the peripheral side to reduce network crossing time.

3 Chapter three: Railway system over public network

3.1 First topic of the research

Communication technologies have always been crucial in railway environments to manage the flow of information among trackside and onboard subsystems fitted in railway infrastructure systems. The constant evolution of telecommunications technologies, encompassing both wired and wireless infrastructures, is a fundamental aspect of research for companies operating in the railway industry to improve efficiency through more precise and detailed information and to enhance safety and reliability of rail transport. As mentioned in the previous chapter, the communication between trackside equipment located in different zones of a railway line is currently handled through a wired and private network. Differently, the flow of information exchanged between the control room and onboard subsystems, such as the train position report and the movement authority to be safety respected, is facilitated through wireless communication using GSM-R. The Global System for Mobile Communications (GSM) is a worldwide standard for wireless mobile communication developed by the European Telecommunications Standards Institute (ETSI). To provide additional functionality for railway communication, ERA (European Union Agency for Railways) and UIC (International Union of Railways) introduced GSM-R, which has become one of the most widely used digital wireless technologies to provide mobile train radio communication services.

However, with the emergence of communication and internet revolution in various domains, the railways cannot lag behind in utilizing advanced wireless technologies to enhance railway operations and improve the user experience. Although GSM-R has successfully served the needs of railways for voice and messaging, it is a circuit switched technology that will be phased out in 2030 and has outlived its utility due to the increasing data driven needs of railways.

Bearing on this consideration, several Railway organizations worldwide are exploring the adoption of advanced technologies to meet present and future requirements. The next generation of integrated wireless networks for railways must not only ensure safe train operation but also provide advanced railway services for the future. This new technology will be regulated as an open standard to ensure wider availability of products, guaranteed minimum functionality and performance, and a longer lifespan of the technology. Additionally, the technology will align with global trends to leverage the global scale of railway operations.

The standardization process will support high-speed operations, requiring mobile voice and data communication for speeds exceeding 250 kph. The future of railway communication will involve video transmission functions, such as real time monitoring of passenger and/or coach status, necessitating a wireless technology capable of supporting real time transfer of large amounts of data. The evolution of communication technology must adhere to latency constraints, which vary based on the application. For example, in the European Train Control System (ETCS) Level 2, the maximum end-to-end communication latency for train control is specified as 500 milliseconds. As train speeds increase in the future, shorter communication delay times will be required. In emergency situations, voice call setup and connection should be completed in less than one second.

The implementation of the new standard in railway applications will result in improved network reliability, specifically regarding the reliable transmission of data information for safe operation, availability (indicating the percentage of time the network is fully functional and operational), and quality of service, which provides different priority levels depending on the service application. Critical applications are essential for train movements and safety and a legal obligation, such as emergency communications, shunting, presence, and trackside maintenance. Performance applications help to enhance the performance of railway operations, such as train departure and telemetry, while business applications support the railway business operation in general, such as wireless internet.

The Future Railway Mobile Communication System (FRMCS) is a worldwide telecommunication system designed by the International Union of Railways (UIC) in collaboration with different stakeholders from the railway sector. It is intended to succeed GSM-R, while also serving as a crucial enabler for the digitalization of rail transport. This solution is expected to become the global standard for railway communications, utilizing mobile broadband ready technology to enhance safety and operational efficiency, support innovative passenger services, and accelerate digital transformation. The FRMCS will leverage features provided by standardized 5G technology and will utilize dedicated frequency spectrum, like the implementation of GSM-R in the past. The fundamental features of 5G, including high mobility, reliability, and low latency, enable services to be provided based on end-user needs through slicing functionality. The implementation of dedicated frequencies may make the new standard a private network solution that serves global railway networks.

The research presented in this industrial thesis, proposed to the Ministero dello Sviluppo Economico (MISE), is situated within the context of a scenario characterized by strong development in the coming years. The aim is to assess the potential use of current public LTE based telecommunication networks for railway

communication, given that the 5G standard is not yet standalone on the national territory and the FRMCS standard is still in the specification phase. The study focuses on the use of LTE technology in railway networks with low density traffic, to determine its feasibility in terms of realization and costs. Specifically, the study evaluates communication between the control room and peripheral nodes to manage the movement of a level crossing using radio frequency technology. Moreover, transitioning from a private network technology to a public radio frequency domain requires evaluation of the radio coverage levels serving the zones of interest, the potential use of protocols such as FSFB2 and PVS without modification, and the implementation of virtual private network (VPN) and cybersecurity countermeasures to safeguard information from external access.

3.2 Railway architecture based on LTE description

Rete Ferroviaria Italiana (RFI) has supplied an experimental railway section, currently served by a private wired network infrastructure, between Nettuno and Padiglione stations to directly test the feasibility of an LTE based solution in the field.

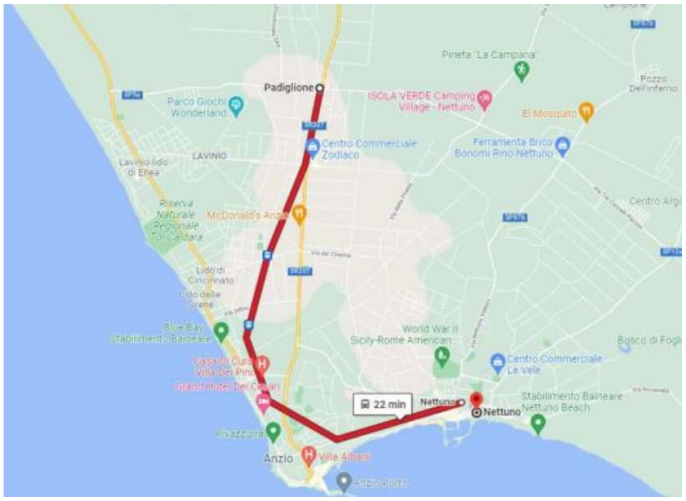


Figure 7 – Experimental track Padiglione - Nettuno

The experimental railway line section, spanning a length of 10 km, will consist of a duplicate of the control room in Roma Termini operating in parallel for testing purposes without disrupting the revenue service system. This control room will communicate with peripheral nodes equipped with object controllers through

radio frequency technology to manage trackside field equipment, such as level crossings, along the available experimental line. This part of railway section is designated as a low-density traffic line, in which train spacing can be achieved by detecting their position using the GPS system. Communication between the train and control room will occur over public LTE infrastructure, replacing the current GSM-R technology used for similar communications, in line with the proposed objectives.

As previously mentioned, utilizing a public LTE network offers the advantage of using an existing radio infrastructure managed by a public operator, thus avoiding the additional cost of constructing a proprietary network like GSM-R. However, this approach also presents potential challenges, including cybersecurity concerns, radio link performance, and availability of signal coverage.

The purpose of this paragraph is to illustrate changes in the architecture of the railway system when utilizing the existing public network to manage signalling and maintenance traffic, in line with the objectives of the industrial research. To establish communication based on an LTE infrastructure, the data communication system consisting of the redundant, fibre based private signalling network discussed in the previous chapter must be replaced by the current public network. In this type of architecture, where bandwidth and radio resources are shared by other users requesting network services, object controllers located at peripheral sites are connected to the public network via LTE 4G routers, each equipped with a SIM card enabling data connections through radio to the central site. To ensure complete redundancy management, different internet service providers must be used to mitigate temporary unavailability of radio resources for a service provider in a specific time or zone due to varying or poor signal coverage levels.

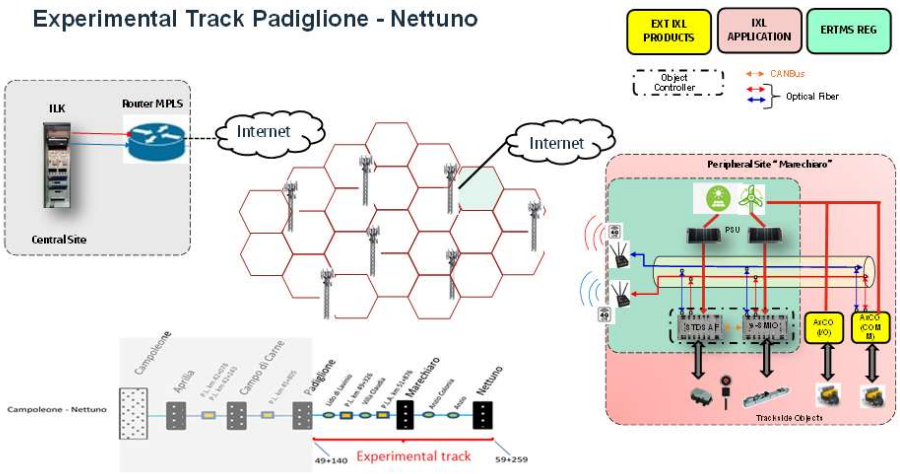


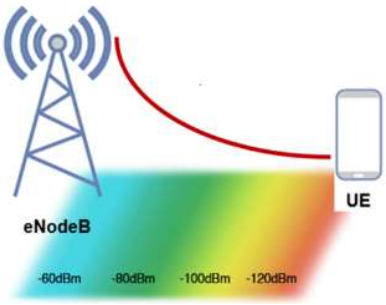
Figure 8 – Experimental track over radio frequency media

The illustration above depicts a hypothetical scenario where a peripheral node situated on a railway line communicates with the core network through a redundant radio frequency technology. On the other side, the control room accesses the internet via an MPLS-CDA infrastructure utilizing a wired private network. Multiprotocol Label Switching is a networking technology used to manage traffic over private wide area networks, that employs "labels" to determine the shortest path for traffic forwarding, rather than network addresses. While the control room has continuous access to the network due to its use of a private network, the peripheral node operates differently. In this hypothetical scenario, a user equipment located at the peripheral node needs to send a RACH (Random Access Channel) message to the network provider to request a connection service. Depending on the traffic already managed by the serving base station eNodeB and the radio resources available in its coverage area, the AGCH (Access Grant Channel) message may or may not allocate radio resources for internet access. This is because the network operates on a shared resource model, where all users requesting a service must share the available resources. A network utilizing LTE technology may encounter communication delays because of radio resource allocation by the telecom provider. To mitigate the impact of potential delays and reduce reliance on the public network, a distributed approach can be adopted in place of a centralized one. This approach involves establishing direct communication between two peripheral nodes equipped with e-SMIO, without routing the communication through a control room. However, this method requires more complex devices to manage the peripherical trackside equipment.

Apart from the challenge of allocating available radio resources, there exists a significant difference between communication based on a mobile network and a wired network. This difference is attributed to the strong dependence of radio frequency infrastructure on various factors such as received signal strength, interference, and modulation schemes that may not remain stable over time. These factors are also affected by the specific geographical topology where the communication node is located as well as the traffic conditions within the coverage area of the serving base station. Ensuring the feasibility of the research requires careful consideration of the placement of LTE routers and antennas. They must be installed at peripheral sites in a manner that guarantees a quality of service that matches or exceeds that available on a wired network. It is also necessary to implement fallback mechanisms that enable switching to 3G/2G connections in cases where network resources are degraded, thereby avoiding interruptions in the communication flow. The following metrics must be observed to determine the appropriate quality level for radio frequency technology:

- ✓ Reference Signal Received Power (RSRP): In an LTE cell network, the received power level can be measured using the average power metric. This metric is obtained by measuring the power received from a single reference signal.
- ✓ Reference Signal Received Quality (RSRQ): It indicates quality of the received signal.
- ✓ Received Signal Strength Indicator (RSSI): The signal strength is a metric that measures the device's ability to detect a signal from an access point or router. This parameter is helpful in assessing whether there is sufficient signal strength to establish a reliable wireless connection.
- ✓ Signal to Interference plus Noise Ratio (SINR): It is defined by dividing the power of a specific signal by the combined power of interference from all other signals and the background noise.

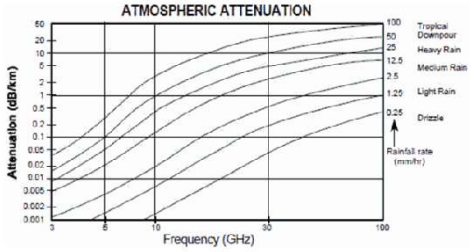
The parameters affecting a mobile network's coverage from a base station can vary over both space and time. This variability can lead to boundary conditions such as traffic, interference, and transmission delays caused by multiple propagations resulting from reflection and refraction phenomena. The graph below illustrates the physical behaviour of signal strength (red line), which is significantly impacted by the distance between the mobile terminal and the serving base station. As a result, the quality and performance of the network tend to deteriorate towards the cell's edge radio frequency condition.



RF conditions	RSRP (dBm)	RSRQ (dB)	SINR (dB)
Excellent	≥-80	≥-10	≥20
Good	-80 to -90	-10 to -15	13 to 20
Mid Cell	-90 to -100	-15 to -20	0 to 13
Cell Edge	≤-100	<-20	≤0

Figure 9 – LTE signals strength and quality values

The signal strength can experience a reduction due to attenuation phenomena, which may result from atmospheric conditions or the distance and presence of obstacles between the device and the serving base station. Atmospheric conditions such as rain can cause interference in communication, and the severity of this interference is often related to the frequency being used. Higher frequencies are more susceptible to interference caused by rain because the wavelength transmitted is similar in size to the raindrops, even though the coverage distance may be shorter. The atmospheric attenuation resulting from weather conditions significantly affects the Received Signal Strength Indicator (RSSI), which is another important parameter to consider when evaluating the feasibility and quality of LTE based radio frequency technology.



Power of received signal	RSSI (dBm)
Excellent	>-65
Good	-65 to -75
Sufficient	-75 to -85
Not sufficient	-85 to -95
No signal	=-95

Figure 10 – RSSI and atmospheric attenuation effect

This phenomenon of interference, known as attenuation, is particularly relevant when using millimetre waves and is not as noticeable in communication using frequencies below 10 GHz, such as in LTE. Measuring radio signal strength enables the determination of the type and location for antenna installation to achieve optimal coverage levels without having to install new radio base stations, as previously mentioned. However, this also requires conducting a preliminary test campaign to assess the FSFB/2 and PVS protocols' robustness in various environmental conditions when using radio frequency technology. The protocol's performance must be assessed under various operational conditions, including the introduction of a virtual private network that encrypts messages transmitted using a secret key exchanged between the control room and the peripheral node.

This is necessary because the communication relies on a public LTE network. Additionally, implementing a private Access Point Name (APN) managed by the telecom provider and connected to the central MPLS router to which SIMs associated with each peripheral node on the line will register is another aspect that must be considered. This approach can help achieve an additional level of security in the communication flow.

It is crucial to consider cybersecurity measures when relying on a public LTE network, where all radio resources are shared among users. To protect the system from digital attacks, data breaches, and other types of cyber threats that may originate from internal or external sources, a cybersecurity platform and policies must be established. This leads to introduce firewall devices into the architecture and implementing the 802.1x protocol for MAC authentication filtering.

Taking these considerations into account, the overall system architecture for managing a control room and two peripheral nodes via a public LTE network can vary between centralized and distributed architecture types, depending on the needs. In this context, introducing cybersecurity devices at the central side is necessary, as depicted in the image below.

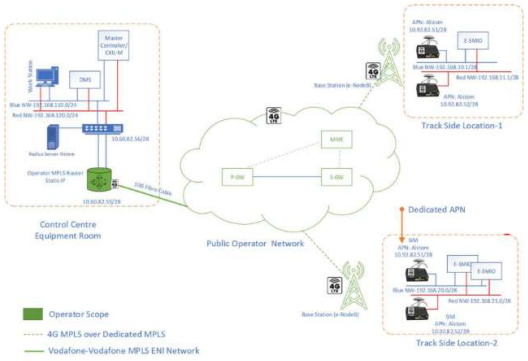


Figure 11 – Railway architecture based on LTE infrastructure

In summary, the various aspects discussed in this chapter that require careful consideration for research on the feasibility of serving a railway system using an LTE based infrastructure can be categorized as follows:

- ✓ Railways protocols adaptability and VPN functionality implementation to encrypt the communication.
- ✓ Introduction of smarter devices with internal logic at peripherical side to implement a distributed architecture.
- ✓ Coverage level and resource allocation.
- ✓ Introduction of cybersecurity policies to protect communication from external access.
- ✓ Definition of a service level agreement to be requested to the telecom provider to ensure that the quality of service meets the required throughput and performance standards appropriate for a railway system.

3.2.1 Proprietary signalling protocols for communication over LTE

As mentioned in the previous chapter, the FSFB/2 and PVS protocols are commonly used for signalling communication between cyclic machines that manage trackside equipment. These protocols were developed to exchange vital communication in compliance with a category 2 network type, which restricts unauthorized access. However, if these protocols are intended to be used in a category 3 network, they must be updated.

To achieve category 3 compliance as required by CENELEC 50129:2018, the solution is to utilize the FSFB/2 protocol at the application layer and the Datagram Transport Layer Security (DTLS) protocol at the transport

layer of the ISO/OSI model. However, this will necessitate an update to the kernel on the object controller devices to integrate this feature.

DTLS is a protocol designed to secure communications that use datagrams. It is based on the stream-based Transport Layer Security (TLS) protocol, which offers a similar level of security. However, since it is a datagram protocol, DTLS does not always guarantee the order of message delivery or the delivery of messages. Nonetheless, DTLS provides benefits associated with datagram protocols, including lower overhead and reduced latency. Due to these benefits, it is well suited for securing delay sensitive applications and services that use datagram transport, as well as for tunnelling applications such as VPNs.

In contrast, the PVS protocol is designed for both closed transmission systems (category 2 of EN-50159 regulation) and open transmission systems (category 3 of EN-50159 regulation), such as those based on LTE technology. In the latter case, it is necessary to enable the cryptographic and encapsulation techniques, such as AES (Advanced Encryption Standard) or AES-CMAC (Advanced Encryption Standard – Cipher - based Message Authentication Code) that are provided for by the PVS protocol itself.

The Advanced Encryption Standard (AES) is a popular encryption algorithm that employs the same key for both encrypting and decrypting data. AES-128 is a specific version of AES that uses a 128-bit key. Encryption involves utilizing intricate mathematical algorithms to transform plain text into ciphertext, which is incredibly challenging to reverse without the original key. Encryption is a common cybersecurity strategy employed to safeguard data transmission in both private and public networks. AES and the two proprietary protocols must be adopted together to enable their utilization in a category 3 network type.

In a radio frequency environment that uses a public network to exchange signalling data, where the same infrastructure handles the transport of traffic for numerous users outside the railway sector, all radio resources are shared. This creates a significant vulnerability that increases the risk of malicious internal or external intrusions. To ensure the secure and safe transport of signalling data over the public LTE infrastructure, it is essential to implement protective measures that prevent unauthorized intrusions and safeguard against the modification of signalling data. In the absence of these protective measures, the railway system's safe management is at risk of suffering severe consequences. Therefore, in addition to enabling the use of signalling protocols mentioned above on category 3 networks through DTLS and AES encryption techniques, another level of security is introduced by incorporating firewall equipment and virtual private networks (VPN) between the central node and each peripheral node.

Firewalls are devices that regulate incoming traffic from external networks attempting to access internal networks that are protected by the firewall. These devices analyse all traffic passing through and use preconfigured Access Control Lists (ACLs) to determine whether to block or permit specific traffic. ACLs are rules that instruct the firewall to differentiate between authorized traffic that can flow from an external network to an internal network and all other traffic that is not permitted by default. ACLs can analyse a broad range of parameters, including source and destination IP addresses, socket port numbers, and the packet's protocol type when analysing network packets. Advanced firewalls can also examine the information content of network packets and identify viruses. On the other hand, a VPN is a software feature that enables the masking of private data transport within a public network. Specifically, a virtual private network transforms the communication systems of a generic public and untrusted network into a virtual private trusted network between two hosts, emulating the properties of a point-to-point private link. It is a computer network that provides online privacy to a user by establishing an encrypted connection from a device to a network. An IPSec VPN tunnel is a mechanism that allows devices on a private network to access the public network as if it was a point-to-point connection that is exclusively dedicated to them. This is accomplished by creating a virtual tunnel on the public network.

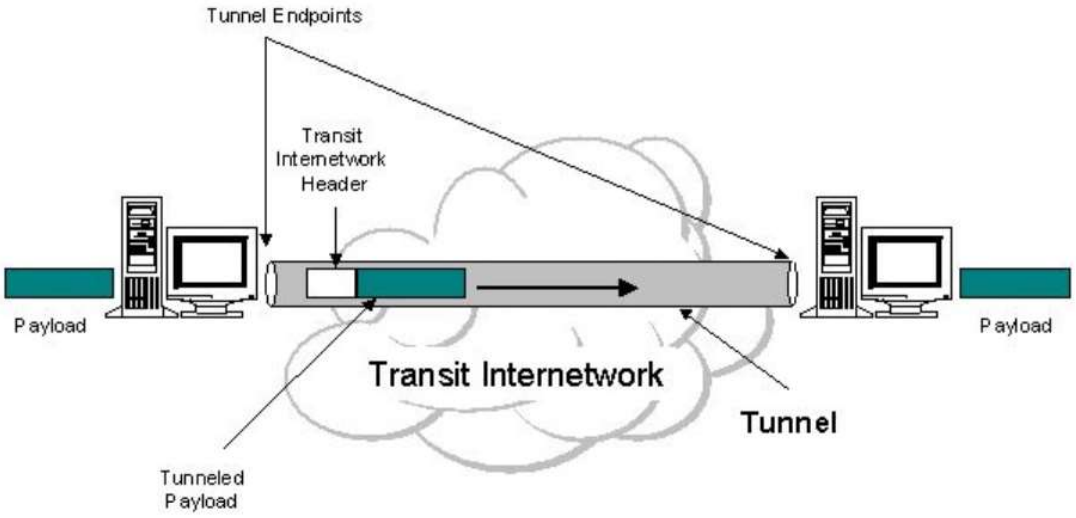


Figure 12 – Tunnelling concept

From the perspective of the public network, the tunnel is made up of the regular communication packet transport, which contains the data exchanged by the private network and is transported on the public network according to the addressing rules used by the public network. A VPN exchanges communication packets on the public network, which raises the issue of protecting the exchanged data against the risk of

unwanted access by unknown users who may gain access to the packets as they are transported on a public network. The VPN transports communication packets on the public network as normal packets, but these packets contain the data exchanged by the private network and are addressed according to the rules of the public network. However, since the VPN utilizes the public network to transfer these packets, there is a possibility of unauthorized access to the data by unknown users who may gain access to them as they are being transmitted on a public network.

To safeguard the data transmitted via a VPN on a public network from potential risks, the IPSec protocol is employed. This protocol operates at layer 3 of the ISO/OSI stack and guarantees the integrity and encryption of the data, as well as authenticated access solely for legitimate VPN users. IPSec VPNs employ well established and thoroughly tested encryption protocols such as DES, 3DES, AES, SHA-1, and MD5, which are defined in various RFCs, including RFC2401 (Security Architecture for the Internet Protocol). An IPSec VPN encrypts a complete data packet and encapsulates it within an outer packet that only discloses the IP addresses of the VPN termination points on the public transport channel. This conceals the private IP addresses (which are visible only to users within the VPN) and only discloses the IP addresses of the machines facing the public network where the VPN is terminated.

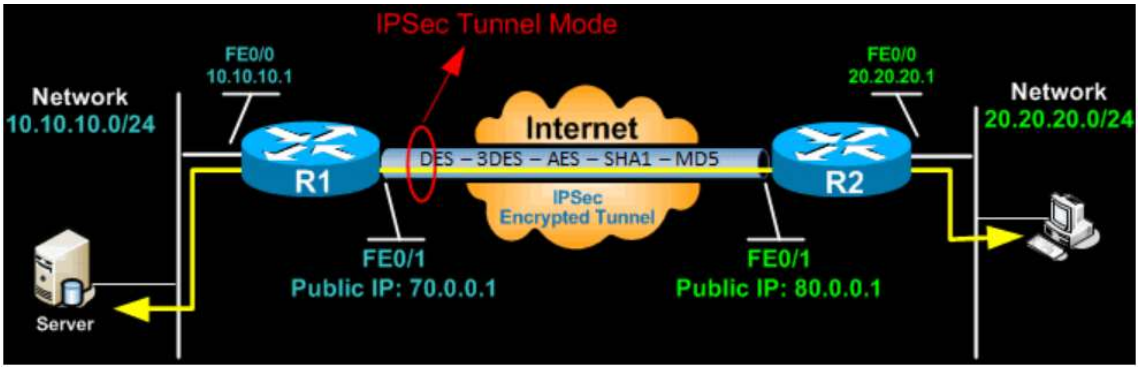


Figure 13 – IPSec Tunnel

IPsec employs protocols like MD5 and SHA1 to authenticate users who have permission to activate a VPN. MD5 is an algorithm based on hashing, featuring a 128-bit encryption key, while SHA1 uses the same technology to offer a 160-bit key. To transmit encryption and authentication keys through a public network, IPsec can utilize one of the following protocols: DH1, DH2, DH5 (all part of the Diffie-Helman group), and IKE (Internet Key Exchange). The use of a virtual tunnel between nodes offers an additional layer of security to the communication between two nodes over a radio medium. Moreover, it enables the transformation between dynamic public IP addresses allotted to a peripheral node with every network registration request,

and private static IP addresses configured within the railway devices to facilitate communication between the subsystems of the targeted signalling system.

Bearing this matter, there have been discussions with public providers about creating VPN tunnels with LTE technology. This functionality is crucial for segregating signalling traffic on LTE from other traffic on the public LTE infrastructure and necessitates SIM cards with static IP addresses. Public providers have indicated that obtaining SIMs with public static IP addresses is not possible, and that static IPs can only be acquired via SIMs registered to a private APN. These SIMs facilitate data transport on the public LTE infrastructure through provider created tunnels, isolating the traffic of private SIMs from public traffic on the same infrastructure. Static IP addresses may be configured on SIMs with private APNs, but these IPs are private and do not have access to the internet. To use this technology, the tunnel must be terminated at both ends: on the SIM side and at a common termination point shared by all SIMs with the same private APN. This shared termination point functions as a central hub for a communication system that allows point to point traffic between each SIM and the common termination point of the tunnel. In instances where two SIMs with the same private APN need to communicate, data packets are routed through the individual SIMs' tunnels' common termination point.

Considering the security requirements and available alternatives, the decision was made to utilize and evaluate a VPN to secure every connection between a signalling system's peripheral site and the central experimentation node, as well as between the train and the central experimentation node. The table below lists the selected parameter values for protecting the signalling data as it travels over the public telecom network.

Information	Values for Operational Network Conduit		
IKE parameters			
IKE version	IKEv2		
Encryption algorithm	AES-GCM		
Encryption key size	256 and higher		
Integrity algorithm	None		
Diffie-Hellman Group	DH 19 (ecp256) and higher or DH 14 (modp2048) and higher / PFS enabled		
PFR transform	sha512		
Authentication type	ECDSA >= 256 or RSA >= 2048		
Authentication method	X.509 certificate-based authentication or PSK with minimum 32 random characters		
Cookie challenging	Enabled / Threshold 25%		
IKEv2 SA rekeying	<8 hours		
ESP parameters			
ESP mode	Tunnel mode		
Encryption algorithm	AES-GCM	AES-GCM	None
Encryption key size	256 and higher	256 and higher	NA
Integrity algorithm	None	None	HMAC-SHA256
ESP SA rekeying	<8 hours	<8 hours	Not applicable

Table 1- VPN parameters chosen

Considering the architecture, the communication traffic has to be subjected to a dual tunnel mechanism and VPNs:

- ✓ A tunnel that is created and managed independently by the provider to implement the Private APN configuration.
- ✓ A VPN built and managed by Alstom to ensure controlled protection on the data of signalling without having to rely solely on the tunnel referred to in the previous point, carried out by the provider for another purpose.

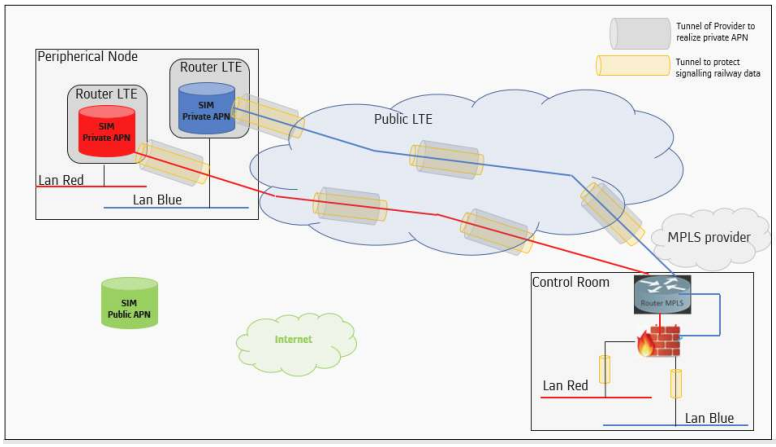


Figure 14 – VPN architecture implementation

3.2.2 Distributed architecture over LTE network

Information system architectures have evolved over time, transitioning from centralized structures to more distributed models that are better suited to handle decentralization and cooperation among the various subsystems that make up a railway signalling system. This change has been driven, in part, by advancements in telecommunication technologies and it can be of significant interest in computing the execution times of commands when transitioning from a private wired network to a public LTE infrastructure as in the case of the current industrial research.

A centralized computer system pertains to a system in which all data and applications are housed on a single processing node, as in the case of the infrastructure railway system communication described in chapter 1, where a centralized interlocking system at a central location manages the private wired network. On the contrary, a distributed computer system is one where applications that collaborate towards common goals are situated on multiple processing nodes (distributed processing), or if the unitary information assets are

hosted on several processing nodes (distributed database). A distributed system generally consists of a group of logically independent applications that communicate and collaborate through a hardware and software infrastructure in order to achieve shared objectives.

The main concept behind partially decentralizing the logic that was previously calculated entirely by the interlocking device installed in the control room is due to the utilization of a radio media shared by multiple users where the priority concept and allocation of radio resources are not guaranteed. This is achieved by adding computational calculations and increasing the complexity at the peripheral node level.

Transferring part of the intelligence from a central node to a peripheral zone, the software of a component must be modified to make it more advanced. As a result, the SMIO must be replaced with evolved versions. This enables the entire system to function more efficiently with reactive cyclic machines since the computational component has fewer Boolean rules to manage field equipment like turnouts or level crossings. By shifting the logic closer to the controlled devices, delays caused by a weak signal or limited radio resources can be reduced. This is due to the aim is to exploit a public network using LTE technology.

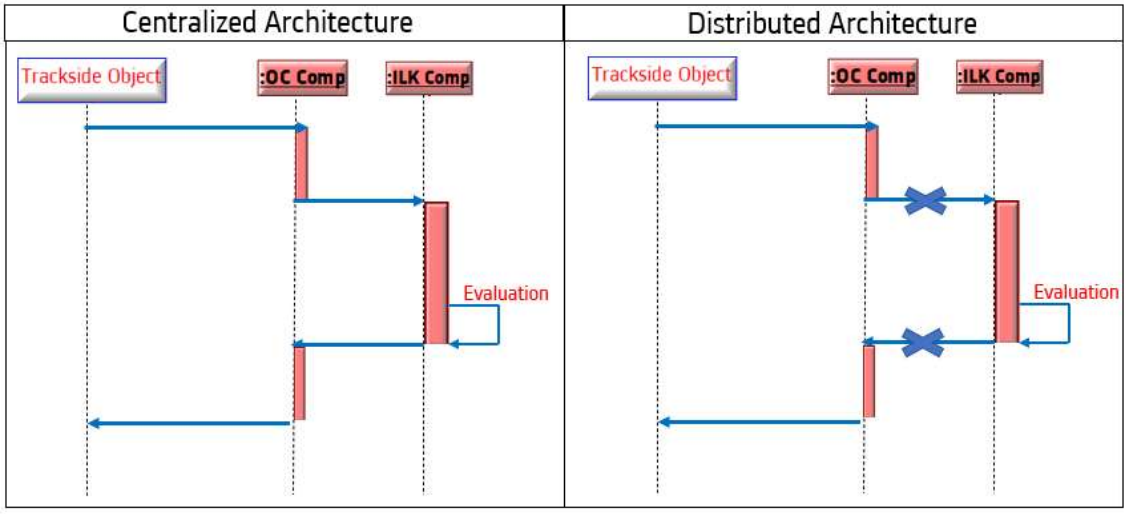


Figure 15 – Distributed architecture over LTE

The above image illustrates the contrast between the steps involved in achieving trackside to trackside object detection in a centralized architecture versus a distributed architecture. In the former case, to detect the status of trackside equipment, information must travel from the field to the peripheral node, where it is then forwarded to the interlocking system that contains the logic installed in the control room. After evaluation, the status is redirected back to the peripheral node, passing through the object controller.

On the other hand, employing a distributed architecture enables direct management of the state of trackside equipment at the peripheral site, eliminating the need to transmit information to the control room and reducing radio communication usage. This approach avoids radio access request issues and resource availability concerns. In conclusion, the hardware and software complexity introduced could be a solution that reduces radio medium usage, enabling a less stringent latency value request to the telecom provider during the definition of the service level agreement, as will be further explored in the current research thesis.

3.2.3 Coverage level and resource allocation

To effectively and reliably utilize a radio frequency based public network such as LTE for a railway or other system, two primary issues that must be addressed are received signal power and resource allocation. These factors determine the feasibility of utilizing this technology, as inadequate radio resources and inadequate coverage means that user equipment cannot request any network services or may not have access to the network at all. As mentioned in the introduction of this chapter, the aim is to leverage the existing radio infrastructure managed by a public operator, avoiding the need to install additional eNodeB along the track to be serviced in cases where the received signal power is poor and not in line with cost constraints. This led to the conclusion that an ERTMS low density system should only be installed using a public network if the area of interest is adequately covered. Track surveys were conducted on the experimental Padiglione-Nettuno track to determine the coverage level and assess the feasibility of utilizing mobile radio coverage for level crossing operations.

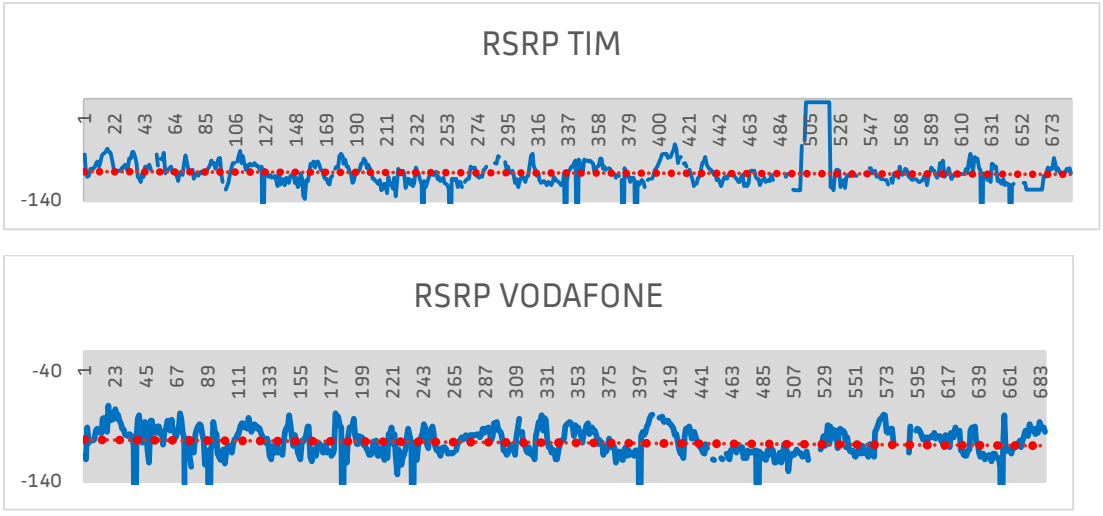


Figure 16 – RSRP evaluation on Padiglione Nettuno experimental track

The Reference Signal Received Power was recorded for both telecom providers selected for the project, along the entire experimental track, using mobile terminals equipped with the SIM of each provider and mounted on a train traveling on the track of interest. As shown in the image above, the survey resulted in an average RSRP level of -105 dBm for both providers. This value is not only an average but also a reference parameter, as it was measured artificially using unsophisticated mobile terminals mounted within the train, which resulted in additional attenuation due to the train carriage material. However, this value indicates that LTE technology can be used for communication between central and peripheral locations on the experimental track, with the constraint of directing the antenna of the LTE routers installed on the peripheral nodes appropriately.

Due to the variation in radio signal caused by environmental conditions, it became necessary to conduct a test campaign to evaluate the robustness of the FSFB/2 and PVS protocols on radio frequency technology under different coverage levels. The objective was to statistically determine the average value and distribution of communication parameters such as communication latency and percentage of connection loss, based on the minimum level of LTE signal power that guarantees uninterrupted exchange of signalling information between the two interacting nodes, without resulting in blockages or interruptions in a railway system. The results obtained from the campaign will serve as the basis for antenna installation in the field. Depending on the available radio coverage, the antennas will need to be positioned in a way that ensures receipt of a signal power equal to or greater than the minimum value required for continuous communication.

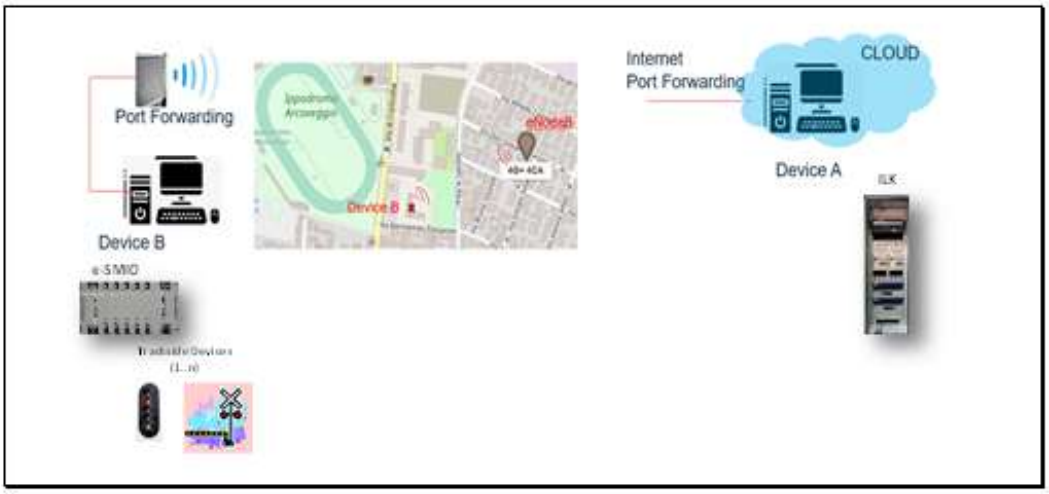


Figure 17 – Test Bench

To evaluate the behaviour of these protocols on the LTE network, the simulation environment depicted in the above figure was set up. A virtual machine on the cloud, with an associated communication simulator, was employed to represent the interlocking system installed in the control room, representing the characteristics of a wired network access. Furthermore, a physical workstation with an attached communication simulator, connected to an LTE router with an internal SIM, was installed to represent a typical peripheral node, with network access through a mobile infrastructure. The simulators were designed to be resilient to communication gaps lasting between 500 milliseconds and one second, to be aligned with the safe response time and preventing the railway system from becoming too slow in command execution.

The simplified test bench used in this evaluation did not include the implementation of redundancy on the LTE network or the MPLS CDA architecture with an associated private APN, as they were not yet available. Instead, a point-to-point VPN was implemented to determine the overhead introduced in the transmitted payload and the delays associated with evaluating the latency, which will be subsequently requested from the telecom provider during the service level agreement phase. Tests have been executed in both favourable field conditions (excellent and good), where the base station and peripheral node were relatively close, and there were no obstacles in between, as well as weak radio frequency condition has been simulated shielding the antenna to capture an attenuated signal. The communication of the protocols was not influenced by the received signal power level but rather by communication disruption in the network and frequency handovers, which have led to the normalization of the communication.

Since it was not possible to configure the simulators with delays exceeding one second to avoid compromising the system's safe response, this evaluation was conducted to request a quality-of-service indication from the telecom provider, which must be uninterrupted in 99% of cases. Additionally, a crucial aspect of being able to use a service is that the mobile terminal can register with the network, resulting in the allocation of radio resources by the serving radio base station.

Radio Resource Management (RRM), which involves the allocation and management of resources, is vital for telecom providers because they have a limited number of radio resources to manage efficiently to provide a minimum level of service to their customers. Telecom operators must pay for the use of these radio resources and must utilize them optimally to maximize their value. Therefore, one of the primary objectives of radio resource management (RRM) is the efficient allocation of radio access resources among users in time, frequency, and space to maximize spectral efficiency, user rate, and network throughput.

Radio access allocation is restricted by the transmission power of nodes and the interference they cause to each other, which can occur in both downlink (DL) and uplink (UL) modes. In DL, the reception of a mobile node may be interfered by transmissions from neighbouring access nodes, while in UL, the reception of an access node may be interfered by transmissions from other mobile nodes connected to neighbouring access nodes.

As a result, there is a rise in spectral efficiency and user rate when transmit power increases and interference decreases. However, an increase in transmit power between an access node and a mobile node in the downlink leads to an increase in interference on mobile nodes connected to neighbouring access nodes. To maintain low interference and maximize Network Spectrum Efficiency, one approach is to allocate each user a unique set of independent, noninterfering, orthogonal, and uncorrelated resources. The Network Spectrum Efficiency is the parameter used by the telecom provider to assess how effectively radio resources are utilized in the network. Telecom providers pay a fee to rent radio resources from National Authorities, and their goal is to make optimal use of these resources. Unlike Link Spectrum Efficiency, Network Spectrum Efficiency is defined as the ratio of the desired bit rate to the channel bandwidth.

Defining the following features:

- ✓ N: Number of radio resources unit available to the system.
- ✓ k: Number of cells per cluster.
- ✓ n = N/K: Number of radio resources unit allocated to each cell.
- ✓ B_t: Total bandwidth available [MHz].
- ✓ $B_c = B_t/N$: Equivalent bandwidth per radio resource unit [MHz].
- ✓ A: Cell area [square x Km].
- ✓ R_b: Minimum bit rate assigned to each radio resource unit.
- ✓ E: Carried traffic within a cell [Erlang/cell].
- ✓ E = E/n: Carried traffic within a cell per radio resource unit [Erlang/cell/RU].

The Link Spectrum Efficiency can be evaluated as the ratio between the minimum bit rate assigned to each radio resource unit and the equivalent bandwidth per radio resource unit, measuring the efficiency of modulation and coding scheme.

$$\eta_{LINK} = \frac{R_b}{B_c} \text{ [bit/sec/Hz]}$$

As the link becomes more complex with advanced coding techniques, the Link Spectrum Efficiency improves, but it does not consider the effect of interference on radio resources. Increasing the complexity of modulation and coding techniques makes the link more susceptible to interference, resulting in inefficiency in a network where radio resources need to be reused over longer distances. Network Spectrum Efficiency accounts for this trade-off between the desire for highly efficient links and the signal to noise ratio (SNR), as well as the need for strong reuse of radio resources over a distance to increase network efficiency.

There are many definitions in literature to define the Network Spectrum Efficiency accounting the efficiency of spectrum usage with reuse. In 1982, when only voice applications were used, the Network Spectrum Efficiency was defined by Hatfield as:

$$\eta_{NET} = \frac{\mathrm{E}}{A\,B_t} \text{ [Earl/MHz/sqKM]}$$

Network Spectrum Efficiency is defined in such a way that if a given cell serves a significant amount of traffic E for a particular bandwidth, the efficiency outcome will be higher. The denominator includes B_t because the more bandwidth is used, the higher the cost for telecom providers to rent radio resources from the National Authority, resulting in a lower Network Spectrum Efficiency for a given carried traffic. The area covered and served is also included in the denominator to consider the fact that smaller cell sizes require a larger number of cells to be deployed to serve the area of interest, resulting in better network capacity. Bearing on the above considerations the Network Spectrum Efficiency can be expressed as function of the Link Spectrum Efficiency as below:

$$\eta_{\mathrm{NET}} = \frac{\mathrm{E}}{\mathrm{A\,B_t}} = \frac{\mathrm{E}}{\mathrm{A\,N\,B_c}} = \frac{\eta_{\mathrm{LINK}}\,\mathrm{E}}{\mathrm{A\,n\,k\,R_b}} = \eta_{\mathrm{LINK}}\frac{\mathrm{e}}{\mathrm{A\,k\,R_b}} \text{ [Earl/MHz/sqKM]}$$

Based on the proportional formula above, it may seem that having the highest possible Link Spectrum Efficiency leads to a high Network Spectrum Efficiency, but this is not entirely accurate. As mentioned earlier, Link Spectrum Efficiency does not factor in interference impairments and cluster size. Thus, the more links that are susceptible to interference, the larger the cluster size k must be, resulting in a lower Network Spectrum Efficiency. This concept can be applied generically from voice applications to data applications, such as the transmission of data for a railway application between the central side and peripheral nodes. Instead of the carried traffic within a cell E, the carried throughput in a cell is used to generalize the concept.

Defining the hereby features:

- ✓ S: Carried throughput within a cell [bits/s/cell].
- ✓ s = S/N: Carried throughput within cell per radio resource units [bit/s/cell/RU].

The Network Spectrum Efficiency can be evaluated as following:

$$\eta_{\mathrm{NET}} = \frac{\mathrm{S}}{\mathrm{A\,B_t}} = \frac{\mathrm{s}}{\mathrm{A\,k\,B_c}} = \text{[bit/s/MHz/sqKM]}$$

where the amount of user throughput in the link, denoted by s, can be obtained when one radio resource unit is assigned to the link, with an equivalent channel bandwidth Bc. The maximum carried throughput per radio resource unit can be estimated using the Shannon formula.

$$s = B_c log_2(1 + SNR)$$

The objective is to introduce the role of interference, and thus, for simplicity, the Shannon formula can be approximated by replacing the signal to noise ratio (SNR) with the ratio between the received power for the useful signal (C) and the sum of the received power from the interference signals (I). According to the definition of reuse distance, all first-tier cells are equidistant from the reference cell. To illustrate this, the coverage area is depicted below in a hexagonal layout, with two clusters represented in grey, each containing cells that belong to different clusters and reuse the same radio resource at a distance of D.

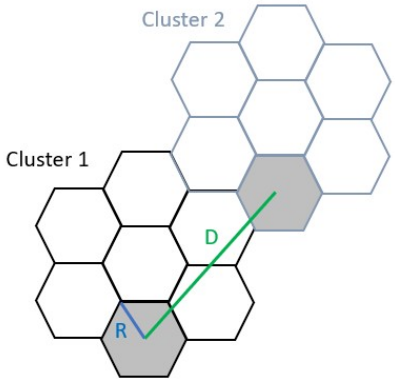


Figure 18 – Clustering and Reuse distance for radio resource unit

For simplicity, all interferers are assumed to provide the same level of received power, neglecting fading, and shadowing impairments and considering only the median component of the received power, which is distance dependent. This implies that, in the example above, the same interference level is provided by the six interferers belonging to the first tier. More generally, if there are N_{int} interferers, the interference power will be equal to N_{int} times the interference power from one individual interference cell. Considering the worst-case scenario from the perspective of the useful user equipment leads to the assumption that the

useful distance is provided by the radius of the cell. As a result, the useful received power at the receiver side can be expressed as:

$$C = P_r = c\, R^{-\beta}$$

where β represents the propagation exponent depending by the medium through which the communication shall travel. The interference received power is given by the sum of the power of the N_{int} interferers. Assuming that all the interferers are in the centre of each interferer cells at a distance D, the overall level of interference received might be expressed as below:

$$I = N_{int}\, c\, D^{-\beta}$$

Considering that the reuse factor defined as the ratio between the radius of the serving cell and the distance between two cells adopting the same radio resource unit in a hexagonal layout is given by $\sqrt{3\,k}^{\beta}$, the overall Network Spectrum Efficiency can be formulated as following:

$$\eta_{\mathrm{NET}} = \frac{\mathrm{s}}{\mathrm{A\,k\,B_c}} = \frac{log_2(1+\frac{C}{I})}{\mathrm{k\,A}} = \frac{log_2(1+\frac{(3\,k)^{\beta/2}}{N_{int}})}{\mathrm{k\,A}}$$

Telecom providers aim to maximize the above approximated expression, which has a significant dependence on coverage and cluster size. The cluster size k is included in the denominator to account for the disadvantage of having a larger cluster, which means a farther reuse of the same radio resource units. This results in fewer radio resources per cell, and hence, a lower Network Spectrum Efficiency. However, the cluster size is also included in the numerator, and thus, it plays a dual role. In conclusion, the Network Spectrum Efficiency is maximum when the cluster size is as small as possible, up to reaching the unitary lower bound.

The concept of dedicating one or more radio resource units moves in the opposite direction of maximizing Network Spectrum Efficiency, which goes against the policy of telecom providers due to the limited number of available resources in the spectrum, as mentioned earlier. It implies addressing a service from a public concept to a private network where there is no sharing at all.

Based on this concept, the GSM-R (Global System for Mobile Communication - Railway) standard has been widely adopted for railway communication. It involves the use of a dedicated network that has been made private by extending 4 MHz paired, of dedicated frequencies near GSM in both uplink and downlink (876-880/921-925 MHz).

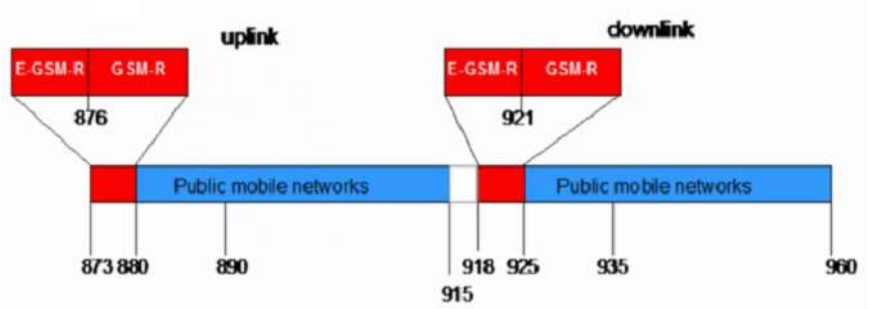


Figure 19 – GSM-R frequency spectrum

GSM-R is a sort of evolution of the 2G radio technologies dedicated for the use of a private network in railway scenario and is characterized by the FTDMA (Frequency-Time Division Multiple Access) in both uplink and downlink communication channels. The notion of radio resource unit in this kind of technology is equivalent to the physical channel. The FTDMA multiple access scheme might be schematized as a plane depicted in the below example, to better understand the concept of radio resource unit.

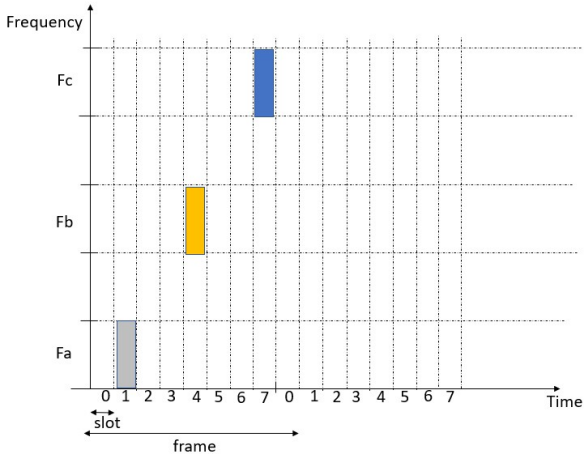


Figure 20 – FTDMA frequency allocation

For the sake of simplicity, the picture above illustrates the downlink channel, while the uplink channel behaves similarly but with a time synchronization shift of three slots. According to GSM technology, the time axis is divided into frames of 4.6 milliseconds, with each frame subdivided into 8 slots. The different channels or radio resources assigned to the cell are represented on the frequency axis. For example, in a cell with three available nonadjacent channels to prevent adjacent channel interference, the set of radio resource units available in the cell is given by the three bands multiplied by 8 slots per frame. The picture above

depicts different radio resources allocated to three different users who have requested a service from the cell where they are registered and located. This paragraph highlights the importance of coverage level and radio resource availability for a reliable network. To leverage a public network, such as LTE technology, where resources are shared among all users requesting a service, it is essential to assign radio resources following a request from the mobile terminal and prioritize the definition of a service level agreement.

In this paragraph has been illustrated how the coverage level and the radio resource availability is fundamental to have a reliable network, having to take advantage of a public network such as LTE technology, where resources, unlike GSM-R, are shared among all the users requesting a service, a very important aspect lies in the assignment of radio resources following a request from the mobile terminal and to the priority concept in the definition of a service level agreement.

3.2.4 Cybersecurity feature handling on a public network

The protection of hardware, software, and data from internal or external cyberthreats on internet connected systems is known as cybersecurity. Individuals and enterprises employ this practice to prevent unauthorized access to data centres and other computerized systems. The importance of this topic continues to increase each day due to the growing number of users, devices, and programs in modern enterprises, as well as the influx of data, a significant portion of which is sensitive or confidential.

In addition, cybersecurity faces ongoing challenges posed by hackers, data loss, privacy concerns, risk management, and evolving cybersecurity strategies. Given that the number of cyberattacks is not anticipated to decrease soon, the importance of cybersecurity continues to grow. Furthermore, the rise of the Internet of Things (IoT) and the resulting increase in entry points for attacks further emphasize the need to secure networks and devices across various application fields, including the industrial research domain being discussed. Indeed, one of the most challenging aspects of cybersecurity is the constantly evolving nature of security risks, which requires continuous updates to protect against potential new vulnerabilities. As new technologies emerge and are used in novel or different ways, new avenues for attack are discovered. Staying abreast of these frequent changes and advances in attacks, and updating practices to protect against them, can be difficult and is a crucial aspect to consider when commissioning a system.

In a cybersecurity framework, the CIA (Confidentiality, Integrity, Availability) triad is commonly used as a model for developing security systems. The CIA triad encompasses the three primary principles of information security: confidentiality, integrity, and availability. These principles are essential for the optimal functioning of a business and can serve as a roadmap for security teams to prioritize and address potential threats. When all three principles are adequately implemented, the organization's security posture is strengthened, and it is better equipped to handle security incidents.

Figure 21 – Cybersecurity CIA triad

Confidentiality pertains to an organization's efforts to ensure that data is kept private and secure. To achieve this, access to information must be regulated to prevent unauthorized sharing of data, whether intentional or accidental. A crucial aspect of maintaining confidentiality is to prevent individuals without proper authorization from accessing assets critical to the system. Conversely, an effective system also guarantees that those who require access have the necessary privileges. However, confidentiality breaches are not always intentional. Human error or insufficient security controls may also be to blame. For instance, a user may fail to protect their password to a workstation or restricted area. Users may share their credentials with someone else, or they may allow someone to see their login details as they enter them. In other situations, a user may not encrypt communication properly, enabling attackers to intercept their information. Additionally, a thief may steal hardware, such as an entire computer or a device used in the login process and use it to access confidential information. To reduce confidentiality breaches, one may classify and label restricted data, implement access control policies, encrypt data, and utilize multi factor authentication (MFA) systems.

Data integrity pertains to the accuracy and trustworthiness of information, which can be intentionally or accidentally compromised. Examples of intentional tampering include bypassing security systems or altering

logs to conceal an attack, whereas accidental violations may arise due to human error or insufficient security measures. To safeguard data integrity, techniques such as hashing, encryption, digital certificates, and digital signatures can be employed. Nonrepudiation is a means of verifying the integrity of information, such as using digital signatures to ensure that the sender of an email cannot deny sending it, and the recipient cannot deny receiving it.

Availability pertains to the ability to access data as and when required. This necessitates that systems, networks, and applications be functioning correctly and that individuals with appropriate permissions can access the data promptly. Various factors such as power outages, natural disasters, or deliberate acts of sabotage such as DoS (denial of service) attacks or ransomware can compromise the availability of data. To ensure availability, organizations can employ redundant systems, stay current with software and security updates, and implement backup and disaster recovery plans. These measures can help to minimize the risk of disruptions and ensure that crucial information and applications are accessible when needed.

Considering the CIA triad model, cybersecurity techniques must be considered and implemented to ensure a reliable railway system and prevent any service disruptions or compromises to safety due to internal or external malicious threats, particularly when leveraging a public network where communications are transmitted through the air. The LTE architecture, like any other radio frequency technology, is controlled by the Mobile Network Operator (MNO) and cannot be managed as a third party. As a result, even though the LTE architecture offers certain security measures, such as cryptography, hardware security, and user equipment authentication, these security mechanisms are not considered part of the defence in depth strategy when dealing with a public network where resources, such as the transmission medium, are shared among all users.

From a cybersecurity standpoint, the primary objective of the first topic in the current industrial research is to establish secure wireless communication using public networks that meet the cybersecurity requirements between the Interlocking system located in the central site and the e-SMIO located in the peripheral site in compliance with the EN-50159 regulation.

As already anticipated in previous chapters, according to EN-50159 classification, public LTE network used by the system outcome of the feasibility in leveraging on current radio frequency technology to serve a communication railway system is considered category 3 network, consisting of systems which are not under the control of the designer, and where unauthorised access must be considered. The Security Level Target to be achieved is SL-T 3 for both network components such LTE routers and railway component as e-SMIO,

preventing the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with moderate resources, as specified by IACS (Industrial automation and control system) specific skills and moderate motivation [9].

Security Level	Means	Resources	Skills	Motivation
SL1	Casual or coincidental violation			
SL2	Simple	Low	Generic	Low
SL3	**Sophisticated**	**Moderate**	**IACS-specific**	**Moderate**
SL4	Sophisticated	Extended	IACS-specific	High

Table 2- IEC 62443 security levels

The cybersecurity analysis of the topic under investigation in the current industrial research has considered the following attacker profiles:

- ✓ State-related: State actors and intelligence agencies typically conduct attacks using professional teams that follow predefined attack calendars and methods. This type of attacker is distinguished by their capacity to carry out offensive operations over an extended period, with access to stable resources and procedures, and the ability to customize their tools and methods to suit the target's topology. Furthermore, these actors have access to resources that allow them to purchase or discover 0-day vulnerabilities, and some can infiltrate isolated networks to carry out successive attacks to reach a target or targets, such as by attacking the supply chain.
- ✓ Organised crime: Like cybercriminal organizations such as mafias, gangs, and outfits, online scams, ransom requests, ransomware attacks, and the use of botnets are common tactics. With the proliferation of readily available attack kits online, cybercriminals are carrying out more sophisticated and organized operations for the purposes of fraudulent or lucrative gains. Some of these actors have the resources to purchase or discover 0-day vulnerabilities.
- ✓ Terrorist: Like cyber-terrorists or cyber-militias, attacks aimed at destabilization and destruction are often carried out with determination, despite not being particularly sophisticated. These attacks may include denial of service attacks, which aim to make, for example, the emergency services of a hospital centre unavailable, or untimely shutdowns of energy production industrial systems. These actors may also exploit vulnerabilities of internet sites and engage in defacement.

- ✓ Amateur: Like script-kiddies or hackers with proficient IT knowledge, individuals motivated by the pursuit of social recognition, fun, or the challenge may engage in basic attacks using readily available attack kits online.
- ✓ Avenger: This attacker profile is guided by a strong sense of vengeance or injustice, such as an employee who was dismissed for serious misconduct, or a service provider who is dissatisfied following the nonrenewal of a contract. They are characterized by their determination and internal knowledge of the systems and organizational processes, which can make them formidable and give them significant power to cause harm.
- ✓ Pathological attacker: This attacker profile is motivated by either pathological or opportunistic tendencies, and sometimes by a desire for gain, such as an unfair competitor, dishonest client, scammer, or fraudster. In this case, the attacker may have a knowledge base in computing that leads them to attempt to compromise the target's information system, or they may use attack kits available online, or subcontract the IT attack by calling upon a specialized outfit. In some instances, the attacker may direct their attention towards an internal source, such as a discontented employee or unscrupulous service provider, and attempt to corrupt them.
- ✓ Specialised outfits: The "cyber-mercenary" profile is characterized by high technical IT capabilities and differs from script-kiddies, although they share a spirit of challenge and pursuit of recognition, with the added focus on financial gain. These groups may be organized as specialized outfits that offer legitimate hacking services. Experienced hackers of this type are often the ones responsible for designing and creating attack kits and tools that are available online, which can then be used "turnkey" by other groups of attackers, possibly for a fee. Their motivations are solely financial, with no other driving factors.

The evaluation of performance and safety aspects and scales was conducted based on the nature of the system's purpose in the industrial railway research and the types of attacks considered. The focus was not on financial, reputation, or compliance topics.

Impact	Low	Medium	High	Very High
Safety	No loss of life, minor injuries.	No loss of life, one or multiple major injuries (< 20 injuries).	Loss of 1 life, multiple major/severe injuries (> 20 injuries).	Major loss of life (> 1).
Performance	Unplanned disruption (for up to a day) on one route.	Unplanned disruption (for up to a week) on any one route or Up to a day on multiple	Unplanned disruption (for up to a week) on multiple routes (critical	All users experience prolonged and unplanned disruption to key routes. Access to major station

Impact	Low	Medium	High	Very High
		routes. No impact on most critical infrastructure.	infrastructure). Major area blocked (e.g.: major city < 8H).	facilities likely to be severely restricted. Major area blocked (e.g.: major city > 8H) or main infrastructure blocked during >1 week.
Financial	Up to €1.000.000 or Minimal financial impact < 0.2% annual turnover.	Between €1.000.001 and €5.000.000 or Up to 3% of annual turnover.	Between €5.000.000 and €50.000.000 or Up to 10% of annual turnover.	Over €50.000.000 or > 10% of the turnover.
Reputation	Adverse local stakeholder reaction.	Adverse local/regional media reports over a period. Localised stakeholder concern.	Significant local/regional reports. National media interest during several days (< 1week) creating public concern. Negative national stakeholder statements.	Extensive prolonged adverse national reporting (>1week) and public disputes with key stakeholders.
Compliance	Minor non-compliance to contracts, regulations, and legislation with Low penalties & liabilities.	Medium non-compliance to contracts, regulations, and legislation with low penalties & liabilities.	Restriction to operate. Major non-compliance to contracts, regulations, and legislation with penalties & liabilities.	Extensive non-compliance to contracts, regulations, and legislation with high penalties & liabilities. Loss of license to operate.

Table 3- Impact criteria

The preceding paragraphs have already addressed:

- ✓ The introduction of encryption techniques such as DTLS and AES for the use of the proprietary FSFB2 and PVS protocols in category 3 networks.
- ✓ The implementation of point-to-point tunnelling to protect information between the central station and peripheral.
- ✓ The implementation of a private APN to allow access only to registered SIMs.

Nevertheless, these implementations alone do not satisfy the constraints imposed by the CIA triad. Therefore, a cybersecurity platform (CSP) subsystem at the central site was necessary.

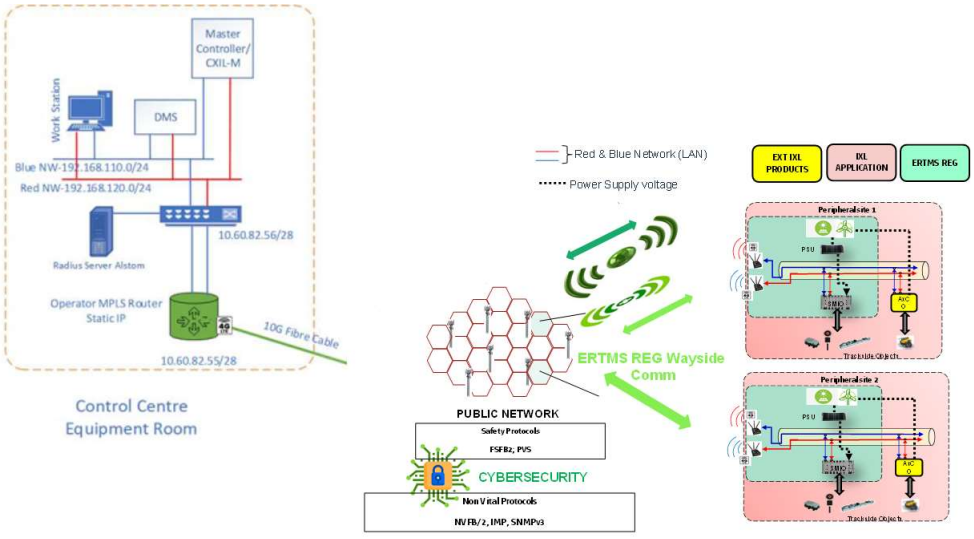


Figure 22 – CSP platform fitted at control room

Cyber Security Platform (CSP) is an Alstom solution integrating several services provided by open source and non open-source commercial components defined as COTS with the purpose to grant security functions allowing the protection of railways' subsystems against security threats. Services provided by this subsystem are virtualized to reduce hardware, energy and maintenance cost and space leading to handle with a flexible and scalable solution. Compliant with those requirements all services are not running on the same operating system environment, then the virtualization makes possible running several services on the same hardware equipment, using several operating systems. Additionally, when a service is obsolete or need to be changed by another one, virtualization makes easier service change by deleting and creating another virtual machine, without impacting the environment. The CSP features adopted with the main purpose to make the system robust against internal and external security threats on the network respecting certification and access policies can be categorized as following:

- ✓ Active Directory for managing user accounts and machine.
- ✓ SYSLOG Collector to gather logs from all signalling system.
- ✓ NTP server to manage the synchronization of all signalling system.
- ✓ Intrusion Detection System (IDS) to gather security events enabling on dedicated firewall devices the feature and filtering rules on data type, data content and user.
- ✓ RADIUS server to authenticate machine and some users.

The Active Directory services are managed by the Centralized Account Management Interface designed to provide Authentication and Directory Services such as user access rights management for all subsystems

and other CSP modules as software deployment and Domain Name System (DNS) which work in conjunction with Active Directory services. Active Directory Module is based on 2 redundant Domain Controllers (PDC+ADC) respectively primary and auxiliary, that will provide directory and authentication services to subsystems.

The Cybersecurity Platform's Syslog server module is responsible for collecting and storing system and user logs from subsystems and other CSP modules connected to the syslog interface. The collected logs are stored in a central repository using a standard protocol, allowing administrators to review the logs for troubleshooting and monitoring purposes. The client-server architecture of Syslog entails clients forwarding system and user logs to the Syslog server for monitoring and troubleshooting. The server listens on a registered port for protocol requests from clients. Syslog is used to gather network and cybersecurity events with a specific level of priority for all subsystems communicating in the railway system context, including the LTE router.

The Network Time Protocol (NTP) was selected primarily to synchronize the time of client or server equipment with reference time sources accurately connected to the network, utilizing the NTP protocol version 3 or 4 and a client-server interchange.

The Radius server module in the Cybersecurity Platform is designed to address the authentication, authorization, and accounting needs for various remote accesses. The Radius component, which operates on a client-server model, establishes conventions and rules for communication between network devices. In accordance with the client-server architecture, a Radius client (or supplicant) sends requests to a Radius server, which processes them and returns a response. The EAP-TLS (Extensible Authentication Protocol) manages the communication between the client and server and is a certificate-based authentication protocol. Authentication is conducted using a TLS handshake and requires mutual authentication through client side and server-side certificates. Therefore, a public key management infrastructure is necessary in the information system.

The Radius, in conjunction with the implementation of the IEEE 802.1X standard, forms the core of the cybersecurity policies based on the features described above in the CSP architecture and their respective roles. These policies aim to establish a MAC authentication protocol that allows devices access to the protected side of the network after successful authentication.

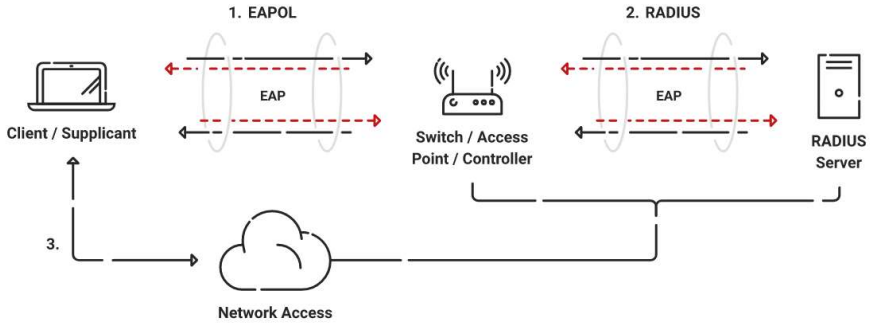


Figure 23 – 802.1x components

Only a few components are required to enable the 802.1X protocol, including a supplicant client, a switch enabled with MAC authentication protocol, an identity provider, and a Radius server.

For a device to participate in the 802.1X authentication, it must have a piece of software called a supplicant installed in the network stack. The supplicant is necessary as it will participate in the initial negotiation of the EAP transaction with the switch or controller and package up the user's credentials in a manner compliant with 802.1X. If a client does not have a supplicant, the EAP frames sent from the switch or controller will be ignored and the switch will not be able to authenticate. The switch or wireless controller plays an important role in the 802.1X transaction by acting as a 'broker' in the exchange. The client does not have network connectivity until there is a successful authentication, and the only communication is between the client and the switch in the 802.1X exchange. The switch/controller initiates the exchange by sending an EAPOL-Start packet to the client when the client connects to the network. The client's responses are forwarded to the correct Radius server based on the configuration in the Wireless Security Settings. When the authentication is complete, the switch/controller decides whether to authorize the device for network access based on the user's status and possibly the attributes contained in the Access_Accept packet sent from the Radius server. If the Radius server sends an Access_Accept packet because of an authentication, it may contain certain attributes that provide the switch with information on how to connect the device on the network. Common attributes will specify which VLAN to assign a user to, or possibly a set of ACLs (Access Control Lists) the user should be given once connected. This is commonly called "User Based Policy Assignment" as the Radius server is making the decision based on user credentials.

The Radius server acts as the "security guard" of the network. When users connect to the network, the Radius authenticates their identity and authorizes them for network use. A user becomes authorized for network access after enrolling for a certificate from the PKI (Private Key Infrastructure) or confirming their credentials. Each time the user connects, the Radius confirms they have the correct certificate or credentials

and prevents any unapproved users from accessing the network through a security mechanism of certificate validation. This guarantees that the user only connects to the network they intend to by configuring their device to confirm the identity of the Radius by checking the server certificate. If the certificate is not the one which the device is looking for, it will not send a certificate or credentials for authentication. At the end the identity provider refers to the entity in which usernames and passwords are stored. In most cases, this is Active Directory or potentially an LDAP server.

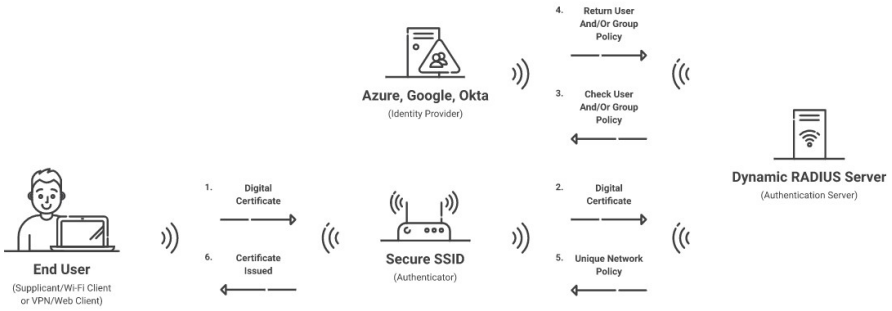


Figure 24 – 802.1x authentication flow

The 802.1X authentication process consists of the following 4 steps:

- ✓ Initialization: This step starts when the authenticator detects a new device and attempts to establish a connection. The authenticator port is set to an "unauthorized" state, meaning that only 802.1X traffic will be accepted and every other connection will be dropped.
- ✓ Initiation: The authenticator starts transmitting EAP-Requests to the new device, which then sends EAP responses back to the authenticator. The response usually contains a way to identify the new device. The authenticator received the EAP response and relays it to the authentication server in a RADIUS access request packet.
- ✓ Negotiation: Once the authentication server receives the request packet, it will respond with a Radius access challenge packet containing the approved EAP authentication method for the device. The authenticator will then pass on the challenge packet to the device to be authenticated.
- ✓ Authentication: When the EAP method is configured on the device, the authentication server will begin sending configuration profiles so the device will be authenticated. Once the process is complete, the port will be set to "authorized", and the device is configured to the 802.1X network.

3.2.5 LTE devices for radio frequency network access

In order to utilize the current LTE radio infrastructure, peripheral nodes must be connected through routers equipped with dedicated SIMs registered to the private APN, which must be requested from the telecom provider, and associated antennas to ensure communication with the serving radio base stations, provided that there is coverage in the area of interest. These devices must not only facilitate communication with the centralized control room via radio frequency technology but also be installed along the railway line, certified to operate in outdoor environments, and prevent interference with other railway devices and applications.

The hypothetical field installation should be positioned near the rails, as shown in the hereafter image. The e-SMIO rack, which manages the field equipment, will be housed in a cabinet along the railway line and connected to the LTE router and antenna. To ensure redundancy, the e-SMIO should be connected to two different LTE routers and antennas, each capable of registering with a different telecom provider. This configuration ensures more reliable communication in case of a failure or poor coverage level in one of the two radio frequency communication links.

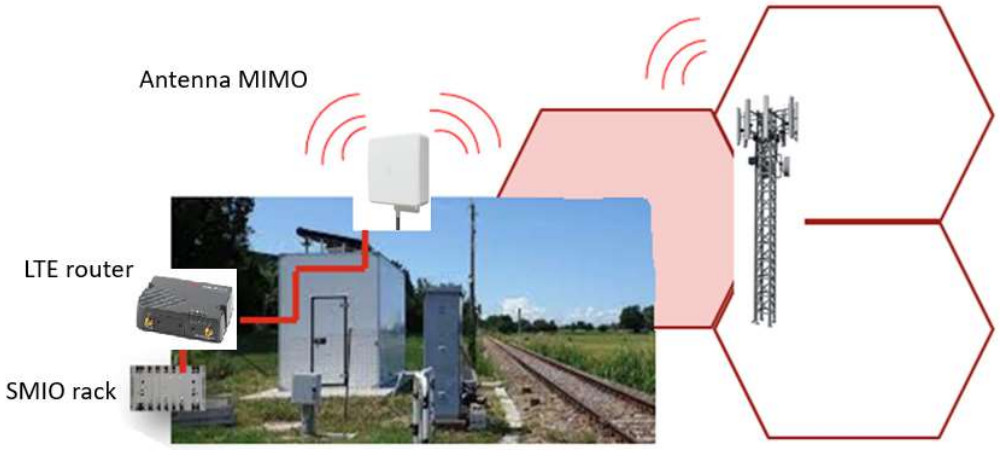


Figure 25 – LTE router installation on peripherical nodes

The Sierra Wireless RV55 LTE-A PRO device has been selected for field installation and as a component of the laboratory test bench setup for preliminary network testing. These tests were intended to evaluate the potential impact of radio coverage levels on the continuity of the cyclic communication between the central and peripheral stations using the FSFB/2 and PVS protocols. The RV55 LTE-A PRO is categorized as 12 and is compatible with LTE network evolution, providing additional bandwidth and availability, resulting in better

coverage, speeds, and reliability. It achieves this by using additional LTE bands that can be bundled together to increase capacity and improve the user experience.

	North America	EMEA	North America	Global	Global
	LTE		LTE-A Pro		LTE-M/ NB-IoT
LTE CATEGORY	Cat 4 (WP7610/WP7607)		Cat 12 (EM7511/EM7565)		Cat M1/NB1 (WP7702)
Peak D/L	Up to 150 Mbps		Up to 600 Mbps		Cat-M1: 300kbps Cat-NB1: 27kbps
Peak U/L	Up to 50 Mbps		Up to 150 Mbps		Cat-M1: 375kbps Cat-NB1: 65kbps
4G LTE Frequency Bands	1900(B2), AWS(B4), 850(B5), 700(B12), 700(B13), 700(B17), 1700(B66)	2100(B1), 1800(B3), 2600(B7), 900(B8), 800(B20), 700(B28)	2100(B1), 1900(B2),1800(B3), AWS(B4), 850(B5), 2600(B7), 900(B8), 1800(B9), 700(B12), 700(B13), 700(B14), 850(B18), 850(B19), 800(B20), 850(B26), 700(B29), 2300(B30), 1500(B32), TDD B41, TDD B42, TDD B43, TDD B46, CBRS B48,1700(B66)	2100(B1), 1900(B2), 1800(B3), AWS(B4), 850(B5), 2600(B7), 900(B8), 1800(B9), 700(B12), 700(B13), 850(B18), 850(B19), 800(B20), 850(B26), 700(B28), 700(B29), 2300(B30), 1500(B32), TDD B41, TDD B42, TDD B43, TDD B46, CBRS B48, 1700(B66)	2100(B1), 1900(B2), 1800(B3), AWS(B4), 850(B5), 900(B8), 700(B12), 700(B13), 700(B17), 850 (B18), 850(B19), 800(B20), 850(B26), 700(B28)
3G HSPA/HSPA+ Frequency Bands*	1900(B2), AWS(B4), 850(B5)	2100(B1), 900(B8)	2100(B1), 1900(B2), AWS(B4), 850(B5), 800(B6), 900(B8), 1700(B9), 850(B19)	2100(B1), 1900(B2), AWS(B4), 850(B5), 800(B6), 900(B8), 1700(B9), 850(B19)	
2G EDGE/GSM/GPRS Frequency Bands		900, 1800			850, 900, 1800, 1900
APPROVALS Regulatory	FCC, IC, PTCRB	GCF, CE	FCC, IC, PTCRB	FCC, IC, PTCRB, GCF, CE, RCM, IFT, Anatel	FCC, IC, PTCRB, GCF, CE, RCM
Carrier	Verizon, AT&T		Verizon, AT&T/FirstNet, US Cellular, Sprint, Tellus	Verizon, AT&T, Telstra(Planned)	Verizon(Cat-M), AT&T(Cat-M)
PART NUMBER Regulatory	1104335	1104337	1104303, 1104302 (Wi-Fi) 1104302 (Wi-Fi)	1104332, 1104331 (Wi-Fi) 1104331 (Wi-Fi)	1104333

Figure 26 – RV55 frequency characterization

The choice of the following LTE device was primarily based on its frequency band allowances, backward compatibility, and fallback with previous radio frequency technologies such as 3G, HSPA, and HSPA+ (including Third Generation and beyond with High-Speed Packet Access radio frequency technology). Additionally, the device's specifications align with the requirements for outdoor installation, ensuring operation within a temperature range of -40 to 70 degrees Celsius. It has been tested for military compliance specifications, including shock, vibration, thermal shock, and humidity, and is classified as IP68 for ingress protection. Another factor that supported the selection of this device was its compatibility with security specifications that align with the management of tunnelling and Radius. This compatibility is particularly desirable as it facilitates the integration of the CSP platform, which provides cybersecurity countermeasures, as highlighted in the preceding paragraph.

The selected antenna for connection to the router is the Airlink Antenna with high gain directional capabilities, which has undergone testing and certification to operate with AirLink routers and gateways. The High Gain Directional antenna provides 2x2 MiMo signal boosting for 2G, 3G, and 4G LTE networks. It contains two separately fed wideband elements in a single housing, enabling the High Gain Directional

antenna (with 6dBi of peak gain at 698-960MHz and 9dBi peak gain at 1710-2700MHz) to support client side MiMo and diversity. The decision to use MiMo and omnidirectional antennas, rather than directional ones, was made to capture a higher average signal strength level at the receiver side, providing a general-purpose solution and taking advantage of the benefits offered by Multiple-In, Multiple-Out technology.

	Specification
INPUT/OUTPUT	Configurable I/O pin on power connector • Digital Input ON Voltage: 2.7 to 36 VDC • Configurable Pull-up for dry contact input • Digital Open Collector Output > sinking 500 mA • Analog Input: 0.5-36 VDC
LAN (ETHERNET/USB)	DHCP Server — Host Interface Watchdog IP Passthrough — PPPoE VLAN
SERIAL	TCP/UDP PAD Mode Modbus (ASCII, RTU, Variable) PPP DNP3 Interoperability Dual Serial option (with an accessory)
NETWORK AND ROUTING	Network Address Translation (NAT) — Reliable Static Route Port Forwarding — Dynamic DNS Policy Routing — Verizon PNTM NEMO/DMNR — IPV6 Gateway VRRP
VPN	IPsec, GRE, and OpenVPN Client Up to 5 concurrent tunnels Split Tunnel Dead Peer Detection (DPD) FIPS 140-2 compatible
APPLICATION FRAMEWORK	ALEOS Application Framework (AAF) LUA Scripting Language
POWER	Input Voltage: 7 to 36 VDC LTE Idle Power: 900mW (75 mA @ 12VDC) Standby Mode Power: 53 mW (4.4 mA @ 12 VDC) triggered on low voltage, I/O or periodic timer Low voltage disconnect to prevent battery drain Built-in protection against voltage transients including 5 VDC engine cranking and +200 VDC load dump Ignition Sense with time delay shutdown Configurable features and ports to optimize power consumption
SECURITY	Remote Authentication (LDAP, RADIUS, TACACS+, DMZ) Inbound and Outbound Port filtering Inbound and Outbound Trusted IP MAC Address Filtering PCI compatible Secure Firmware Update

	Specification
NETWORK MANAGEMENT	Secure mobile network & asset management application available in the cloud or licensed platform in the enterprise data center Fleet wide firmware upgrade delivery Router configuration and template management Router staging over the air and local Ethernet connection Over-the-air software and radio module firmware updates Device Configuration Templates Configurable monitoring and alerting Remote provisioning and airtime activation (where applicable)
ROUTER MANAGEMENT	ALMS Local web user interface AT Command Line Interface (Telnet/SSH/Serial) SMS Commands SNMP
EVENTS ENGINE	Custom event triggers and reports Configurable interface, no programming Event Types: Digital Input, Network Parameters, Data Usage, Timer, Power, Device Temperature and Voltage Report Types: RAP, SMS, Email, SNMP Trap, TCP (Binary, XML, CSV) Event Actions: Drive Relay Output
ENVIRONMENTAL	Operating Temperature: -40°C to +70°C / -40°F to +158°F Operating Temperature (Wi-Fi variant): -30°C to +70°C / -22°F to +158°F Storage Temperature: -40°C to +85°C / -40°F to +185°F Humidity: 95% RH @ 60°C Military Spec MIL-STD-810G conformance to shock, vibration, thermal shock, and humidity IP64 rated ingress protection
INDUSTRY CERTIFICATIONS	Safety: IECEE Certification Bodies Scheme (CB Scheme), UL 60950** Vehicle Usage: E-Mark (UN ECE Regulation 10.04), Rail Usage: EN50155 ISO7637-2, SAE J1455 (Shock & Vibration) Hazardous Environments: Class 1 Div 2– Ambient temperatures of -30°C to +60°C Environmental: RoHS, REACH, WEEE

Figure 27 – RV55 additional specifications

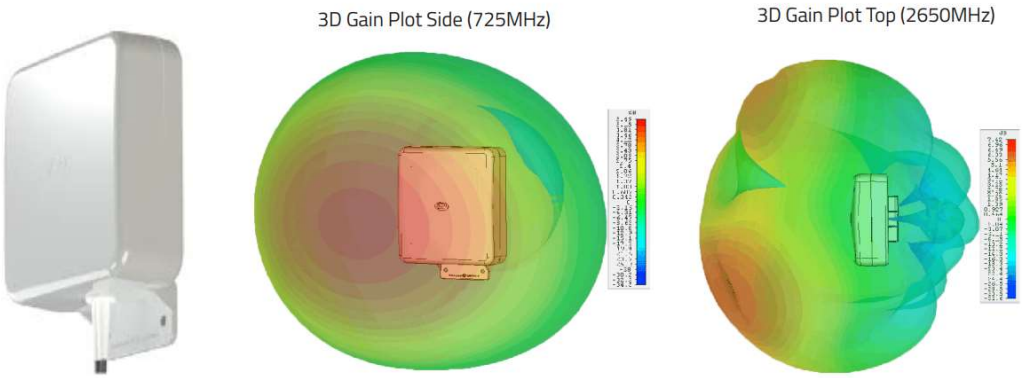


Figure 28 – Antenna and gain plot

MiMo technology stands for Multiple-In Multiple-Out communication, which sends the same data as several signals simultaneously through multiple antennas, while still utilizing a single radio channel. This is a form of antenna diversity, which uses multiple antennas to improve signal quality and strength of a radio frequency link. The data is split into multiple data streams at the transmission point and recombined on the receive side by another MiMo radio configured with the same number of antennas. The receiver is designed to consider the slight time difference between receptions of each signal, any additional noise or interference, and even lost signals. By transmitting the same data on multiple streams, the MiMo technology introduces redundancy into data transmission that classic single antenna setups (SiSo: Single In, Single Out) can't provide and therefore it can leverage on several advantages such as to receive high signal level owing to multipath phenomena and reduce the impairments of fading.

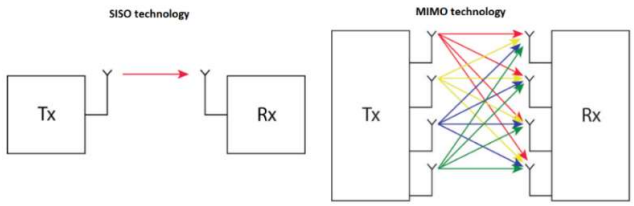


Figure 29 – SiSo Vs MiMo technology benefit

MiMo technology can take advantage of multipath propagation, which involves using bounced and reflected radio frequency transmissions to improve signal strength even without a clear line of sight. MiMo receives and combines multiple streams of the same data that are received at slightly different time intervals, enhancing the overall throughput, and allowing for improved quality and quantity of data to be transmitted over the network. This technology is particularly useful in urban environments, where signal degradation between single antennas without a clear line of sight is a significant issue. By utilizing multiple data streams, MiMo can reduce issues such as fading that cause lost or dropped data packets, resulting in a better-quality transmission.

3.2.6 Telecom Service Level Agreement

The activities outlined thus far, including the selection of a radio frequency technology architecture that meets cybersecurity levels through the implementation of tunnelling and MAC authentication protocols, the definition of a private APN that only allows predefined SIMs to register to the network, the need for a high

quality signal at the receiver side in peripherical zones of interest, and the requirement for continuous and cyclic communication to prevent any interruptions or disruptions to the railway system, have established the basis for determining a Service Level Agreement (SLA) for the feasibility of the industrial research being conducted. A Service Level Agreement (SLA) is a contract between a service provider, in this specific case, a telecom provider and a customer, which outlines the types and standards of service to be provided, as well as the architecture required to achieve the objectives. One of the main considerations when leveraging the public network and utilizing existing radio infrastructure managed by a public telecom provider is the avoidance to install additional and dedicated eNodeB for signal coverage evaluation. Additionally, due to the nature of radio frequency signals, which can be influenced by environmental and atmospheric conditions, and the desire to implement a redundant network architecture to minimize service interruptions, the SLA has been defined to request a quality of service as the outcome of input requirements directed to two different telecom providers, TIM, and Vodafone.

The results of track surveys, including radio signal measurements on the experimental track Padiglione Nettuno and the outcome obtained from test conducted into the laboratory test bench installed in Bologna, have indicated a sufficient coverage level to ensure data communication for both chosen telecom providers. As the track Padiglione Nettuno is a Ministero dello Sviluppo Economico (MiSE) experimentation, and the laboratory activity is part of the European railway industrial research for the upcoming commercial operation of ERTMS standard for low density lines over the next two years, a cost saving strategy has been adopted. Since the two projects share the same objective and input requirements, a service level agreement was requested from TIM for the experimentation, while a second agreement was requested from Vodafone for laboratory and further evaluations. The input requirements for achieving a railway service based on radio frequency architecture were examined, considering aspects such as signal coverage, service continuity, backward compatibility with 3G, throughput, latency, security in line with cybersecurity, the ability to customize service quality, and obtaining priority policies or radio resources dedicated solely to railway purposes, both for trackside-to-trackside communications and future trackside to onboard communications. A first bunch of input requirements have been therefore addressed to both telecom providers for the service level agreement definition:

- ✓ Minimum RSRP/RSRQ/SINR in the peripherical nodes and along the railway line:
  -95 dBm/-15 dB/8 dB.
- ✓ Average RSRP/RSRQ/SINR in the peripherical nodes and along the railway line:
  [-85; -95] dBm/ [-10; -15] dB/ [12; 15] dB.

- ✓ Service availability & continuity > 99.9% per year.
- ✓ Fixed trackside connectivity by 4G backhaul.
- ✓ LTE HOSR (Handover Success Rate) > 99%.
- ✓ LTE PS CDR (Call Drop Rate) < 1%.
- ✓ LTE PS Service setup time < 1 second.

The list above regarding the coverage level, the continuity in the service operation and further features for the setup of a communication has been allowed by the telecom providers always under the constraint of a good signal coverage around the installation.

A second group of requirements is represented by the type of communications and quantity of data to be transmitted with the definition of a maximum latency to be respected and tolerable, communication gaps that do not impact on the normalization of railway protocols causing disruptions as listed hereby:

- ✓ Latency < [100 - 150] milliseconds.
- ✓ Upper-bound in communication gap < 500 milliseconds.
- ✓ Priority in managing vital (signalling data) and not vital data (maintenance data) with different QCI (Quality of Service Class Identifier) as specified by the 3GPP standardisation.

The specified packet delay budget is intended to maintain the performance, response times, and reaction times of the signalling system, comparable to those achieved with a dedicated private fibre network. However, if there is a network communication loss exceeding 500 milliseconds, the signalling equipment will enter a safety restrictive state, resulting in train movement being stopped.

In contrast to the first set of requirements, the fulfilment of this second set cannot be guaranteed by the telecom operators, as it depends on environmental and traffic factors within the cell served by the radio base station. Consequently, the telecom provider can only offer the best bit rate available according to the surrounding conditions. To address this uncertainty in radiofrequency-based communication, the protocols were configured to be robust to communication gaps of up to one second without affecting the safety response time. Finally, the following set of requirements aims to obtain dedicated resources from the telecom providers to create a more reliable and secure network that can be compared to a private network, despite its public network nature:

- ✓ SIM ARP profile = 1 (pre-emption capability, no vulnerability).
- ✓ RAN Sharing configuration (common carrier – easily, or dedicated carrier) on LTE for dedicated PLMN. Availability of dedicated RB resources (8<RB<10).

- ✓ Dedicated APN infrastructure.

Nowadays, the 4G standard does not allow for the assignment of dedicated radio resources to specific users, except in the military field, and it does not provide priority to assigned users, as it is a public network where resources are shared. Consequently, these requirements were not addressed by the telecom operators, also due to legislative constraints. These features can, however, be accommodated with 5G radio frequency, which is likely to be the basis of the Future Radio Mobile Communication System (FRMCS) under specification. 5G is characterized by its ability to manage the concept of priority and provide diversity in the use of requested services through the slicing concept. Only the final requirement listed above will be fulfilled by both telecom providers, and a similar solution, based on MPLS-CDA technology, as shown in the figure below, was proposed to create a private APN to which only certain SIMs can register, enabling more secure communication.

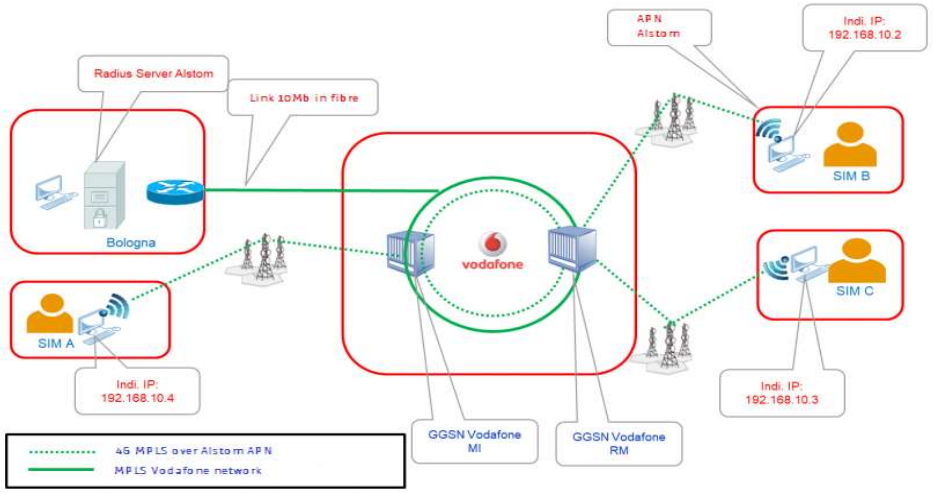


Figure 30 – MPLS infrastructure and dedicated APN

3.2.7 Trackside equipment handled through LTE existing infrastructure

The purpose of this paragraph is to emphasize a practical implementation of data communication between two nodes situated at a considerable distance, such as in two different regions (e.g., Emilia Romagna and Tuscany) on the railway system. This implementation was demonstrated to the Ministero dello Sviluppo Economico and utilized existing radio frequency technology, such as LTE.

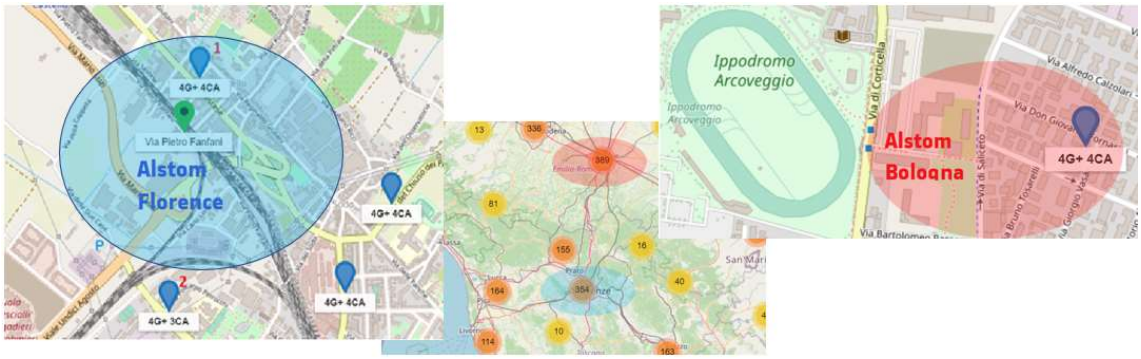


Figure 31 – Real railway system communication over LTE

The test activity carried out in a real environment was intended to evaluate the feasibility of the proposed solution and provide results for the continuously evolving industrial research, as per the interest of MISE. Due to bureaucratic and logistic constraints with RFI (Rete Ferroviaria Italiana), it was not possible to use the installations on the experimental track Padiglione Nettuno.

As previously mentioned, the research program is an essential strategic point for Alstom, and despite continuous communication with telecom providers, the implementation solution of MPLS technology associated with the private APN is still being finalized. Therefore, a communication based on a distributed architecture was established instead of setting up a flow between the control room served by a wired network and a generic peripheral node covered by an existing eNodeB.

As highlighted in previous paragraphs, the research aims to create communication between the control room and peripheral nodes and among peripheral nodes, implementing a distributed communication. Part of the logic must be decentralized to limit radio access in the flow data exchange, ensuring safety response time constraints are met.

To verify the movement of a turnout located in Florence and the reception of permissive signals from the remote field upon route setting, an environment was established using real components for LTE vs. LTE communication without redundant network configuration. The overall architecture of this setup, based on the aforementioned components, is illustrated in the accompanying picture.

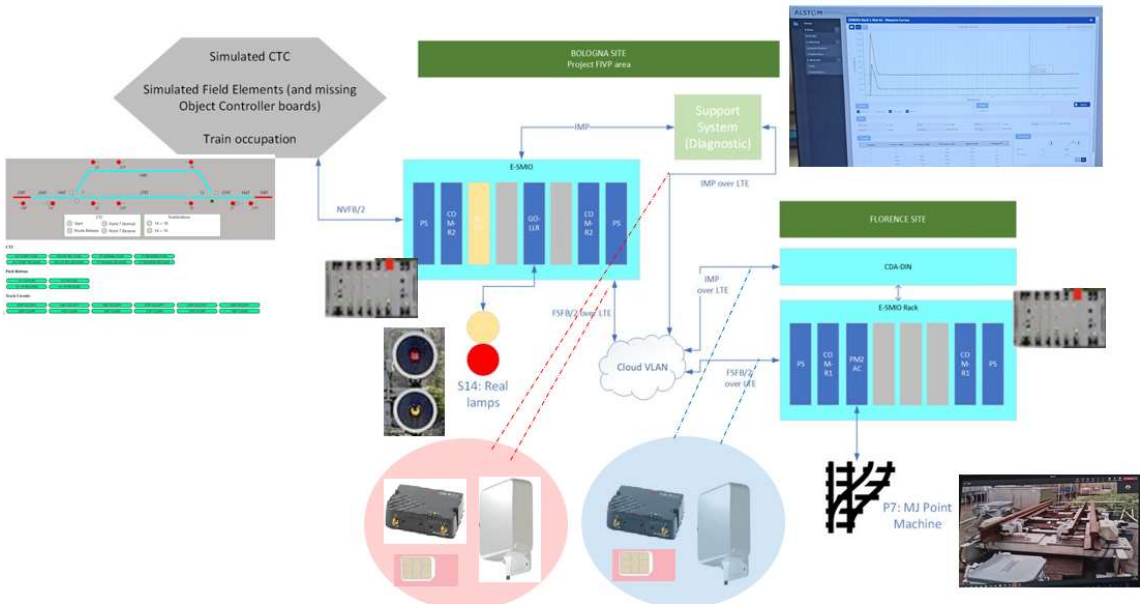


Figure 32 – Real railway system communication over LTE setup

The test has been conducted with the following hardware, software, and cloud configuration, once a sufficient coverage level had been evaluated in both peripheral nodes:

- ✓ 2 routers LTE (Sierra Wireless RV55 LTE-A-PRO) and associated MiMo antennas (AirLink High Gain Directional) respectively fitted in Bologna and Florence equipped with Vodafone SIM M2M registered to the public Vodafone APN (m2mbis.vodafone.it) since the private Alstom one is under finalizing phase.
- ✓ 1 real e-SMIO located in Bologna configured with part of the logic to control a turnout and simulated field equipment such track circuits and signals for the route setting.
- ✓ 2 real signal lamps connected to the real e-SMIO located in Bologna.
- ✓ 1 real e-SMIO located in Florence configured to communicate with the other e-SMIO located in Bologna.
- ✓ 1 real turnout located in Florence.
- ✓ 1 Support System based on web browser application to manage maintenance aspect and field measurements.
- ✓ 1 HMI (Human Machine Interface) simulated based on web browsing application to send commands.

- ✓ 1 VPN instantiated between the two nodes based on Google Cloud without further cybersecurity implementation since MPLS architecture and Radius still under realization.

The test was carried out using the system architecture illustrated above, after verifying a sufficient level of received signal, which fell within the range of [-80; -90] dBm in the two identified areas of interest. These areas were in Bologna and Florence and were served by different eNodeBs.

To achieve a distributed architecture that reduces communication interchanges and system response time over a radio frequency media, part of the logic for controlling a turnout and simulated field equipment such as track circuits and signals for route setting was implemented on real e-SMIO hardware devices. This approach eliminates the need for a centralized controller.

A web-based HMI simulation was interfaced to operate with the logic configured in the local and peripheral devices, as well as the real equipment managed by the two e-SMIO units, such as turnouts and lamps. This allowed for the sending and receiving of commands between the communicating environments. The successful actuation of the commands, using radio frequency technology, was verified bidirectionally by moving a peripheral device managed turnout and receiving the status of the lamps according to a given itinerary command.

In the first test, the command sent using the HMI successfully changed the status of the turnout managed by the e-SMIO located in Florence. A camera recording the real railway equipment confirmed the actuation and movement of the peripheral device. In the second test, a simulated route was set on the peripheral zone using the previously configured logic. This allowed for the status of the lamps fitted in Bologna to be received, and they were lit according to the status of the set route.

3.3 Research and Design activity results

The tests, witnessed by the MISE institute, have yielded positive preliminary results for using a public telecommunications network based on current radio frequency architectures. This confirms the feasibility of utilizing the existing LTE infrastructure for ERTMS railway communication, which involves low traffic.

The main implementation aspects and constraints involve verifying the signal coverage, which can affect communication latency, and installing a dedicated antenna type to capture the most powerful and compliant signal with railway certifications. Additionally, the low traffic of the network managed by the radio

base stations serving the geographical areas in which the peripheral zone resides must be considered. These factors influenced the decision to adopt the LTE solution for low density railway lines.

To improve signal strength, the MiMo antenna was chosen to leverage antenna diversity technology. However, this solution will only be adopted if the received power is lower than -100 dBm and will not be installed in cell edge topology, as indicated in the service level agreement requested from the telecom provider. Based on these constraints, the two peripheral nodes located in Bologna and Florence had signal power within the range of [-80, -90] dBm, enabling bidirectional communication with latency values lower than the communication cycle of the railway equipment. This ensured that the use of the public network did not cause additional delays in the final response of the system.

Due to legislative constraints expressed by telecommunications operators, ad-hoc radio resources such as the assignment of a dedicated frequency of use, pre-emption policies in the network registration phase, and dedicated quality of service cannot be utilized. Consequently, the solution is best implemented in areas characterized by a good level of signal coverage and low traffic density. This approach reduces safety disservices caused by the traffic load managed by the public network during continuous and cyclical communication among railway equipment. It also minimizes interferences in the use of the same media caused by other users served by the same radio base station.

When operating on an open and public network, guaranteeing safety in system response times through good received signal levels may require the optimization of directional antennas. However, leveraging on a public network necessitates changes in protocols to introduce encryption techniques and certificates to ensure the authenticity of data exchanged between the railway subsystems. This is done to reduce access to the network and data content. To address these concerns, the deployment of DTLS protocol and encryption techniques is necessary, enabling the use of proprietary protocols FSFB/2 and PVS in a public and open network at level 3. Since these protocols are designed to operate on closed networks, an update of the kernel on the object controller devices is needed to integrate this feature.

Kernel modifications are currently in progress, and tests have been conducted to launch the protocols without adding encryption techniques, as they are not yet supported by current railway hardware devices. This was done solely for the purpose of assessing the feasibility of instantiating communication using LTE architecture. To achieve this, a Virtual Private Network (VPN) was deployed for two reasons: to create a point-to-point tunnel between the communication of the two peripheral nodes, establishing a sort of dedicated channel where access is granted based on knowledge of the cryptographic key, and to translate

the dynamic IP provided by the telecom operator during network registration to a static one. However, the introduction of tunnelling resulted in an approximately 10% increase in communication latency, without significantly affecting the overall safety response time of the system in command actuation.

Further developments are being considered to enhance the overall system's protection, which is still in the finalization phase. These include requesting a private APN to which only certain SIMs can register on the network and access for communication, as well as introducing a cybersecurity subsystem. This means that every time a peripheral site needs to access the network, it must overcome two protection barriers. The first is defined by the private APN, which verifies whether the generic SIM belongs to its pool. The second requires access to be granted only if the MAC address is registered in the Radius through the MAC authentication protocol.

To conclude, the results obtained are primarily dependent on the levels of coverage and traffic intensity of the telecommunications provider. Therefore, these results only serve as a starting point for the research and are not yet ready for commercial implementation. As per the agreements with the telecommunications providers, the system must be improved by customizing specific quality of service indicators, including cyber policies and accessibility to the system managed by Radius through the MAC authentication protocol, to ensure maximum reliability.

4 Chapter four: Machine Learning world

4.1 Machine Learning introduction

The ongoing advancement of technology, which is leading to an increasing amount of communication through both wired and Radio Frequency channels, has brought about a heightened interest in the field of Machine Learning. This is due to the growing importance of exchanging data between diverse entities. Machine Learning is part of Computer Science and seeks to optimize a system's efficiency by allowing it to improve its knowledge or perform specific tasks for which it was designed. This is achieved through learning from the environment in which it operates.

Machine Learning plays a crucial role in the realm of artificial intelligence and big data analysis. Although Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on creating systems that improve performance or learn from data, AI is a more general term that describes systems or machines that replicate human intelligence. While the terms Machine Learning and Artificial Intelligence are often used interchangeably, they do not have the same meaning. It should be noted that everything related to Machine Learning falls under Artificial Intelligence, but the latter is not limited to only Machine Learning.

The field of Machine Learning is closely associated with pattern recognition and statistical inference. It involves working with data and processing it using specific algorithms to uncover underlying patterns in the selected features. These patterns can then be used to analyse new data and achieve the primary objective of learning from the available input data. Although it may appear to be a ground-breaking new technology, Machine Learning is not new. In fact, the first steps in the development and understanding of the Machine Learning field were taken as early as 1950.

The 31 October 1950, Alan Turing, a researcher at the University of Manchester, published, an important scientific journal in the psychological and philosophical field, an article entitled "Computing Machinery and Intelligence". The article opens with a very strong provocation: "Can machines think?".

Back in 1950, while studying games like checkers and chess, Alan Turing envisioned that machines could learn, memorize, and execute strategies that could be codified into rules. Although the computers of the time were not yet capable of implementing this idea, Turing posed a challenge to philosophers and psychologists

by defining rules for determining when a machine is simulating human behaviour (the famous Turing Test). This sparked one of the most intriguing debates in history.

Since that time, numerous studies have been conducted, leading to the development of Machine Learning into the 21st century. Today, it is a tool used in a wide range of daily human activities, including interacting with banks, receiving suggestions for online purchases and movies based on previously watched genres on streaming platforms.

As previously mentioned, Machine Learning is not a new concept and has been steadily advancing over the years. However, its foundations have remained unchanged and consist of:

- ✓ Statistics: The objective of Machine Learning is to derive a model from observed data, and to achieve this, it must employ statistical measures to evaluate the model's performance, estimate it, and filter out noise present in the data.
- ✓ Computer science algorithm design: The purpose of this is to describe methodologies for optimizing parameters and executions.

Bearing this in mind, Machine Learning is well suited to address various types of problems based on the specific task for which the system has been designed, including:

- ✓ Classification: This is a type of supervised learning that involves providing input data with corresponding labels or targets. This process involves predicting the class of a given data point by approximating a mapping function f from input variables x to discrete output variables y. An example of a real-world application of classification is email spam detection.
- ✓ Prediction: The term "prediction" refers to the output of an algorithm that has been trained on a historical dataset and applied to new data to forecast the likelihood of a particular outcome. If the events to be predicted are like those previously classified, it is possible to determine which class the new event belongs to. One real world application of prediction is market trend forecasting.
- ✓ Pattern recognition: This is a machine learning technique that uses data analysis to identify recurring patterns and regularities in data samples. This involves identifying common characteristics or repetitive structures between the data samples to establish patterns. The technique is capable of rapidly and accurately recognizing partially hidden patterns even in unfamiliar objects. One example of this application is facial recognition.
- ✓ Clustering: This is an unsupervised procedure that involves identifying groups within data. Objects with similar characteristics are grouped together under the same cluster, which is formed by

minimizing the distance between each point in the cluster and its respective centroid. One practical application of clustering is grouping a company's customers into several clusters based on their similarities.

- ✓ Regression: This is a technique used in supervised learning applications to analyse the relationship between independent variables or features and a dependent variable or outcome. This approach is commonly employed for predictive modelling in machine learning, where an algorithm is used to predict continuous outcomes, as opposed to classification techniques where values are discrete. One real world application of regression is predicting housing prices using large databases of real market data.
- ✓ Transfer Learning: This is a machine learning technique that involves improving the learning process for a new task by transferring knowledge from a related task that has already been learned. This technique involves reusing a pretrained model as the starting point for a new task, thereby optimizing the model's progression to solve the new task and achieving significantly higher performance than training with only a small amount of data. One real world application of transfer learning is a vision system that can learn to detect objects invariant to changes in lighting, rotation, and other factors by learning useful features after observing somewhat unrelated/different objects from different points of view and in different lighting conditions.
- ✓ One-shot Learning: This is a technique commonly used in computer vision to learn information about object categories from one or a few training images, in contrast to other methods. This can be accomplished using architectures with augmented memory capacities, such as Neural Turing Machines. These architectures allow for the quick encoding and retrieval of new information, thereby avoiding the need to inefficiently relearn their parameters as gradient based networks compute.

As Machine Learning is widely adopted in different application fields, tasks can be classified into several categories, including Unsupervised Learning, Supervised Learning, and Reinforcement Learning.

In few words, in Supervised Learning, the objective is to generate a formula based on input and output values. Unsupervised Learning involves finding associations between input values to create groups. In Reinforcement Learning, an agent can learn through a trial-and-error method by interacting with the environment and receiving delayed feedback upon the actions executed.

The following table provides a general overview of the main properties that characterize the three different families of Machine Learning.

Criteria	Supervised ML	Unsupervised ML	Reinforcement ML
Definition	*Agent learns using labelled data*	*Agent trained using not labelled data*	*Agent learns interacting with the environment with no predefined data*
Type of problems	*Regression and Classification*	*Association and Clustering*	*Exploitation or Exploration*
Supervisions	*Yes*	*No*	*No*
Objective	*Output calculation, minimizing the risk function f (x, ŷ)*	*Underlying patterns discovering*	*Learning from a series of actions*
Application field	*Risk evaluation, Image recognition or video stream classification, Network traffic prediction*	*Anomaly detection, Marketing, Network traffic pattern recognition*	*Self-Driving Cars, Gaming, Healthcare, Recommendation system, network resource allocation*
Algorithms	*Linear Regression, Lasso Regression, SVM, DT, DF, RF.*	*K – Means, C – Means, A priori*	*Value-Based methods: Q – Learning, SARSA*

Table 4- Machine Learning Properties

4.1.1 Unsupervised Learning

While unsupervised learning has not yet been widely implemented, this methodology represents the future of Machine Learning and its potential. Unsupervised machine learning involves training models on raw and unlabelled training data. This approach is often used to identify underlying patterns and trends in raw datasets or to cluster similar data into a specific number of groups based on their similarity. It is also frequently used as an approach in the early exploratory phase to gain a better understanding of the datasets.

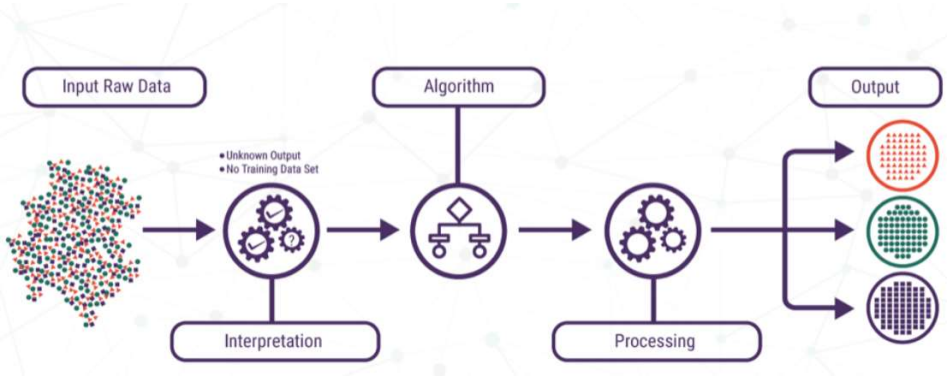


Figure 33 – Unsupervised Learning process overview

As previously mentioned, during the process of unsupervised learning, the system lacks concrete datasets, and the outcomes to most problems are largely unknown. In simpler terms, the AI system and the ML objective are blinded when they enter the operation. The system has faultless and immense logical operations to guide it along the way, but the absence of proper input and output algorithms makes the process even more challenging.

Once it has the input data, the unsupervised learning system learns as much as possible from the available information. The system works independently to recognize the problem of classification, as well as differences in shapes and colours. Using the information related to the problem at hand, the unsupervised learning system can recognize all similar objects and group them together. The labels assigned to these objects are determined by the machine itself. Technically, there may be incorrect answers due to a certain degree of probability. However, like how humans operate, the strength of machine learning lies in its ability to recognize errors, learn from them, and improve its estimations in the future.

4.1.2 Supervised Learning

Supervised learning aims to establish an input output relationship using a dataset of pre labelled samples. This involves learning a parametrized risk function that needs to be minimized to solve the optimization problem at hand. The risk function is a combination of a loss function, which can be selected, based on how training data are managed, such as Absolute Error, Squared Error, or Mean Squared Error, and a joint distribution that describes the relationship between the input and output.

In supervised learning, the output of the algorithm is fed into the system. This means that the machine already knows the output of the algorithm before it begins working or learning from it. An example of this concept would be a student learning from an instructor in a course, where the student knows what they are learning from the course beforehand.

In supervised learning, since the output of the algorithm is already known, the system only needs to determine the steps or process required to reach the output from the input. The machine is taught through a training dataset that guides it. If the process deviates and the algorithm produces results that are significantly different from the expected output, the training data is used to guide the algorithm back on the right path.

Supervised Machine Learning is currently the most used type of machine learning for solving classification and regression problems. The input variable (x) is connected to the output variable (y) using an algorithm. All the input, output, algorithm, and scenario are provided by humans.

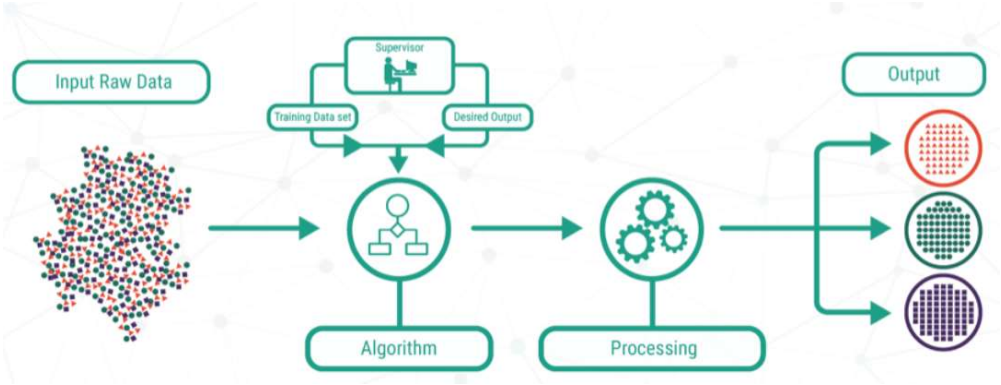


Figure 34– Supervised Learning process overview

Supervised Machine Learning techniques offer advantages in terms of their ability to define specific labels, which allows for training the algorithm to distinguish between different classifiers using decisional bounds. This specificity in label definition also contributes to the supervised algorithm's higher accuracy and reliability compared to other machine learning techniques. However, these advantages in accuracy and reliability can make it challenging to adopt supervised techniques in real time environments due to the large amount of data and computational complexity involved.

4.1.3 Reinforcement Learning

Reinforcement Learning aims to optimize the behaviour of an agent in a non-stationary environment and is distinct from both unsupervised and supervised learning. In this approach, the environment provides rewards and a new state based on the actions of the agent. Therefore, in reinforcement learning, we do not teach an agent how to perform a task, but rather provide it with rewards, whether positive or negative, based on its actions. To leverage the Reinforcement Learning technique, it is necessary to provide a formal description of the environment in which the agent will operate. The objective of this description is not to precisely define the environment, but rather to make general assumptions about the properties that the specific environment possesses. Reinforcement Learning algorithms typically assume that the environment can be designed as a Markovian Decision Process, where the future state depends only on the previous state,

without considering the entire previous history. When a problem satisfies the Markovian hypothesis, it can be treated as a dynamic process with one step back in memory. This means that the problem can be treated as episodic, where episodes represent repeated attempts to achieve a certain objective set by the problem. All the algorithms related to this approach of learning base their choices on the assumption that the values provided by the environment are only a function of the current state and the action taken by the agent in the previous instant in episodic environments, where a termination state is always defined, or in non-episodic environments where there is no termination state. Therefore, in addition to agents living in an environment assumed to be a Markovian Decision Process, the Reinforcement Learning paradigm is identified by four sub-elements: policy, reward signal, value function, and optionally, a model of the environment. The policy element defines how the agent learns from the environment and behaves in each moment. It is the core component of a Reinforcement Learning agent because it solely determines the agent's behaviour.

A policy can be simplified and summarized as a mapping between the actions taken by the agent at each moment and the states perceived by the environment.

The second important element is the reward signal, which is sent by the environment and received by the agent. The goal of Reinforcement Learning is to maximize the total rewards received during an episode, and the reward signal represents this objective.

Bellman's equations express the problem of maximizing the expected sum of rewards in terms of a recursive relationship with the value function. The third element is the value function of the states, which is also known as the Q-function. It defines the total amount of rewards that an agent, living and interacting with an environment, is expected to accumulate in the future.

When comparing the reward and value functions, the former indicates what is optimal in the short-term by computing one cycle of the diagram below, while the latter determines the long-term desirability of the states by continuously computing the cycle after taking into consideration the best states to follow and the rewards available in those states. Rewards are directly given by the environment, while the values must be estimated and re-estimated from the sequences of observations made by the agent throughout its entire existence.

The fourth and final element is the model of the environment. It can be viewed as an entity capable of simulating the behaviour of the environment. Given a state-action pair, the model can predict the outcome of the next state-action pair.

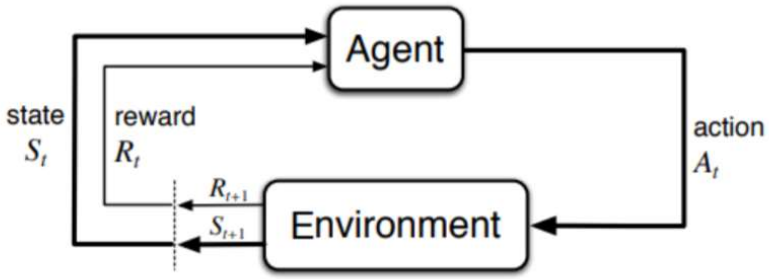


Figure 35 – Reinforcement Learning cycle

The agent receives the current state S_t from the environment and reacts with action A_t. This changes the environment to state S_{t+1} and causes a reward of R_{t+1}.

Reinforcement learning can solve very complex problems that cannot be addressed by conventional techniques. Its objective is to achieve long-term results and maximize performance, rather than reaching an ideal behaviour in the interaction with the environment. The algorithms used in reinforcement learning are characterized by the ability to maintain a balance between exploration and exploitation. Exploration involves trying different approaches to determine if they are better than what has been attempted before, while exploitation involves using the approaches that have worked best in the past. Other learning algorithms do not possess this balance between exploration and exploitation.

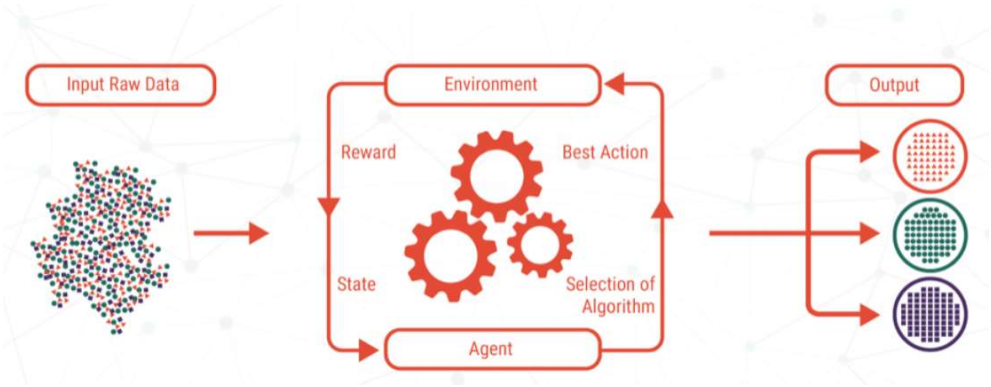


Figure 36– Reinforcement Learning process overview

Bearing on the considerations above, the following comparisons can be made among the three machine learning techniques:

- ✓ Supervised Vs Reinforcement Learning: In Supervised Learning, an external supervisor with sufficient knowledge of the environment is involved in the learning process to better understand and complete the task. However, in problems where the agent can perform various subtasks on its own to achieve the overall objective, the presence of a supervisor becomes impractical and unnecessary. Once the machine begins learning from its own experiences, it can utilize that knowledge to inform future actions. This is perhaps the most significant and crucial difference between reinforcement and supervised learning. Both types of learning involve a mapping between input and output. However, in Reinforcement Learning, there is an exemplary reward function that informs the system of its progress down the right path, unlike in Supervised Learning.
- ✓ Unsupervised Vs Reinforcement Learning: Reinforcement Learning operates on a mapping structure that guides the machine from input to output. In contrast, Unsupervised Learning does not have such a feature. In Unsupervised Learning, the machine's focus is on locating patterns rather than mapping for achieving the end goal. For instance, if the machine's task is to suggest relevant news updates to a user, a Reinforcement Learning algorithm would seek regular feedback from the user and use that feedback to build a reputable knowledge graph of all news related articles that the person may like. In contrast, an Unsupervised Learning algorithm would look at many other articles that the user has read, like the current one, and suggest something that matches the user's preferences.

4.2 Dimensionality reduction techniques in ML

Dimensionality reduction is a technique used to reduce the number of features in a dataset. Machine learning datasets typically contain hundreds of columns, or features, which can create a large sphere in a three-dimensional space. The goal of dimensionality reduction is to decrease the number of columns to a manageable count, such as transforming the three-dimensional sphere into a two-dimensional object.

As the number of features or factors in a dataset increases, it becomes more challenging to visualize and work with the training set. Additionally, features within a dataset are often correlated, leading to noise and redundancy that can affect the complexity of the model and increase the likelihood of overfitting. Training a machine learning model on a large dataset with many features can result in a model that is overfitted to the training data and performs poorly on real data. The purpose of dimensionality reduction is to reduce the number of features in the training data, leading to a simpler model and avoiding overfitting.

The main benefits of dimensionality reduction can be summarized as follows:

- ✓ noise and redundant features elimination.
- ✓ model's accuracy and performance improvement.
- ✓ usage of algorithms unfitted for more substantial dimensions.
- ✓ memory storage reduction.
- ✓ data compression reducing computation time and facilitating faster training of the data.

Dimensionality reduction techniques can be categorized as data transformations that can either be linear or non-linear. These techniques are typically organized into two methodologies: feature selection and feature extraction. The goal of feature selection is to identify a subset of input variables that are most relevant to the original dataset, while feature extraction (also known as feature projection) converts the data from a high-dimensional space to a lower-dimensional one.

The dimensionality reduction techniques used in this thesis for the purpose of reducing the dimensionality of input arrays for clustering representation include:

- ✓ PCA (Principal Component Analysis).
- ✓ T-SNE (t-distributed Stochastic Neighbour Embedding).

4.2.1 PCA: Dimensionality reduction in details

Principal Component Analysis is one of the leading linear techniques of dimensionality reduction. This method directly maps the data to a lower-dimensional space, maximizing the variance of the data in the low-dimensional representation.

Reducing the number of variables in a dataset inevitably involves sacrificing some degree of accuracy, but the key in dimensionality reduction is to trade a small amount of accuracy for simplicity. Smaller datasets are easier to explore and visualize, making data analysis faster and more straightforward for machine learning algorithms that don't need to process extraneous variables. The following technique can be considered an unsupervised learning problem in which the process of obtaining principal components from a raw dataset is performed, preferably after standardizing the input data. Standardization aims to normalize the range of the initial continuous variables so that each variable contributes equally to the analysis.

Performing standardization prior to PCA is crucial because PCA is highly sensitive to the variances of the initial variables. If there are significant differences between the ranges of the initial variables, the variables

with larger ranges will dominate over those with smaller ranges, resulting in biased results. Mathematically, the standardization can be implemented subtracting the mean of the given distribution (μ) and dividing by the standard deviation of the given distribution (σ) for each value of each variable.

$$z = \frac{x - \mu}{\sigma}$$

Ensuring that data is transformed to scales with the same order of magnitude is an important aspect to consider before proceeding with PCA. This process can be simplified and described in the following six main steps:

- ✓ Step 1: Take the whole dataset consisting of d+1 dimensions and ignore the labels such that the new dataset becomes d dimensional.
- ✓ Step 2: Compute the mean for every dimension of the whole dataset.
- ✓ Step 3: Compute the covariance matrix of the whole dataset.
- ✓ Step 4: Compute eigenvectors and the corresponding eigenvalues.
- ✓ Step 5: Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d × k dimensional matrix W.
- ✓ Step 6: Use this d × k eigenvector matrix to transform the samples onto the new subspace.

The steps mentioned above are performed to identify a low-dimensional set of axes that summarizes the available input data and create a new set of properties by combining the old ones. Mathematically, PCA performs a linear transformation that moves the original set of features to a new space composed of Principal Components. These new features do not possess any real world meaning, except for algebraic purposes. It is important to note that new features can be created by linearly combining existing features. Principal Component Analysis seeks to identify properties that exhibit the maximum possible amount of variation across classes, to create the Principal Component Space. The algorithm utilizes concepts such as Variance Matrix, Covariance Matrix, Eigenvector, and Eigenvalue pairs to perform PCA and provide a set of eigenvectors and their respective eigenvalues as output.

The Eigenvectors represent the new set of axes in the Principal Component Space, while the Eigenvalues contain information on the amount of variance that each Eigenvector possesses. To scale the dimensionality of the dataset, Eigenvectors with greater variances are selected, while those with less variance are discarded. For instance, when data is transferred from the original space to the Principal Component Space using PCA, the data points are projected in a manner that provides the most informative viewpoint of the object. This involves selecting the directions in which the data exhibits the greatest variance.

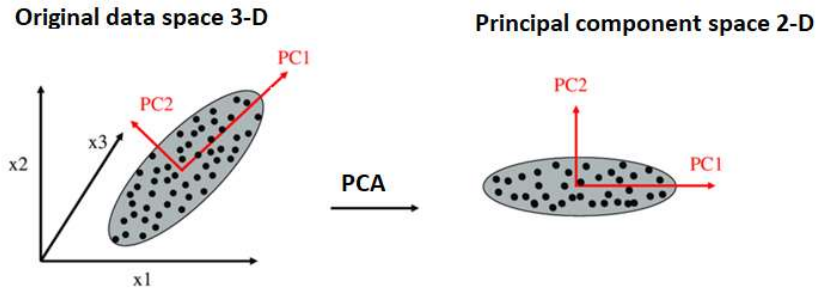


Figure 37 – PCA dimensionality reduction example

The following toy example, consisting of three labelled and standardized features, has been used to explore and illustrate the steps involved in the full PCA process. The objective of this process is to reduce dimensionality by moving from the original space of the input data to the Principal Component Space.

Feature 1	Feature 2	Feature 3	Label
90	90	90	1
30	60	60	2
60	90	60	3

Figure 38 – PCA: Toy example input dataset

After standardizing the input data and ignoring the labels, the dataset moves from d+1 dimensions to a new d-dimensional dataset. In modern machine learning paradigms, d can be thought of as the *X_train* data, while 1 can be thought of as the *y_train* labels. Together, *X_train* and *y_train* make up the complete training dataset. Therefore, after dropping the labels, the new d-dimensional dataset should be used to identify the principal components.

X_train *y_train*

Feature 1	Feature 2	Feature 3	Label
90	90	90	1
30	60	60	2
60	90	60	3

d + *1*

Dimension

X_train

Feature 1	Feature 2	Feature 3
90	90	90
30	60	60
60	90	60

d

Figure 39 – PCA: Toy example labels removed

The data presented in the table above can be represented in matrix A, where each column displays values for a feature and each row displays values for a class. For the given toy example, the resulting matrix is characterized by three rows and three columns.

$$A = \begin{vmatrix} 90 & 90 & 90 \\ 30 & 60 & 60 \\ 60 & 90 & 60 \end{vmatrix}$$

Next step regards the computation of the mean for every dimension of the whole dataset, obtaining a matrix $\bar{A}$.

$$E(A) = \bar{A} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

The matrix in outcome is therefore characterized by one row and number of columns based on the features of the entire dataset.

$$\bar{A} = [60 \quad 80 \quad 70]$$

The computation of the covariance matrix for the entire dataset, which is also known as the variance-covariance matrix, is the next step in the PCA process. The variance-covariance matrix is a square matrix where the diagonal elements represent the variance, and the off-diagonal elements represent the covariance.

$$Cov(A) = \begin{bmatrix} Var\,(x_1) & \cdots & Cov\,(x_n, x_1) \\ \vdots & \ddots & \vdots \\ Cov\,(x_n, x_1) & \cdots & Var\,(x_n) \end{bmatrix}$$

To determine the covariance matrix, it is necessary to use the formulas for variance and covariance. Variance is a measure of the variation of a single random variable and can be expressed as follows:

$$Var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Covariance is a measure of the extent to which two random variables vary together and can be characterized by positive, negative, or zero values. A positive covariance indicates a positive relationship between the two variables, while a negative covariance indicates a negative relationship. If two elements do not vary together, they will display a zero covariance.

Supposing to have two variables X and Y, the covariance can be computed using the following formula:

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

where:

- ✓ $Cov(X,Y)$: covariance between X and Y variables.
- ✓ $x_i \; and \; y_i$: member of X and Y variables or of the available dataset.
- ✓ $\bar{x}$ and $\bar{y}$: mean of X and Y variables or of the available dataset.
- ✓ n : total number of members or observations.

The number of features in a dataset determines the dimension of the variance-covariance matrix. Therefore, the covariance matrix for the entire dataset will be a square matrix with d × d dimensions, where d is the number of features in the dataset. In the given toy example, there are three features: feature1, feature2, and feature3, which can be associated with x, y, and z, respectively. The covariance matrix for the toy example is defined as follows:

$$\text{Cov(A)} = \begin{vmatrix} Var(x) & Cov(x,y) & Cov(x,z) \\ Cov(x,y) & Var(y) & Cov(y,z) \\ Cov(x,z) & Cov(y,z) & Var(z) \end{vmatrix} = \begin{vmatrix} 900 & 450 & 450 \\ 450 & 300 & 150 \\ 450 & 150 & 300 \end{vmatrix}$$

The elements belonging to the diagonal of the covariance matrix, represent the variance of scores for each feature. The feature1 has the biggest variance (600) while feature2 and feature3 are characterized by the same variance value (200), meaning that the feature1 scores have more variability with respect to the other two features. Once the covariance matrix has been calculated is possible to proceed with the next step of PCA full process with the objective to compute Eigenvectors and corresponding Eigenvalues through the application of the following linear transformation to the covariance matrix:

$$\det(\text{Cov(A)} - \lambda I) = 0$$

where λ is a scalar of the linear transformation and I the identity matrix. The determinant can be evaluated by expanding the linear transformation:

$$\det\left(\begin{bmatrix} 900 & 450 & 450 \\ 450 & 300 & 150 \\ 450 & 150 & 300 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}\right) = \det\left(\begin{bmatrix} 900-\lambda & 450 & 450 \\ 450 & 300-\lambda & 150 \\ 450 & 150 & 300-\lambda \end{bmatrix}\right)$$

After simplifying the linear transformation, it is possible to find the determinant of the matrix to solve the linear transformation equation.

$$-\lambda^3 + 1500\lambda^2 - 202500\lambda = 0$$

The eigenvalues of the matrix can be computed equating the above equation to zero and solving it for λ, therefore, to compute the eigenvalues of a matrix is sufficient to find the zeros of the characteristic polynomial of the matrix, obtaining the following solutions:

$$\lambda_1 = 1350,\ \lambda_2 = 150,\ \lambda_3 = 0$$

The eigenvectors associated with each eigenvalue can be obtained from the above calculations, resulting in the following solution for the corresponding eigenvectors.

$$v_1 = \begin{pmatrix} 0.816 \\ 0{,}408 \\ 0{,}408 \end{pmatrix}, \qquad v_2 = \begin{pmatrix} 0{,}577 \\ -0{,}577 \\ 0{,}577 \end{pmatrix}, \qquad v_3 = \begin{pmatrix} 0 \\ -0.707 \\ 0{,}707 \end{pmatrix}$$

The next step involves sorting the eigenvectors by decreasing eigenvalues and selecting k eigenvectors with the largest eigenvalues to obtain a d × k dimensional matrix W. This step is necessary to reduce the dimensionality of the feature space by projecting it via PCA onto a smaller subspace. The eigenvectors obtained in the previous step will form the axes of this new feature subspace.

Although the eigenvectors define the directions of the new axis, they all have the same unit length of 1. Therefore, it is necessary to examine the corresponding eigenvalues of the eigenvectors to determine which eigenvectors should be dropped for the lower-dimensional subspace.

In general, the eigenvectors with the lowest eigenvalues carry the least amount of information about the data distribution and are therefore the ones that should be dropped. The typical approach is to rank the eigenvectors by their corresponding eigenvalues in descending order and select the top k eigenvectors. This involves sorting the eigenvalues in decreasing order.

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \begin{pmatrix} 1350 \\ 150 \\ 0 \end{pmatrix}$$

In the given toy example, the objective is to reduce a 3-dimensional feature space to a 2-dimensional feature subspace. Therefore, the two eigenvectors with the highest eigenvalues must be selected to construct the d × k dimensional eigenvector matrix W. The eigenvectors corresponding to the two largest eigenvalues are as follows:

$$W = \begin{bmatrix} 0{,}816 & 0{,}577 \\ 0{,}408 & -0{,}577 \\ 0{,}408 & -0{,}577 \end{bmatrix}$$

At the end, to compute the two principal components and project the data points onto the new subspace, the 3x2 dimensional matrix W just computed, it is used to transform the samples onto the new subspace via the equation y = W′ * x where W′ is the transpose of the W matrix.

$$y = W' * x = \begin{bmatrix} 367{,}42 & -183{,}71 & -183{,}71 \\ 0 & -106{,}06 & 106{,}06 \end{bmatrix}$$

The results presented in the toy example were obtained using the following Python script. The script was designed to compute the covariance matrix and its associated eigenvalues and eigenvectors, with the aim of reducing a 3-dimensional feature space to a 2-dimensional feature subspace.

```
# # Principal Component Analysis
import numpy as np, array
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
from sklearn import decomposition
from sklearn.decomposition import PCA

data=np.array([[90, 30, 60], [90, 60, 90], [90, 60, 60]])
cov_matrix = np.cov(data)
print("covariance matrix is", cov_matrix)
np.linalg.eig(cov_matrix)
eigenvalue, eigenvector =np.linalg.eig(cov_matrix)
print("Eigenvalues are:",eigenvalue)
print("Eigenvector are:",eigenvector)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(cov_matrix[:,0], cov_matrix[:,1], cov_matrix[:,2], cmap = "viridis", marker='o')
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.title('Representation in 3D before PCA')
plt.show()
# create the PCA instance
pca = decomposition.PCA(n_components=2)
# fit on data
pca.fit(cov_matrix)
# access values and vectors
print(pca.components_)
print(pca.explained_variance_)
# transform data
B = pca.transform(cov_matrix)
print(B)
sns.scatterplot(data=B)
plt.title('Representation in 3D after PCA')
```

```
plt.show()
```

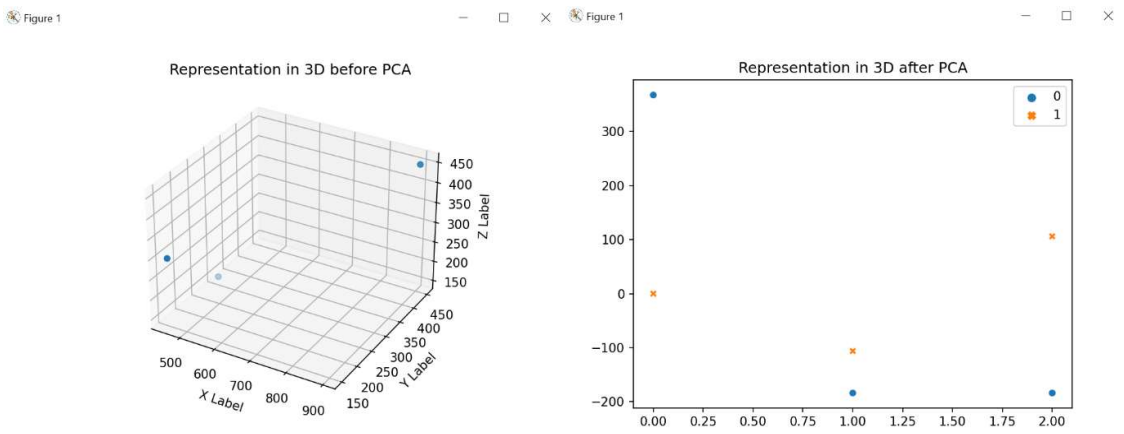


Figure 40 – PCA: Toy example 3-D to 2-D dimensionality reduction

4.2.2 T-SNE: Dimensionality reduction in details

T-SNE, or t-distributed Stochastic Neighbour Embedding, is a technique used for dimensionality reduction, primarily for visualizing data in 2D and 3D maps. This method uses non-linear dimensionality reduction to identify non-linear relationships in the data. It is particularly useful for separating data that cannot be separated by a straight line, making it a popular choice for dimensionality reduction.

T-SNE is an iterative method, unlike PCA, and cannot be applied to another dataset. PCA uses the global covariance matrix to reduce data, and the same result can be obtained by applying the covariance matrix to a new set of data. This is useful when it is necessary to reduce the feature list and reuse the matrix created from training data. In contrast, T-SNE is primarily used to understand high-dimensional data and project it into a low-dimensional space such as 2D or 3D.

When working with data that has more than 2 or 3 features, it is important to check for any clusters in the data. This information can be useful in understanding the data and, if necessary, in selecting the number of clusters for clustering models such as k-means.

Assuming we have data in a 2D space, as shown in below figure, our objective is to reduce its dimensionality to a 1D space.

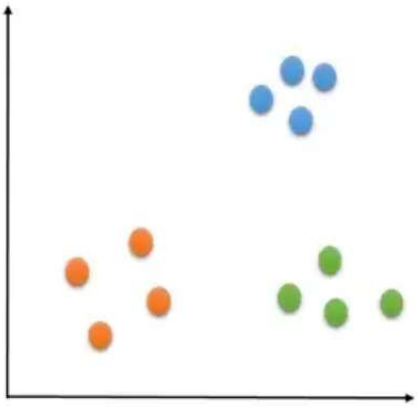


Figure 41 – T-SNE: Data represented in 2D

Each colour in the above figure represents a cluster with a different density. Projecting the data onto just one dimension, we would obtain an overlap of at least two clusters, depending on the axis chosen for the projection. This is illustrated in the below picture, and it results in some uncertainty in the dimensionality reduction process. T-SNE method can help overcome this issue.

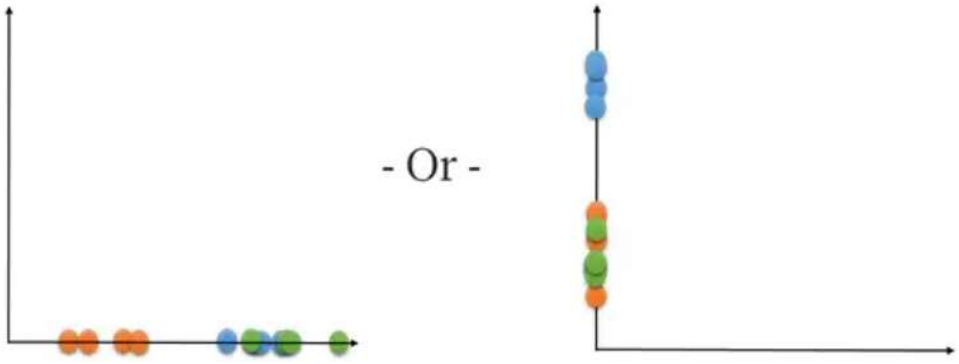


Figure 42 – T-SNE: Data projected in 1D

Basically, the T-SNE method operates with the following three stages:

- ✓ Computation of a joint probability distribution that represents the similarities between the data points.
- ✓ Creating a dataset of points in the target dimension and then calculating the joint probability distribution for them.

- ✓ Using gradient descent to transform the dataset in the low-dimensional space such that the joint probability distribution representing it would be as similar as possible to the one in the high-dimensional space.

The first step of the T-SNE algorithm involves calculating the Euclidean distance between each point and all the other points in the dataset. These distances are then transformed into conditional probabilities that represent the similarity between every pair of points, indicating how likely they are to be neighbours.

The conditional probability of point x_j to be close to point x_i is represented by a Gaussian centred at x_i with a standard deviation of σ_i. For nearby datapoints, $p_{j|i}$ is relatively high, whereas for widely separated datapoints, $p_{j|i}$ will be almost infinitesimal (for reasonable values of the variance of the Gaussian, σ_i). Mathematically, the conditional probability $p_{j|i}$, meaning the probability of point x_i to have x_j as its neighbour is given by:

$$p_{j|i} = \frac{e^{-\frac{\left||x_i - x_j|\right|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\left||x_i - x_k|\right|^2}{2\sigma_i^2}}}$$

Clusters can be characterized by different densities, therefore, to take care of it, the denominator of the above formula is normalized by the sum of all the other points placed at the Gaussian centred at x_i. As depicted on Figure 41 the density of the orange cluster, for instance, is lower than the density of the blue cluster, therefore computing the similarities of each two points by a Gaussian only should provide lower similarities between the orange points compared to the blue ones.

The joint probability distribution can be formulated from the conditional distributions created using the following equation:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

Using the joint probability distribution rather than the conditional probability is one of the improvements in the method of t-SNE relative to the former SNE owing to the symmetric property of the pairwise similarities ($p_{ij} = p_{ji}$) which helps simplify the calculation at the third stage of the algorithm.

Objective of the second stage is to create a dataset of points in a low-dimensional space evaluating the joint probability distribution of the points belonging to the dataset created.

In the T-SNE algorithm, a random dataset of points with the same number of points as the original dataset is created. This random dataset has K features representing the target dimension. In the toy example mentioned, the objective is to reduce the dimensionality to a 1D space, so K is equal to one. However, if the purpose is to use dimension reduction for visualization, K may be 2 or 3.

The most straightforward option to achieve the goal of finding a similar probability distribution in the low-dimensional space is to use Gaussian distribution. However, this is not the best approach due to certain issues, such as crowding. Therefore, to overcome this problem, the T-SNE algorithm uses Student t-distribution with a single degree of freedom. This helps to address the "short tail" characteristic associated with the Gaussian distribution [2].

On this stage, marking the probabilities by q, and the points by y the conditional probability $q_{j|i}$ is expressed by:

$$q_{j|i} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

By using Student t-distribution instead of Gaussian distribution, the T-SNE algorithm is better able to handle randomly distributed values. This is because the student t-distribution drops quickly and has a "long tail," which ensures that points are not compressed into a single point, thus reducing uncertainty in the results.

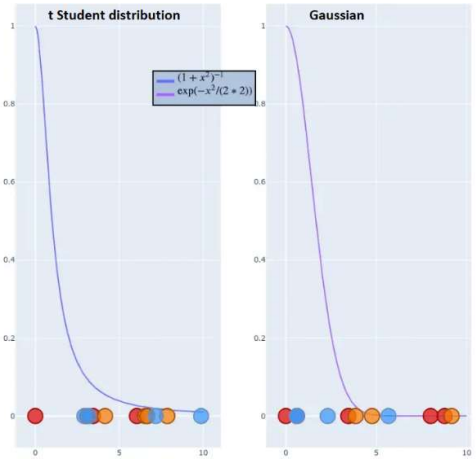


Figure 43 – T-SNE: Gaussian Vs t-Student Distribution tails

The T-SNE algorithm uses t-distribution instead of Gaussian distribution because of the heavy tails' property of the former. This property enables moderate distances between points in the high-dimensional space to become extreme in the low-dimensional space, thus preventing the "crowding" of points in the lower

dimension. Additionally, using t-distribution improves the optimization process in the third part of the algorithm.

The final step of the T-SNE algorithm involves using gradient descent to transform the dataset in the low-dimensional space such that the joint probability distribution representing it is as close as possible to the one in the high-dimensional space.

Kullback-Leibler divergence between the conditional probabilities $p_{j|i}$ and $q_{j|i}$, which measures how much two distributions are different from one another, is applied with the purpose to make the joint probability distribution of the data points in the low dimension as similar as possible to the one from the original dataset.

For distributions P and Q in the probability space of χ, the Kullback-Leibler divergence is defined by:

$$D_{KL}(P||Q) = \sum_{x \in \chi} P(x) log\left(\frac{P(x)}{Q(x)}\right)$$

The similarity between the two distributions is measured using the KL divergence, with a smaller value indicating greater similarity, and a value of zero indicating identicalness. The gradient descent technique minimizes the KL divergence as the cost function, and this allows the lower dimensional dataset to be transformed such that its joint probability distribution is as similar as possible to the one from the original data. The cost function for the gradient descent is the KL divergence between P and Q, which represent the joint probability distributions of the high and low dimensions, respectively. It is given by:

$$C = KL(P||Q) = \sum_{i} \sum_{j} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

By minimizing the cost function, the T-SNE algorithm optimizes the values of the points in the low-dimensional dataset, which can then be used for visualization purposes. Referring to the example mentioned earlier in this paragraph, the application of the T-SNE method separates the clusters in the low-dimensional space, eliminating the uncertainty and overlap that may have resulted from projecting the data onto the axes.

Figure 44 – T-SNE: Data represented in 1D

We have discussed the key steps involved in two dimensionality reduction techniques: Principal Component Analysis (PCA) and t-distributed Stochastic Neighbour Embedding (t-SNE). These techniques are typically used for datasets where the points can be separated in a linear or non-linear way, respectively.

For illustration purposes and to compare the visualization results of these techniques, we used the DIGITS and MNIST datasets from the literature. The DIGITS dataset consists of 65 features, while the MNIST dataset has 785 features. In both cases, we performed feature dimensionality reduction to obtain a representation with only 2 features. The resulting visualizations are presented below.

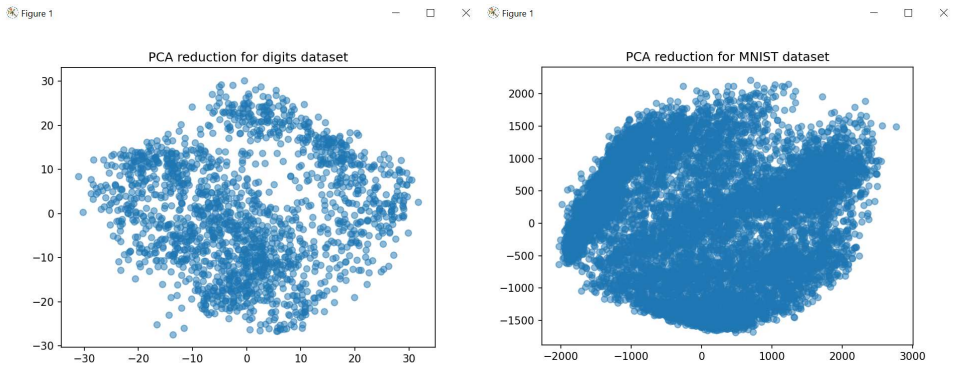


Figure 45 – PCA: representation of Digits and MINST dataset

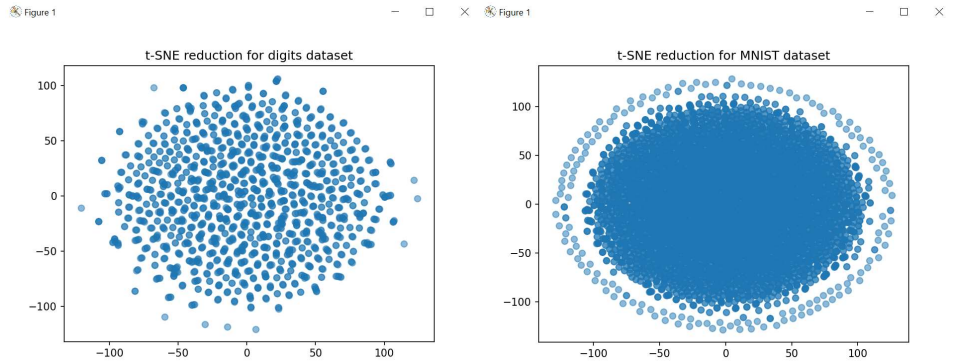


Figure 46 – T-SNE: representation of Digits and MINST dataset

PCA performs reasonably well on the DIGITS dataset, identifying a clear structure. However, on the MNIST dataset, it encounters a problem with "crowding." In contrast, the T-SNE algorithm is less affected by this issue and produces good results for both the DIGITS and MNIST datasets.

All the results presented in this paragraph were obtained using the following Python script, which was designed to compare the performance of PCA and T-SNE in reducing the dimensionality of two different input datasets.

```
# Load Python Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from sklearn import decomposition
# Read in Data
digits = pd.read_csv('./digits.csv')
mnist = pd.read_csv('./fashion-mnist_test.csv')
print("Dimensions of Data => ", digits.shape, mnist.shape)
#PCA reduction dimensionality algorithm
pca = decomposition.PCA(n_components=2)
X_PCA_dig=pca.fit_transform(digits)
X_PCA_mnist=pca.fit_transform(mnist)
print(X_PCA_dig.shape, X_PCA_mnist.shape)
# Creating a scatter plot of the datapoints
plt.scatter(X_PCA_dig[:, 0], X_PCA_dig[:, 1], cmap="random", alpha=0.5)
plt.title('PCA reduction for digits dataset')
plt.show()
plt.scatter(X_PCA_mnist[:, 0], X_PCA_mnist[:, 1], cmap="random", alpha=0.5)
plt.title('PCA reduction for MNIST dataset')
plt.show()
#t-SNE reduction dimensionality algorithm
X_TSNE_dig = TSNE(n_components=2, learning_rate='auto',init='random',
perplexity=1).fit_transform(digits)
# Creating a scatter plot of the datapoints
plt.scatter(X_TSNE_dig[:, 0], X_TSNE_dig[:, 1], cmap="random", alpha=0.5,)
plt.title('t-SNE reduction for digits dataset')
plt.show()
X_TSNE_mnist = TSNE(n_components=2, learning_rate='auto',init='random',
perplexity=1).fit_transform(mnist)
# Creating a scatter plot of the datapoints
plt.scatter(X_TSNE_mnist[:, 0], X_TSNE_mnist[:, 1], cmap="random", alpha=0.5,)
plt.title('t-SNE reduction for MNIST dataset')
plt.show()
```

4.3 Clustering technique

Clustering is an unsupervised machine learning technique that involves grouping a dataset into clusters such that objects in the same cluster are like each other, while those in different clusters are dissimilar. This technique is commonly used in data mining to explore data, identify patterns, and organize large datasets into meaningful groups. The objective of clustering is to divide a database of n objects into K groups based on their similarity. Objects in the same group should have high similarity, while those in different groups should have low similarity. The primary steps involved in clustering are as follows:

- ✓ Pattern representation (optionally including feature extraction or selection).
- ✓ Definition of a pattern proximity measure appropriate to the data domain.
- ✓ Grouping.
- ✓ Data extraction.
- ✓ Cluster validity.

Pattern representation in clustering is closely related to the number of available patterns and the number of classes. Feature selection is a technique used to identify the most effective subset of the original features for clustering. This process can reduce training times and minimize variation in the data.

The general concept behind feature selection is that the dataset often contains redundant features that can be removed without losing important information. Feature extraction involves creating new features by performing operations on the original features, while feature selection returns a subset of these features.

The proximity between patterns in clustering is determined by minimizing a distance criterion. However, selecting an appropriate distance metric can be challenging, as it significantly influences the shape of the resulting clusters and the quality of the clustering. The distance metric can either be chosen manually by a human expert or learned by an algorithm from the data.

A distance function, also called a distance metric, is a function $f: X \times X \longrightarrow \mathbb{R}$ that defines a distance between each pair of elements of a set X. A set with a metric is called a metric set. For each $x, y, z \in X$, a metric d satisfies the following properties:

- ✓ Non negativity: $d(x, y) \geq 0$.
- ✓ Identity of indiscernible: $d(x, y) = 0 \Leftrightarrow x = y$.
- ✓ Symmetry: $d(x, y) = d(y, x)$.

- ✓ Triangle inequity: d (x , z) ≤ d (x , y) + d (y , z).

The most common distance, which is used for clustering objective, is the Euclidean distance. In general, if we have p variables X_1, X_2,… X_p measured on a sample of n subjects, the observed data for subject i can be denoted by x_{i1}, x_{i2},… x_{ip} and the observed data for subject j by x_{j1}, x_{j2},… x_{jp}. The Euclidean distance between these two subjects can be expressed as below:

$$d_{i,j} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$

The main advantage about this distance is that it does not depend on the addition of new objects to the analysis, which may be outliers. Against this, Euclidean distance is affected from differences in scale among the dimensions from which the distances are computed and therefore, if one variable has a much wider range than others then this variable will tend to dominate. To get around this problem each variable can be standardized.

An alternative concept to that of the distance is the similarity function $s(x_i , x_j)$, that compares the two vectors x_i and x_j . This function should be symmetrical ($s (x_i , x_j) = s (x_j , x_i)$), and has a large value when x_i and x_j are somehow similar and constitute the largest value for identical vectors. A similarity function where the target range is [0, 1] is called a dichotomous similarity function. Specifically, the cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. This metric is a measure of orientation and not magnitude. The cosine similarity between the vector x_s and x_t can be expressed as below:

$$Similarity\ (x, y) = \frac{x_s x'_t}{\sqrt{(x_s x'_s)(x_t x'_t)}}$$

The cosine distance between two points is one minus the cosine of the included angle between points (treated as vectors). This equation is following:

$$D = 1 - \frac{x_s x'_t}{\sqrt{(x_s x'_s)(x_t x'_t)}}$$

As mentioned before, cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Thus, it is a judgement of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1 and two vectors diametrically opposed have a similarity of -1, independently of their magnitude.

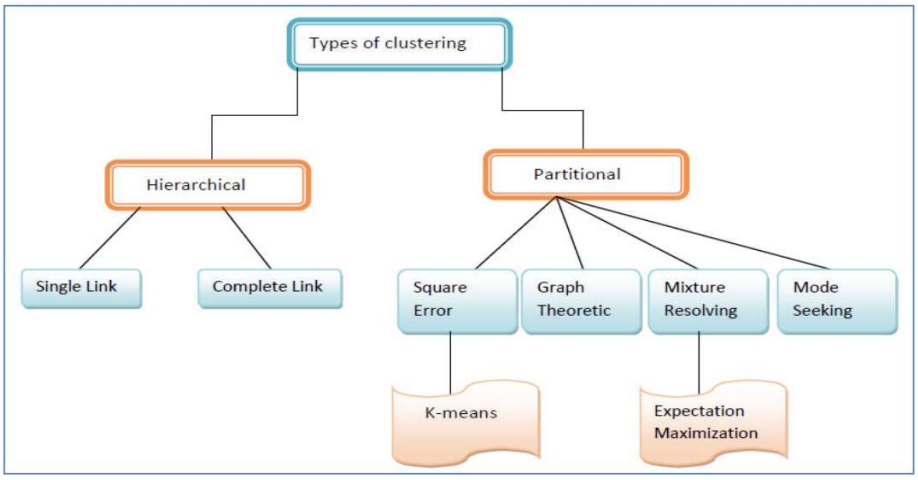


Figure 47 – Clustering techniques

The clustering process involves several implementation steps that vary depending on the type of clustering used. For instance, partitioning methods like k-means algorithm optimize a clustering criterion to identify a partition, while fuzzy clustering assigns a degree of membership to each object in each cluster. Hierarchical methods create a hierarchical decomposition of the data objects.

Data abstraction is the process of reducing a dataset to a simplified representation of the whole by removing non-essential features. In clustering, data abstraction typically involves creating a compact description of each cluster, often in terms of cluster prototypes or representative patterns such as the centroid.

Cluster validity is a crucial step that follows the clustering process and assesses the quality of the results. Cluster validity determines what makes one clustering result better than another by evaluating the quality of the clustering process.

4.3.1 Hierarchical clustering algorithm

Hierarchical clustering algorithms are designed to create a hierarchy within the data, represented by a dendrogram that indicates the number and size of the groups allowed. These algorithms typically fall into two categories: sequential fission (agglomeration) or fusion (division) methods. Agglomerative clustering methods construct the hierarchy level by level, starting from the bottom, while divisive clustering methods work from the top down.

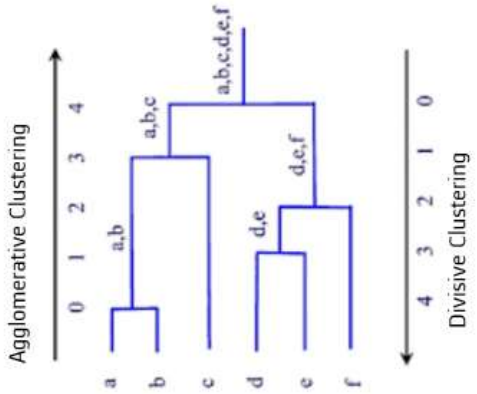


Figure 48 – Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up approach where each data point starts in its own cluster and is then recursively grouped into larger clusters. In contrast, divisive hierarchical clustering is a top-down approach that starts with one cluster containing all the data points and then recursively divides it into smaller clusters.

One significant advantage of hierarchical clustering is that it does not require the number of clusters to be specified beforehand, and the resulting dendrogram can provide a meaningful taxonomy of the dataset. However, the main disadvantage of hierarchical clustering is its high computational complexity. Agglomerative hierarchical clustering has a time complexity of O(n3), while divisive hierarchical clustering has a time complexity of O(2n), making both algorithms infeasible for larger datasets.

4.3.2 Fuzzy clustering algorithm

Clustering can be categorized as either Soft clustering (Overlapping Clustering) or Hard Clustering (Exclusive Clustering). In hard clustering, each object must either belong to one cluster or not. In contrast, soft clustering allows objects to belong to two or more clusters with varying degrees of membership. In this case, the data is associated with an appropriate membership value, meaning that each cluster contains memberships characterized by a degree value between 0 and 1.

The Fuzzy C Means (FCM) algorithm is the most widely used algorithm for soft clustering. FCM is a data clustering technique that assigns each object to a cluster based on a membership grade, indicating the degree of belonging to that cluster. The algorithm was first introduced by Jim Bezdek in 1981.

A basic difference between FCM and K-means is that FCM is taking more time for computation than that of K-means. The time complexity of K-mean algorithm is O(ndci) and time complexity of FCM is $O(ndc^2i)$ [5], where n represents the number of data points, c the number of clusters, d the number of dimension and i the number of iterations.

4.3.3 K-means: Partitional clustering algorithm

Partitional clustering involves dividing data into a fixed number of disjoint clusters. The primary objective is to minimize the dissimilarity between samples within each cluster while maximizing the dissimilarity between clusters. This approach is often called centroid based clustering since clusters are typically represented by their centroid.

K-Means is the most used partitional clustering method and has been popular since it was first published in 1955 [1]. It remains one of the simplest and most effective clustering algorithms, largely due to its ease of implementation, efficiency, and empirical success. This is a method that partitions a dataset into k clusters, aiming to minimize the sum of squared distances within each cluster. This approach is defined by its objective function, which seeks to minimize the sum of squared distances between each data point and its assigned cluster centre.

The objective function is defined as:

$$arg \min_{S} \sum_{i=1}^{k} (\sum_{x_j \in S_i} ||x_j - \mu_i||^2)$$

where, x_j is a data point in the dataset, Si is a cluster (set of data points) and μ_i is the minimizing this objective function leads K-means to converge to the global optimum when clusters are well separated.

One unique aspect of k-means that distinguishes it from other clustering methods is that the number of clusters is predetermined before clustering occurs. This can be viewed as both a strength and a weakness. One advantage of a fixed number of clusters is that the k-means method does not introduce new clusters in the presence of anomalous data points. Instead, it assigns the anomalous point to the closest cluster.

One limitation of using a fixed number of clusters in k-means is that it may not always be clear how many clusters a dataset contains. Choosing an unsuitable value of k can lead to poor results, rendering the k-means method unusable. Therefore, a crucial step for any unsupervised algorithm is to determine the optimal number of clusters into which the data can be clustered.

The Elbow method is a popular approach for determining the optimal value of k in clustering. This method provides a visual way of selecting the best value for k. The Elbow method works by measuring the sum of squared errors for various numbers of clusters. The sum of squared errors represents the sum of the squared distances of each data point from its cluster's centroid. By plotting the sum of squared errors for each number of clusters, one may observe a point where the slope goes from steep to shallow, indicating a decrease in the error sum. This point is known as the elbow point and represents the optimal number of clusters that can be used as input for the K-means algorithm.

Figure 49 illustrates an example of the elbow method process, where the curve shows that the greatest reduction in error sum occurs at around 30 clusters. The curve bends sharply at this point, indicating that increasing the number of clusters beyond 30 leads to only small decreases in error sum. Therefore, 30 is likely the optimal number of clusters for this dataset.

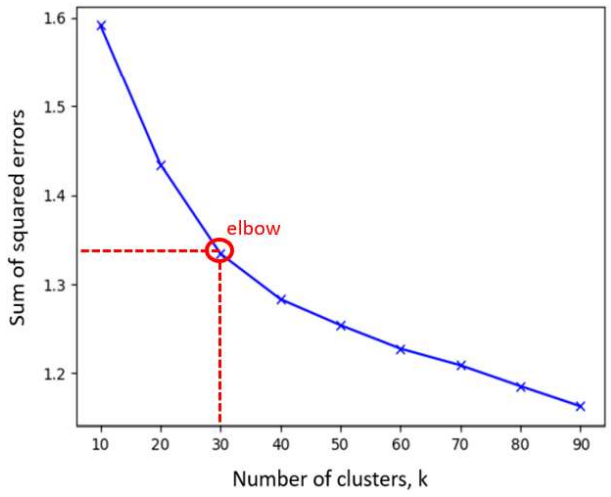


Figure 49 – Elbow method showing optimal k

In addition to selecting the optimal number of clusters k, the K-means algorithm requires two additional user specified parameters: cluster initialization and distance metrics. However, one disadvantage of this method is the sensitivity of results to the initialization of cluster centroids. Different locations of cluster

centres can result in different clustering outcomes, leading to standard errors that are challenging to correct in statistical analysis.

K-means is commonly used with the Euclidean distance metric for computing the distance between points and cluster centres, resulting in the formation of spherical or ball-shaped clusters. However, for high-dimensional data, the Euclidean distance may be less meaningful in a spherical space. In such cases, alternative distance metrics like Cosine similarity or Pearson correlation are preferred, and the spherical k-means algorithm may be used instead. This algorithm employs an iterative refinement process to generate its final clustering based on the number of clusters specified by the user and the dataset. The optimal number of clusters can be determined using methods such as the Elbow method, and the K-means algorithm will iterate until the desired number of clusters is obtained.

At the beginning of the K-means algorithm, k centroids are randomly selected as the mean values of k clusters, and the nearest data points to these centroids are assigned to form k clusters. The algorithm then iteratively recalculates the new centroids for each cluster until it converges to an optimal solution. K-means clustering is best suited for numerical data with a low number of dimensions, as the algorithm relies on computing mean values for the data. Therefore, numerical data with a relatively small number of dimensions is most appropriate for K-means clustering.

The algorithm works as below:

1. The centroids of k clusters are initialized randomly with K points based on a predefined value of k.
2. To create the k clusters, each data point in the dataset is assigned to the nearest centroid based on distance. The Euclidean distance is often used to calculate the distance between each data point and the initialized centroids.
3. The centroids are recalculated by taking the average of all the data points assigned to each cluster, which helps to minimize the total intra-cluster variance.
4. Steps 2 and 3 are repeated iteratively until certain criteria are met. Typical criteria include when there are no further changes in the centroid values, when the sum of distances between the data points and the centroid of each cluster no longer changes, when the data points assigned to the clusters are the same as in the previous assignment, or when the maximum iteration number has been reached (in cases where the algorithm is given a fixed iteration time).

One advantage of the K-means algorithm is its ease of implementation and low computational complexity, which involves only computing and comparing distances among data points and grouping clusters. This

makes K-means computationally faster than hierarchical clustering, with a time complexity of O(n), where n is the number of data samples. Moreover, K-means can scale up to large datasets and is easily adaptable to new data samples.

On the other hand, one disadvantage of the K-means algorithm is that the results may differ from one execution to another, lacking consistency due to its dependence on the number of clusters, which must be specified manually, and the initialization of centroids. K-means also struggles with clustering datasets of varying sizes and densities and cannot identify outliers, which may affect the clustering process.

4.4 Machine Learning process mindset

Understanding the various stages involved in a machine learning process is crucial when dealing with unknown and often large datasets. This knowledge enables one to properly prepare the input data and select suitable techniques and algorithms to solve problems such as clustering, classification, regression, or dimensionality reduction.

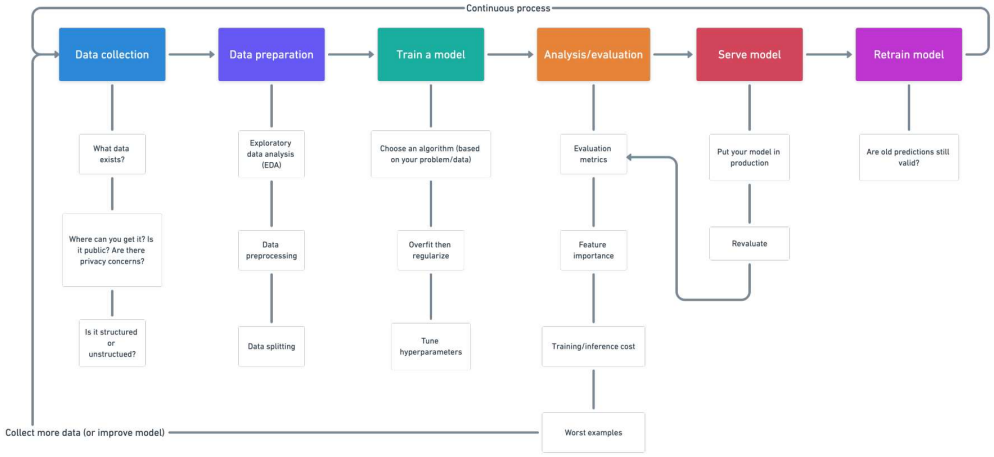


Figure 50 – Machine Learning process

The purpose of the data collection phase is to gain an understanding of the types of data that are available and need to be managed by the machine learning algorithm. Input data can be structured if it is formatted in a tabulated style or unstructured if there is no rigid structure.

To create a data dictionary for the features, it is necessary to understand the type of data being managed. The available data is often in raw format, so it must be manipulated to be fed to the machine learning

algorithm that will be chosen to solve the problem of interest. The first step of the data preparation stage involves exploring the data through specific analyses with the objective of identifying the feature variables in the input and the target variables in the output, and determining whether there are missing values, redundant information in terms of features, or outliers stored in the data of interest. Once knowledge about the data has been acquired, it may be necessary to pre-process the data to be modelled. Depending on the content of the dataset, it may be necessary to use one of the processing techniques listed below:

- ✓ Feature imputation: For machine learning to learn from data, missing values must be filled.
- ✓ Feature encoding: Values must be converted to numerical form to satisfy the constraint that all values used in machine learning models must be numerical.
- ✓ Feature normalization or standardisation: Data scaling is necessary to ensure that all features are represented on the same scale and to avoid some features from dominating others. Feature scaling, also known as normalization, involves shifting values so that they fall between 0 and 1. This is achieved by subtracting the mean value and dividing by the range between the maximum and minimum values. Feature standardization, on the other hand, standardizes all values to have a mean of 0 and a unit variance. This is done by subtracting the mean and dividing by the standard deviation of that feature. The resulting values may not fall between 0 and 1. Standardization is more robust to outliers than feature scaling.
- ✓ Feature engineering: Transforming data into potentially more meaningful representations can be achieved by incorporating domain knowledge. This transformation can be implemented through techniques such as decomposition, discretization, which involves grouping larger groups into smaller ones, combining two or more features, or using other parts of the data to indicate potentially more significant information.
- ✓ Feature selection: Selecting the most important features of a dataset for modelling can reduce overfitting, training time, and improve accuracy by reducing the overall amount of data to be trained. Dimensionality reduction techniques such as Principal Component Analysis (PCA) can accomplish this by taking many features and using linear algebra to reduce them to fewer dimensions.
- ✓ Feature balancing: To prevent imbalances in the features within the available data.

The final step in data preparation is data splitting, where the data is divided into a training set to be used for model learning, a validation set to tune hyperparameters, and a test set for evaluating the performance of the model.

After the data has been pre-processed and an algorithm and learning type have been selected to match the problem of interest, the training phase can begin. This is the central part of the machine learning process, during which the algorithm can recursively learn from the data and produce the desired outcome.

The next step, which is equally important as all the previous phases, involves analysing and evaluating the performance of the model by introducing evaluation metrics such as:

- ✓ Confusion Matrix.
- ✓ Accuracy.
- ✓ Precision.
- ✓ Recall.
- ✓ F1-score.

A Confusion matrix is a matrix of size N x N utilized for evaluating the performance of a classification model, where N represents the number of target classes. It compares the actual target values with the predicted values generated by the machine learning model, providing a comprehensive overview of the model's performance and the types of errors that occurred.

For the sake of illustration, let us consider a binary classification problem. The Confusion Matrix in this case consists of two rows and two columns, representing the four possible combinations of actual and predicted values.

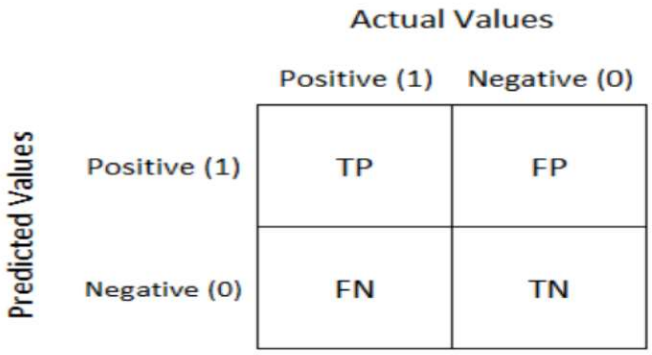


Figure 51 – Confusion Matrix for binary class

When interpreting the Confusion Matrix, the target variable can have two values: Positive and Negative. The columns represent the actual target variable values, while the rows of the matrix correspond to the predicted values of the target variable. The values within the Confusion Matrix are characterized as follows:

- ✓ True Positive (TP): The predicted value matches the actual value meaning that the actual value is positive, and the model predicts a positive value.
- ✓ True Negative (TN): The predicted value matches the actual value meaning that the actual value is negative, and the model predicts a negative value.
- ✓ False Positive (FP): The value is wrongly predicted meaning that the actual value is negative, but the model predicts a positive value.
- ✓ False Negative (FN): The value is wrongly predicted meaning that the actual value is positive, but the model predicts a negative value.

Having all these values allows for the calculation of other essential elements in evaluating model performance, such as accuracy, precision, recall, and F1-score.

Accuracy represents the proportion of correctly predicted values out of all available classes and can be expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is a metric that measures the proportion of true positives out of all classes predicted as positive. It is a useful evaluation metric when false positives are more concerning than false negatives, and it can be calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

Recall measures the proportion of true positives out of all actual positive classes, and it is typically more relevant when false negatives are more concerning than false positives. The calculation for recall is as follows:

$$Recall = \frac{TP}{TP + FN}$$

The F1-score can be viewed as the harmonic mean of Precision and Recall, providing a combined measure of these two metrics. It reaches its maximum value when Precision is equal to Recall.

$$F1 - score = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

The final two phases of a typical machine learning process involve deploying the model in production to observe its behaviour in the real world and retraining the model based on feedback received from the field.

5 Chapter five: Detection for Railway equipment

5.1 Second topic of the research

Despite the inspiration that arose from attending the "Silvano Pupolin" Information Engineering Summer School in July 2021, where machine learning techniques were presented in various application domains, the second topic of this research is closely related to the first, which was discussed in previous chapters and remains of great importance for railway industry purposes.

The first topic of the research aimed to utilize a public network based on Radio Frequency technology instead of a wired private network to facilitate mission and safety critical communications between a central node and various peripheral nodes located along the railway line. This approach was intended to reduce infrastructure costs.

The objective of the second topic is to use the same wireless channel to collect all the logs and measurements of on field railway equipment for maintenance purposes.

Maintenance of railway systems is one of the most challenging tasks faced by railway infrastructure managers across the globe, owing to the high effort required in terms of time and costs. Reliability is critical for railway services or assets, which necessitates keeping the equipment in good working order. Regular maintenance of railway systems is therefore essential to achieve this objective. Consequently, innovative maintenance solutions and the integration of maintenance into operations are continually being studied and developed to facilitate better railway equipment and rolling stock management.

Railway maintenance refers to the process of preserving the condition or state of being preserved on the track or related to vehicles. However, there are various approaches to maintaining a railway asset in service, as described in [8]:

- ✓ Corrective Railway Maintenance: This task involves identifying, isolating, and resolving faults so that failed equipment can be replaced or restored to operational condition within the established tolerances or limits for in service operations. With this approach, no actions are taken to prevent faults, as the only way to detect them is by waiting for equipment to fail.

- ✓ Preventive Railway Maintenance: This task involves regularly monitoring the status or condition of railway equipment to reduce the likelihood of failure. Such an approach allows infrastructure managers to identify a decline in the health of an asset, enabling timely preventive action to be taken before a failure occurs, and thereby ensuring better overall system reliability.
- ✓ Predictive Railway Maintenance: Predictive maintenance techniques are intended to determine the condition of in-service equipment to predict when it will fail, ensuring highly detailed forecasts rather than just likelihoods. This approach offers cost savings compared to routine or time-based maintenance, as tasks are only performed when necessary.

In conclusion, the aim of this second industrial research topic is to develop a model based on Machine Learning techniques, implemented through Python code (provided in the Appendix), to aid in the detection of nominal and anomalous states of railway equipment by analysing on field measurements. This will save time in identifying anomalies due to the large amount of data gathered at the central site from all railway equipment and reduce human errors in maintenance activities.

5.2 ML in literature for Railway application field

Before delving deeper into the research field, a preliminary literature study was conducted, revealing that many studies have been conducted on fault detection for railway equipment using machine learning techniques across all three families.

As previously discussed, machine learning can be classified into three types: supervised learning, unsupervised learning, and reinforced learning. In supervised learning, the predictors and response variables are known and used to build the model. In unsupervised learning, only the response variables are known. In reinforced learning, the agent learns actions and consequences by interacting with the environment.

The purpose of this section is to present an overview of the state of the art of the different machine learning families in the railway domain. Due to the need to process and use large datasets, Machine Learning (ML) approaches are providing promising solutions in almost every field of application, opening new horizons for the smart operations, management, and maintenance of transportation infrastructure such as the railway sector. Various models have been proposed in the literature to process large amounts of industrial data [11] or data collected from on field measurements or real time monitoring of railway trackside behaviour [10]. The objective is to predict faults in advance, as it has been demonstrated that equipment sound or its

operating state can exhibit features different from normal working conditions up to a week before a failure occurs [11].

In reference [11], a significant amount of historical sound data was collected and pre-processed using Short Time Fourier Transform (STFT) to convert the signal from "time-amplitude" to "time-frequency energy" format. A supervised learning approach was then used for hidden fault identification and classification. This was achieved through the implementation of a Convolutional Neural Network (CNN), which was found to achieve higher prediction accuracy compared to BP Neural Network and multi-classification Support Vector Machine (SVM) methods.

In reference [10], a large amount of data consisting of different curves evolving over time was processed using the Fast Dynamic Time Warping (Fast-DTW) methodology. The goal was to measure the similarity between time series curves generated by input data collected on railway turnout operating behaviour. Fast-DTW was used to generate delta curves as deviations from the reference curve. Unlike DTW, Fast-DTW enables the management of large amounts of input data by reducing the search space and time-consuming operations. However, it does not achieve the same level of accuracy unless the granularity is extended to obtain a more accurate warp path, which comes at the cost of increased model complexity. Although railway turnouts are among the most important trackside equipment in a railway system, enabling safe train movement between adjacent tracks, machine learning approaches have also been studied in cases where limited fault data is available [12].

In this case, as the problem to be addressed is characterized by a limited amount of available input data, a general deep learning approach based on neural networks that requires a large amount of training data is not recommended or feasible. Therefore, the objective of fault diagnosis for railway turnouts is addressed using the Deep Forest (DF) methodology. DF is a decision tree ensemble approach that features a cascading structure and multi-granularity scanning to address high-dimensional input problems such as the turnout power curve collected through on field measurements. It eliminates the need for segmenting the power curve and fully utilizes the available data once the features have been extracted through pre-processing activities.

Another application of this learning technique involves predicting fault detection in other railway trackside equipment, such as track circuits, using a novel solution based on acoustic data as input [13]. The acoustic signals are obtained by moving a cart equipped with dedicated microphones along the railway line and labelled by a railway track engineer. A comparison was performed between supervised learning models such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF),

and deep learning approaches such as Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN).

The Random Forest technique, which involves averaging multiple deep decision trees trained on different parts of the same training set to reduce variance, was found to be the most accurate in predicting faults, while the performance of the Convolutional Neural Network could potentially be improved by increasing the amount of input data. Similar conclusions were reached in studies conducted in related fields, such as data driven detection of Railway Point Machines Failures [14], prediction of railway assets [15], and automation of safety classification with respect to accidents at railway stations [16]. In all these studies, ensemble learning techniques were found to be effective in cases of limited fault data, with Random Forest achieving the highest accuracy compared to the Decision Tree family.

The Recurrent Neural Network (RNN) class, which is associated with an infinite impulse response, and the Convolutional Neural Network (CNN) class, which is associated with an infinite impulse, have been employed in other situations where the objective was to predict faults in advance, under the condition of non-limited input data evolving over time in a non-stationary environment, as discussed in [17] and [18]. The Long Short-term Memory (LSTM) Neural Network is also presented, which aims to address historical temporal dependencies by introducing a specialized memory cell to the network architecture. This cell allows a specific value to be remembered through a recurrent connection with itself.

The performance in terms of accuracy was compared between the Long Short-term Memory (LSTM) Neural Network and a Feedforward Convolutional Neural Network in learning track circuit faults such as insulated joint defect, mechanical rail defect, electrical disturbance, and ballast degradation, starting with input measurement data evolving over time. The LSTM network was found to exhibit better performance, achieving good accuracy in diagnosing every fault. However, it may be influenced by events that occurred a long time in the past if the memory size is not chosen appropriately. On the other hand, the Convolutional Neural Network showed weakness in detecting certain types of faults, such as electrical disturbances, but demonstrated more robustness in dealing with long temporal dependencies due to the absence of a memory cell.

A combination of the two supervised learning techniques could potentially be implemented to achieve additional benefits, although it would introduce additional complexity to the architecture and require more time and resources to train the model.

In [19], a different machine learning technique is proposed for high-speed railway systems to address the vanishing or exploding gradient problem caused by the long-term sequential data dependency of RNN, as well as the time-consuming complexity of standard LSTM and CNN. The new approach is still based on Neural Network and aims to predict fault detection in real time by introducing a variant of Long Short-term Memory (LSTM) called Gated Recurrent Unit (GRU). Due to the large scale of the system being monitored, a hierarchical solution has been proposed. This approach involves low-level sub-monitors that supervise the conditions of their local regions and a top-layer monitor that collects all the feedback received by the sub-monitors to achieve a conclusive evaluation of the entire system based on a voting strategy.

The solution described above has demonstrated the ability to predict faults in real time, outperforming other Neural Network models in terms of accuracy, runtime, and hardware resource utilization on FPGA. It achieves a trade-off between accuracy and efficiency in terms of energy consumption in a non-stationary environment [7]. The primary goal of implementing a system is to find a balance between plasticity and stability, which can be influenced by changing conditions over time. However, it is important to note that training a model can result in a significant carbon footprint, as indicated in [3] and [4].

Having efficient training algorithms is crucial, as learning can be distributed to the edge to prevent the carbon footprint caused by training algorithms and avoid economic barriers rather than environmental ones. In a non-stationary environment, where the independent and identically distributed assumptions no longer hold, the agent must learn over time, as environmental conditions may change due to seasonality and periodicity. This evolution over time can affect the system in terms of faults, malfunctioning, and communication problems, leading to an obsolete system in the absence of continuous training. Moving intelligence as close as possible to where data is generated, as in the case of Embedded and Edge systems, and designing models capable of dealing with time variant situations are possible solutions and research areas to be compliant with the hardware and environmental characteristics of the agent's fit. The evolution of IoT technology in the last decade has greatly influenced the relocation of models within devices that operate directly in the field, resulting in reduced decision latency, energy efficiency, and incremental and adaptive learning benefits since systems can directly acquire data from the field.

Battery powered devices, characterized by low computing ability, energy, and memory, pose significant challenges in designing algorithms to be run in embedded and edge AI. Creating a strong connection between hardware, software, and machine learning models is essential [6]. In situations where there is a large amount of unlabelled data and multiple unknown modes in normal conditions, with only a few samples in some modes, unsupervised techniques must be considered. In the railway application field, unsupervised

techniques have been proposed for diagnosing the faults of point machines [20], detecting anomalies in railway systems [21], and achieving early fault detection in predictive maintenance [22].

The papers propose various unsupervised learning algorithms, including K-means, C-means, Self-organizing Map, and Auto Encoder, each with its own specificity. Among these, Deep Auto Encoder (DAE) is promoted as the best solution in terms of performance and accuracy. K-means is widely used for classification problems, dividing the dataset into predetermined clusters based on Euclidean distance. On the other hand, C-means utilizes a more modular approach where each data point belongs to every cluster to some degree. Self-organizing map (SOM) is an unsupervised learning technique belonging to the artificial neural network (ANN) class, which is commonly applied to clustering problems and data exploration. Using the Self-organizing map (SOM) technique, the distribution of input data instances is represented in a one or two-dimensional array. This is achieved by using a set of finite neurons, each of which is characterized by two basic properties: position and connections to other neurons, known as neighbours.

Lower-dimensional spaces can be used to fit complex multidimensional and large datasets, thereby reducing complexity and the need for iterative learning processes.

Auto-encoder is a neural network belonging to the same family as feedforward Multilayer Perceptron. However, it differs in that it utilizes an unsupervised learning approach, as defined in [21], allowing the model to learn features from unlabelled data through an encoder function that attempts to learn a compressed representation of data, and a decoder that produces a reconstruction of the data. In the presence of unlabelled data, supervised machine learning approaches and the combination of unsupervised and supervised learning approaches fail to provide accurate classification, necessitating the use of a semi-supervised machine learning approach that can handle both labelled and unlabelled data. The semi-supervised learning technique can be divided into three different methods. The first method uses self-training, in which each classifier learns independently and relies solely on its own predictions. The second method uses co-training, as in [23] and [24] which are related to fault prediction and remaining useful life prediction in the railway domain, where two basic classifiers are trained from the data source, using the most confident unlabelled data to be added to the labelled data in the learning process.

The third method is graph-based, where each sample diffuses its label information to its neighbours until a globally stable state is achieved on the entire dataset. In [23], a semi-supervised co-training-based approach is used for bearing remaining useful life (RUL) prediction, combining a BP Neural Network able to fit any finite input output mapping with a sufficient number of neurons in the hidden layer with Support Vector

Regression (SVR), which is similar to the Support Vector Machine but with the introduction of a loss function, exploiting the benefits of both labelled and unlabelled data. The same approach based on co-training technique is proposed in [24], combining different classifiers, such as an "eager" classifier based on Decision Tree (DT) with a "lazy" classifier based on K-nearest neighbours (KNN). Both [23] and [24] conclude that accuracy improves with an increase in the number of iterations compared to self-training of supervised and unsupervised learning techniques.

In addition to the techniques for fault detection and predictive maintenance in the railway sector, reinforcement learning has also been explored in recent years to address relevant problems in railway planning and operations, without focusing on fault detection and predictive maintenance as in the other learning techniques. In [25], a reinforcement learning approach is applied to train scheduling, mainly for the characteristics of a timetable that can also be modelled as a discrete-time Markov Decision Process (MDP) to facilitate the learning process of an agent in responding appropriately to an incompletely known environment.

5.3 STDS-AF: Description and working principle

Alstom's new generation solution for managing audio frequency track circuits without the use of isolated mechanical joints is called STDS-AF, which stands for Smart Train Detection System Audio Frequency. Due to the availability of on field measurements in terms of currents and voltages, which are collected and sent to the Control Room through maintenance links based on LTE infrastructure, it has been decided to diagnose fault detection for the STDS-AF railway equipment. This is because the equipment will be installed on a large scale by the company in different lines. The STDS-AF has been designed to replace Alstom's existing track circuit families, implementing Eco Power features to limit power consumption in accordance with the following certifications:

- ✓ ISO 9001 of 2005: “Quality management systems Requirements”.
- ✓ ISO 14001 of 2005: “Environmental management systems – Requirements with guidance of use”.

The Eco Power principle involves dynamically reducing the amplitude of the generated signal based on the measure of the transmitted current and restoring its level when necessary. This allows the STDS-AF to adapt its power consumption to the real needs of the external environment, reducing energy consumption by up

to 30% compared to the previous system, without sacrificing performance and functionality. The new architecture of this railway equipment under revenue service operation introduces improvements such as:

- ✓ Immunity to new trains emission and adaptability to track environmental variations.
- ✓ High availability due to hot module redundancy.
- ✓ Long Track Circuit management (up to 2000 m).
- ✓ Extended range of Operating frequencies.
- ✓ High power efficiency.
- ✓ Multi-Track topology.
- ✓ Concentrated and distributed installation (Section & Joint Management).
- ✓ Replace the existing track circuit families such as DTC, STDS.

STDS-AF is an electronic system for railways or metro lines that generates and receives modulated signals to and from the track for the following purposes:

- ✓ perform train detection function, to determine the Track Circuit status (clear/occupied).
- ✓ perform broken rail detection function, to determine the rail integrity of the Track-circuit.
- ✓ perform track to train transmission optionally and if configured, to realize ATC function for train speed control.

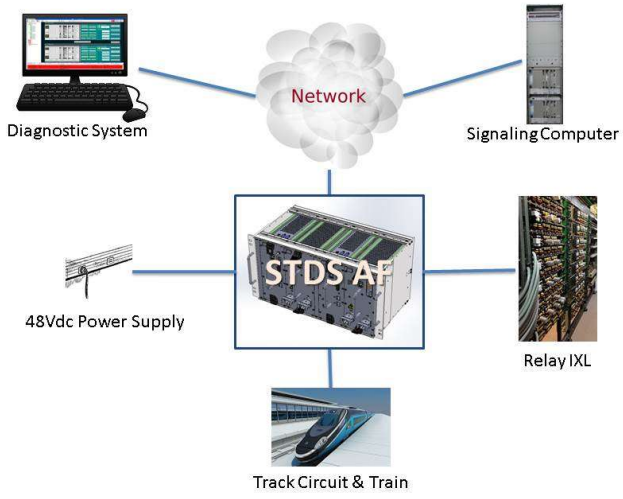


Figure 52 – STDS-AF environment

STDS-AF can receive a 48Vdc power supply without requiring an additional power supply unit. It interfaces with the SC and Diagnostic system via an Ethernet link and with the relay IXL through free voltage contacts (Vital Output). The STDS-AF detects the presence of a train in the TC by transmitting a known modulated signal to the track and verifying if it is recognized at the receiver side. If a train is present on the section, its axles shunt the rails, causing the received voltage to drop. This same technique can also be used to detect the integrity of the rails for broken rail detection.

The STDS-AF field interface comprises one transmitter and two receivers, each of which is independent and configurable to implement different topologies as required by the specific installation. The table below lists some typical configurations.

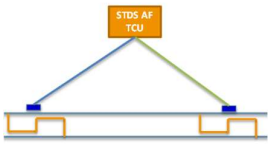
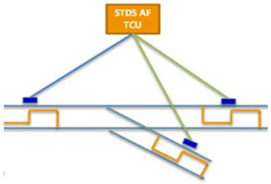
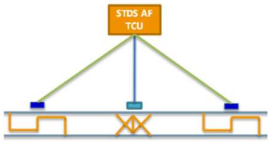
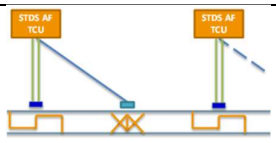
Track configuration	Type	Description	Layout
End Fed	Section Management	One STDS AF TCU located between two different electrical joints shall manage one track circuit.	
End Fed with one branch	Section Management	One STDS AF TCU shall manage simultaneously the Tx and the RX of 2 different track circuits.	
Centre Fed	Section Management	One STDS AF TCU (RX connected to two different electrical joints - TX on Mid-point joint) shall manage one track circuit with double length.	
Distributed End Fed	Joint Management	Two STDS AF TCU, each one located in front to the electrical joint, shall manage simultaneously the Tx and RX of 2 different track circuits located near the joint	
Distributed centre Fed	Joint Management	Two STDS AF TCU, located in front to the electrical joint, shall manage two different track circuits with double length.	

Table 5- STDS-AF Track circuit Layouts

The design of STDS-AF includes the ability to interface with various electrical joints, as below:

- ✓ "S" bond: Electric joint to separate two adjacent track circuits.
- ✓ Terminal bond: An electrical joint is used to define a track circuit in the presence of one or two mechanical interruptions of the rails. If there is only one mechanical interruption, any traction current can continue to the next track circuit by running through the full length of the cable of the joint. If there are two mechanical interruptions of the rails, the traction current, if present, can reach the next track circuit through a cable connected to the centre of the joint. From the next track circuit, the above-mentioned cable can be connected to one rail or to the centre point of an impedance bond.
- ✓ Short circuit bond: The joint is installed at the boundary between a track circuit and a zone where both rails must have the same electrical potential.
- ✓ Mid-Point bond: This is a component that creates an impedance at the centre of the track circuit, allowing the transmitted signal to be fed in both directions, to the left and right of the mid point.

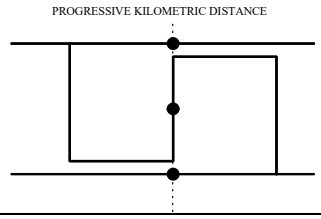
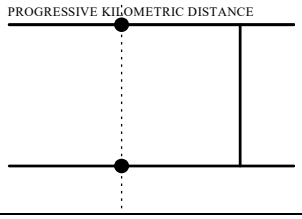
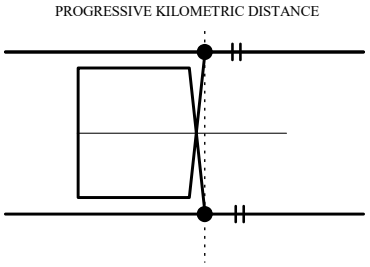
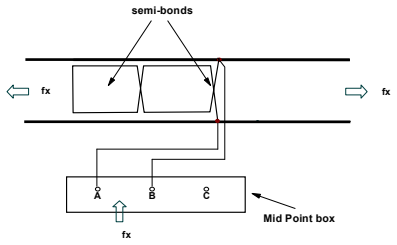
Connection scheme	
"S" bond 	**Short circuit bond** 
Terminal bond 	**Mid-Point bond** 

Table 6- STDS AF Electrical Joints type

The track circuit is a railway component that enables the identification of a train's position within a railway line. Mechanical joints may be used to isolate the track and separate it into different sections, with each section consisting of a transmitter and a receiver. The transmitter sends current through the rail circuit, which only reaches the receiver if there is no train in the section. However, if the wheels of a train create a short circuit, the receiver will no longer receive the current, indicating the presence of a train in that section.

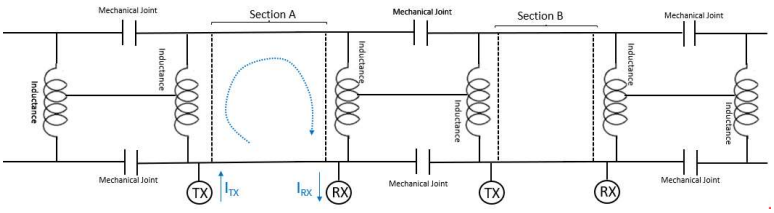


Figure 53 – Mechanical Joint

Mechanical joints represent an interruption of the rail, that is costly and require maintenance. To address these limitations, electrical joints have been developed. These joints enable the same separation and train identification function from an electrical perspective, without requiring physical interruptions of the rail.

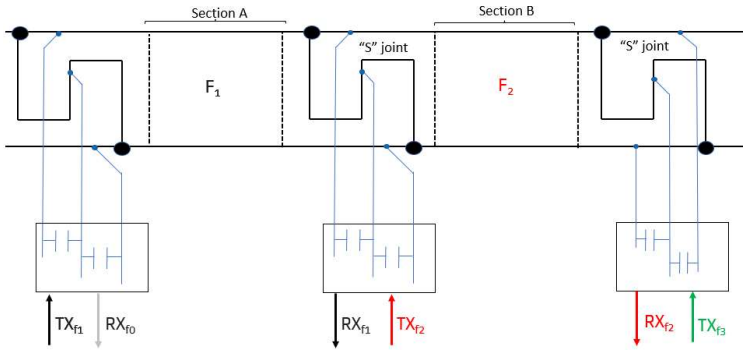


Figure 54 – "S" Electrical Joint

The resonance phenomenon is utilized to achieve electrical separation between two sections of the track circuit using electrical joints. The cable, which has an "S" shape and runs alongside the rail, acts as an inductance. By resonating this inductance with a specific capacitance value, a parallel resonant LC circuit is created in the area of interest. This LC circuit has a high impedance value for one frequency and a low impedance value for the next frequency. When a signal is injected into the track circuit by the transmitter with a specific frequency, the electrical joint ensures that a large portion of the electrical signal is conveyed only to one side, while a residual portion is sent to the opposite side, creating electrical separation between

the adjacent sections. The equivalent circuit of the electrical joint, which includes the "S" shaped cable and capacitors, is depicted in the picture below.

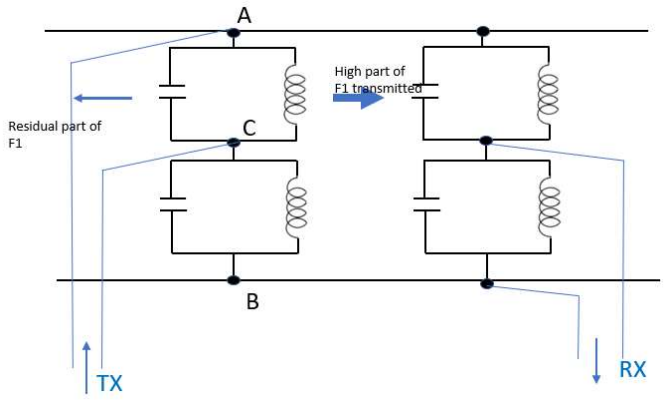


Figure 55 – "S" Electrical Joint - LC equivalent scheme

The above circuit is treated by the two rails as two parallel LC circuits, each resonant at a different frequency. If a specific frequency f1 is applied between the two rails, the parallel circuit will be characterized by a certain impedance at f1, while the circuit below will have zero impedance. As a result, the centre point C of the electrical joint is shifted to point B since the second circuit has zero impedance. When a voltage is transmitted at a certain frequency, the signal is propagated towards the right direction, with only a small amount transmitted to the opposite side. The electrical joint exhibits a symmetric behaviour at the receiver side, as the frequency propagated from the transmitter will encounter zero impedance on the upper resonant circuit and high impedance in the lower LC circuit, which is calibrated for the f1 frequency. By using different frequencies, it is possible to create track circuit sections for detecting the presence of a train. The track circuit acts as a transmission line between the transmitting and receiving joints, enabling the detection of a train or broken rail by measuring the reduction or absence of the transmitted signal.

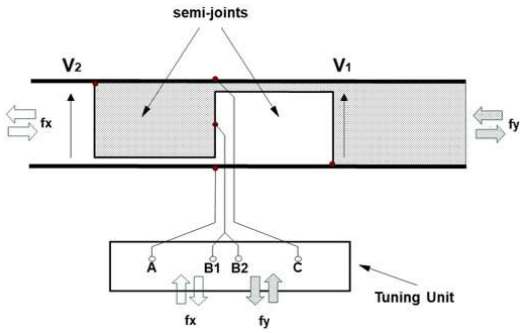


Figure 56 – "S" Electrical Joint working principle

The device functions by injecting a differential mode signal between the rails at one end of the track circuit and measuring it at the other end. Based on the received signal measurements, STDS-AF determines whether the track circuit is clear or occupied. To declare a track circuit as clear, the measured amplitude at the receiving end should be greater than the configured free threshold value and less than the configured maximum threshold value. The electrical joint used in track circuits managed by STDS-AF enables the delimitation of track circuit sections without interrupting the rails. In the event of a broken rail, the electric traction current return is ensured from the intact rail and the S cable for the imbalanced part.

Referring to the above picture, the electrical joint is linked to the tuning box via three conductors labelled "A", "B", and "C". The "B" conductor is connected to the centre of the electrical joint and is functionally associated with both track circuits. The signal for the left track circuit (relative to the electrical joint) operating at f_x frequency is either injected or extracted from the track via conductors "A" and "B", while the signal for the right track circuit (relative to the electrical joint) operating at f_y frequency is either injected or extracted from the track via conductors "B" and "C".

The electrical joint comprises two semi-joints with different lengths, each corresponding to one of the two track circuits. It is constructed using three cables that form an "S" shape and a fourth side that runs alongside the rail, creating a rectangular loop. The left track circuit, which operates at f_x frequency, uses the right semi-joint, while the right track circuit, which operates at f_y frequency, uses the left semi-joint.

5.4 STDS-AF: Railway equipment to be diagnosed

The STDS-AF railway equipment has been selected for diagnosis, as it will be widely used in the coming years for both new installations and to replace the previous devices used for train detection. Additionally, on field measurements of currents and voltages are available. Currently, this device is being installed on nonpriority portions of the railway line where only nonrevenue service wagons pass, to test its performance in a real environment and to gather data for further analysis. This thesis aims to analyse and develop a machine learning algorithm for maintenance activities to detect nominal and anomalous states, with the two STDS-AF track circuits installed in San Pellegrino station serving as the reference:

- ✓ Track circuit 304: It has a length of 762 meters, and it is equipped with compensation capacitors of the inductive rail component.
- ✓ Track circuit 153: it has a length of 81 meters, and it is installed without compensation capacitor.

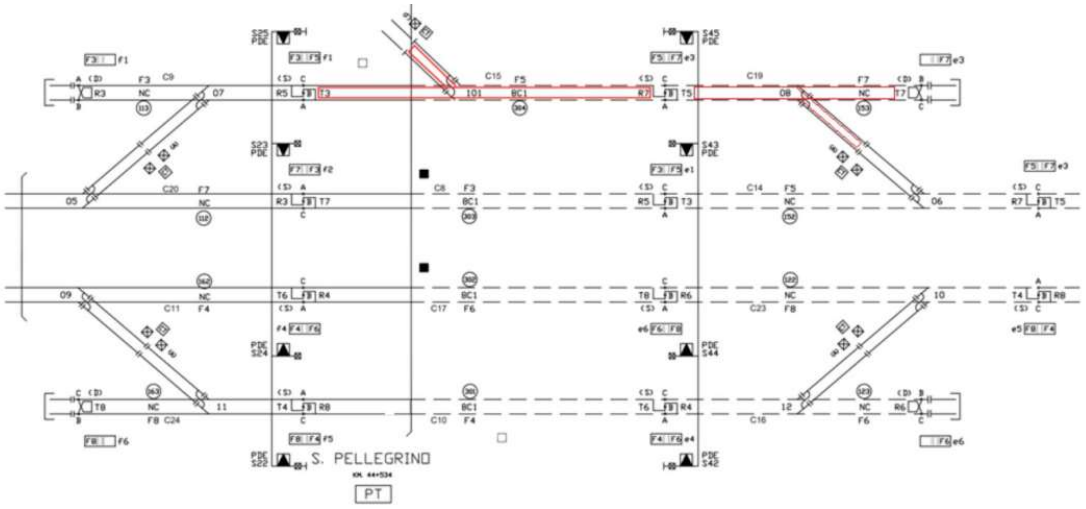


Figure 57 – San Pellegrino STDS-AF installation

To conserve space when transmitting data through the network, current and voltage measurements are sampled every 100 milliseconds. The figure below provides an example of the measurements taken from the STDS-AF component installed in the field.

Timestamp [+02:00]	VTX	ITX	VRX1	VRX2	Status	Sample	NRRelaySt atus	EcoPower Status	TxActive	Rx1Clear	Rx2Clear	Rx1OccCnt	Rx1ShOcc Cnt	Rx2OccCnt	Rx2ShOcc Cnt	ExtPS	IntPS	TempTXRX
2022-04-27 04:43:59.058	23	74	2,55	4,45	t	98454	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:43:59.568	23,25	74	2,55	4,425	t	98455	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:43:59.758	23	72	2,525	4,45	t	98456	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.008	23	74	2,55	4,45	t	98454	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.108	23,25	74	2,55	4,425	t	98455	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.208	23	72	2,525	4,45	t	98456	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.308	23	74	2,55	4,45	t	98457	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.408	23	72	2,525	4,425	t	98458	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.508	23	74	2,525	4,45	t	98459	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:00.608	23,25	74	2,55	4,45	t	98460	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:00.708	23	74	2,525	4,425	t	98461	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:00.808	23	74	2,55	4,45	t	98462	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:00.908	23	72	2,525	4,425	t	98463	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:01.008	23	74	2,525	4,45	t	98464	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:01.108	23	74	2,55	4,425	t	98465	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:01.208	23	74	2,525	4,45	t	98466	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:01.308	23	74	2,55	4,45	t	98467	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44
2022-04-27 04:44:01.408	23	72	2,525	4,425	t	98468	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:01.508	23	74	2,55	4,45	t	98469	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:01.608	23	74	2,55	4,425	t	98470	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,8	44
2022-04-27 04:44:01.708	23	74	2,525	4,425	t	98471	TRUE	FALSE	TRUE	TRUE	TRUE	50	30	66	24	49,2	47,7	44

Figure 58 – STDS-AF Current Voltage measurement

The below table describes the meaning of all the variables within the log.

Dataset column	Description
Timestamp	Measurement Timestamp with a sampling rate of 100 milliseconds only in case a value change has occurred.
VTX	Tx Voltage (RMS) RMS value of TX (range 0-255.75Vrms)

ITX	Tx Current (RMS) RMS value of ITX (range 0-2.046Arms)
VRX1	RX1 (Main Receiver) Voltage (RMS) RMS value of VRX of Main receiver (range 0-25.575Vrms)
VRX2	RX2 (Auxiliary Receiver) Voltage (RMS) RMS value of VRX of Auxiliary receiver (range 0-25.575Vrms)
Status	TBD t = Transition (the data are changed if compared with the previous cycle) < = Start of a missing data period > = End of a missing data period An empty cell means that data are not changed within the last 10 messages (1 second). In this case the values are the same of the previous CSV row
Sample	Incremental number of the sample
NRRelayStatus	N/R Relay status FALSE = N/R Relay coil de-energized TRUE = N/R Relay coil energized
EcoPowerStatus	EcoPower status FALSE = EcoPower functionality not active TRUE = EcoPower functionality active
TxActive	TX status FALSE = Transmitter not generating Track Circuit signal TRUE = Transmitter generating Track Circuit signal
Rx1Clear	RX1 (Main receiver) status FALSE = Occupied (also when TCU is available) TRUE = Clear
Rx2Clear	RX2 (Auxiliary receiver) status FALSE = Occupied (also when TCU is available) TRUE = Clear
Rx1OccCnt	Rx1 (Main Receiver) Occupation counter Number of occupations detected by Main receiver since TCU entered in NORMAL (in service) mode
Rx1ShOccCnt	RX1 (Main Receiver) Short Occupation counter Number of undue occupations (i.e. occupations in which the RMS value of received signal is out of range for less than 1 second) detected by Main receiver since TCU entered in NORMAL (in service) mode
Rx2OccCnt	RX2 (Auxiliary Receiver) Occupation counter Number of occupations detected by Auxiliary receiver since TCU entered in NORMAL (in service) mode.
Rx2ShOccCnt	RX2 (Auxiliary Receiver) Short Occupation counter

	Number of undue occupations (i.e., occupations in which the RMS value of received signal is out of range for less than 1 second) detected by Auxiliary receiver since TCU entered in NORMAL (in service) mode.
ExtPS	48V External Power Supply Measure of 48V External Power supply (Range 25V-75.8V)
IntPS	48V Internal Power Supply Measure of 48V Internal Power supply (Range 35V-60.4V) N/A means that the data is not available
TempTXRX	Temperature – TXRX Internal Tx/Rx temperature (Range -50 +204°C) N/A means that the data is not available

Table 7- Machine Learning Properties

Current and voltage measurements are collected from both the transmitter and receiver sides and stored in the ITX, VTX, VRX1, and VRX2 columns. Plotting these four variables enables the identification of occupancy and clearance events resulting from the passage of wagons.

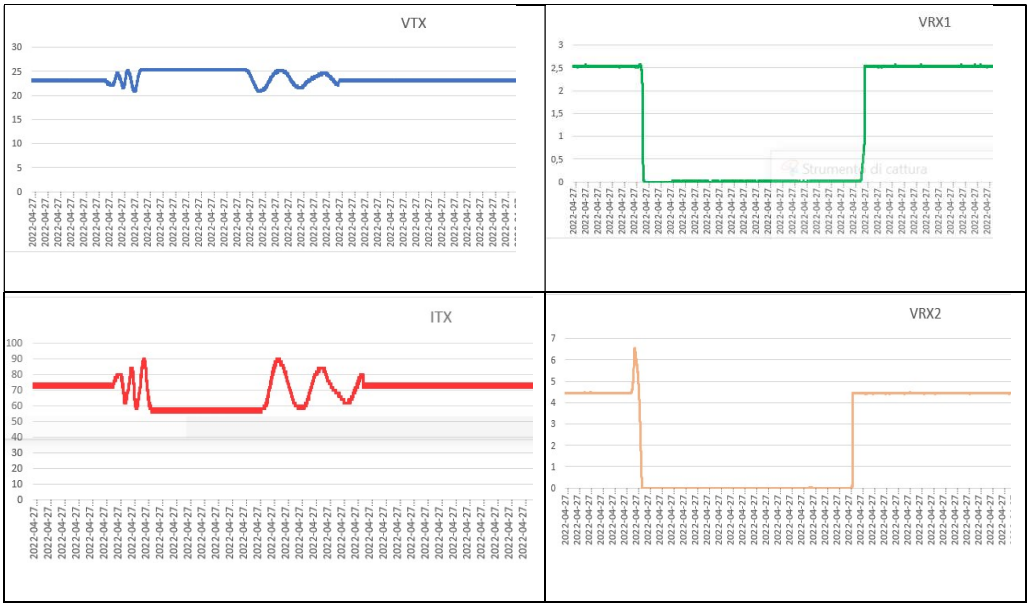


Figure 59 – STDS-AF Current Voltage measurement shape

Based on the above illustration, the sections of the graph where the receiving voltages are zero indicate the presence of occupied track circuits due to the closed circuit created by the train wheels and the rails. Conversely, sections where the voltage is non-zero represent free track circuits. Similar evaluations can be made based on the transmitted current values, where the portions between the fluctuations, caused by the capacitors installed on the track circuit, indicate occupancy events, and the steady line represents clearance.

5.5 Fault detection algorithm workflow

The aim of the current chapter is to describe the process used to develop a Python based machine learning algorithm for detecting the normal operation behaviour and equipment failures of an STDS-AF track circuit.

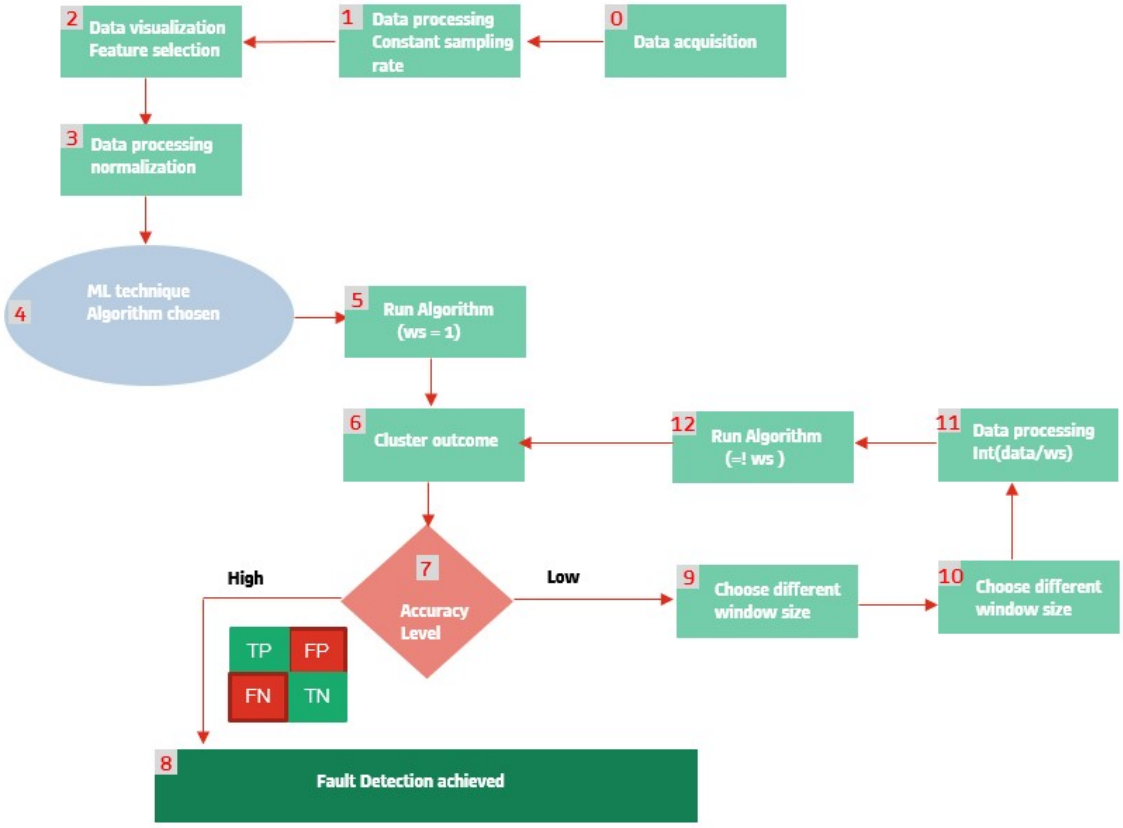


Figure 60 – Fault Detection Algorithm workflow

The first step in developing the algorithm involves acquiring the data obtained from on field measurements. The current and voltage measurements collected from the field are recorded only when there is a change in measurement, every 100 milliseconds, to conserve memory during data transmission. As a result, the input

dataset does not have a constant sampling rate. To address this issue, a Python loop attached in the appendix was utilized to pre-process the data by copying the previous row with the same measurement values whenever the difference between two consecutive records exceeded the 100-millisecond sampling rate. This approach was adopted to create a dataset with a constant sampling rate. In this thesis, the sampling rate is also referred to as the window used to observe the behaviour.

	Timestamp	VTX	ITX	VRX1	VRX2		Timestamp	VTX	ITX	VRX1	VRX2
0	2022-04-27 04:43:59.058	23.00	74	2.550	4.450	0	2022-04-27 04:43:59.058	23.0	74	2.55	4.45
1	2022-04-27 04:43:59.568	23.25	74	2.550	4.425	1	2022-04-27 04:43:59.158	23.0	74	2.55	4.45
2	2022-04-27 04:43:59.758	23.00	72	2.525	4.450	2	2022-04-27 04:43:59.258	23.0	74	2.55	4.45
3	2022-04-27 04:44:00.008	23.00	74	2.550	4.450	3	2022-04-27 04:43:59.358	23.0	74	2.55	4.45
4	2022-04-27 04:44:00.108	23.25	74	2.550	4.425	4	2022-04-27 04:43:59.458	23.0	74	2.55	4.45

Figure 61 – Data pre-processed for constant sampling rate

In the next step, the data was visualized, and the transmitted voltage and received voltage were selected as the features of interest for algorithm development. Since these features have varying values, a normalization process was utilized to ensure that they were of the same order of magnitude, without compromising the resolution and shape of the signals.

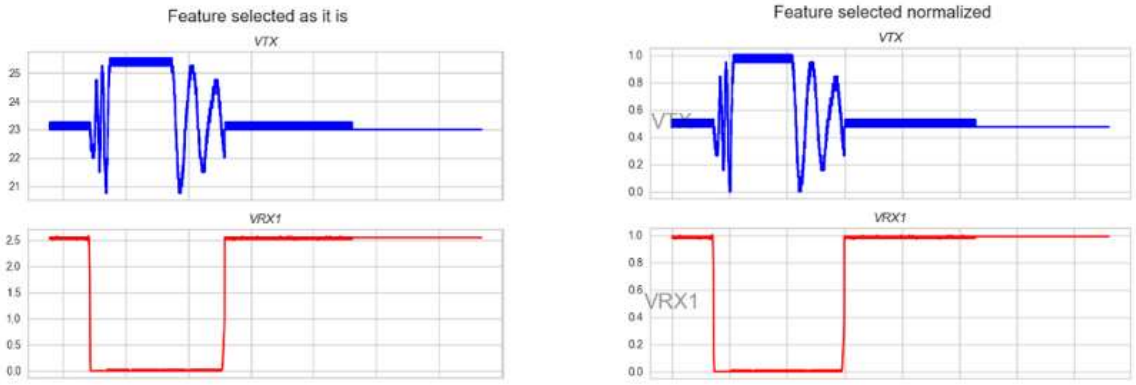


Figure 62 – Features normalized

An unsupervised technique, such as K-means, was used to develop the model due to the availability of structured but unlabelled data. The K-means algorithm, which aims to minimize the Euclidean distance between each point and the centroids, was executed by considering the measurements individually to group those with similar characteristics. After obtaining clusters from the K-means algorithm, the accuracy of the model was evaluated using a confusion matrix and calculating the true positive, true negative, false positive, and false negative values.

If the accuracy of the model is not exhaustive, a recursive process is initiated to improve it. Rather than considering individual points, observations are grouped in pairs or more to work with arrays and multidimensional arrays to achieve optimal accuracy. To automate this research, a further data pre-processing phase was implemented to determine whether the length of the input dataset was divisible by the size of the chosen array. If not, rows were appended to the end of the file to obtain an integer division between the length of the input data and the array size. Once the new observation window size was determined, the algorithm was restarted to obtain new clusters and evaluate the model until an exhaustive performance was achieved, compared to a window size compatible with occupancy events.

As previously mentioned, the STDS-AF equipment is not installed on the main track. Consequently, the data collected only contains a limited number of occupancy and clearance events resulting from switching activities and the passage of wagons. To develop an algorithm based on machine learning techniques capable of detecting equipment failures, impairment events were synthetically introduced into the input data, to achieve the objective of the diagnosis. To evaluate the model and achieve the intended objective, the following two failures were artificially generated within the available input data:

- ✓ No Power: Both features VTX and VRX1 selected imposed null.
- ✓ Peak Voltage: The selected features, VTX and VRX1, are characterized by a sudden increase in values.

The algorithm should be capable of clustering the events as shown in the cartesian graph below. This should be done by observing the two selected features, transmitted voltage and received voltage, in the input data, which contains occupancy, clearance, and synthetically injected failures.

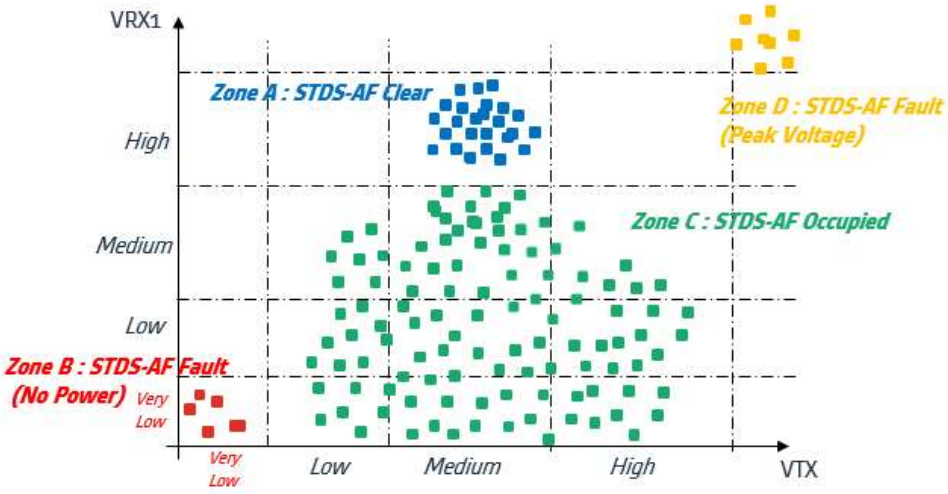


Figure 63 – Clusters expectation

Ideally, the K-means algorithm should subdivide the data into distinct zones based on their similarity, resulting in the following clusters:

- ✓ Zone A: This is represented by the top central part of the above image, indicating a clearance event.
- ✓ Zone B: This is represented by the lower left-hand side of the graph above, where both selected features are characterized by low values, indicating the absence of a power failure.
- ✓ Zone C: This is represented by the bottom central part of the image above, indicating an occupancy event.
- ✓ Zone D: This is represented by the upper right-hand side of the above graph, where both selected features are characterized by high values, indicating a peak voltage.

5.6 Fault detection algorithm results

The algorithm was evaluated using three different datasets with an unsupervised machine learning approach, given the structured and unlabelled input data, to solve a clustering problem in detecting the nominal and anomalous states of the STDS-AF railway equipment to be diagnosed.

The model evaluation was conducted progressively, starting with a dataset containing only occupancy and clearance events to confirm whether the algorithm could distinguish between the two states characterizing the nominal behaviour of the railway equipment.

Subsequently, data containing failures were introduced to the algorithm to improve its ability to differentiate between the nominal and two different anomalous events, which were synthetically injected to achieve the final objective of accurately identifying all states for fault detection purposes.

Different window sizes were used for each of the three datasets to observe the model's behaviour and improve its precision and related accuracy. Based on the above considerations, different use cases and related results utilizing all three input datasets will be discussed in the following chapters of this thesis, as listed in the table below.

State Detection: for data containing occupancy and clearance events	**One Fault Detection**: for data containing occupancy, clearance, and no-power events	**Two Faults Detection**: for data containing occupancy, clearance, no-power, and no peak voltage events

UC1	window size = 1	UC1	window size = 1	UC1	window size = 1
UC2	window size = 2	UC2	window size = 2	UC2	window size = 2
UC3	window size = 8	UC3	window size = 8	UC3	window size = 8
UC4	window size = 600	UC4	window size = 600	UC4	window size = 600

Table 8- List of the algorithm test

5.6.1 State Detection Use Case 1

For this particular use case, a dataset comprising only occupancy and clearance events was utilized. The K-means algorithm was applied to this dataset with a sample rate of 100 milliseconds that means a window size of 1, to estimate the events,

The two selected features, VTX and VRX1, fitted in the available input data, were visualized to confirm the presence of two distinct states. They were then normalized to ensure that they were of the same order of magnitude. Successively data has been processed through the Elbow method to identify the optimal value of k to assign to the K-means algorithm.

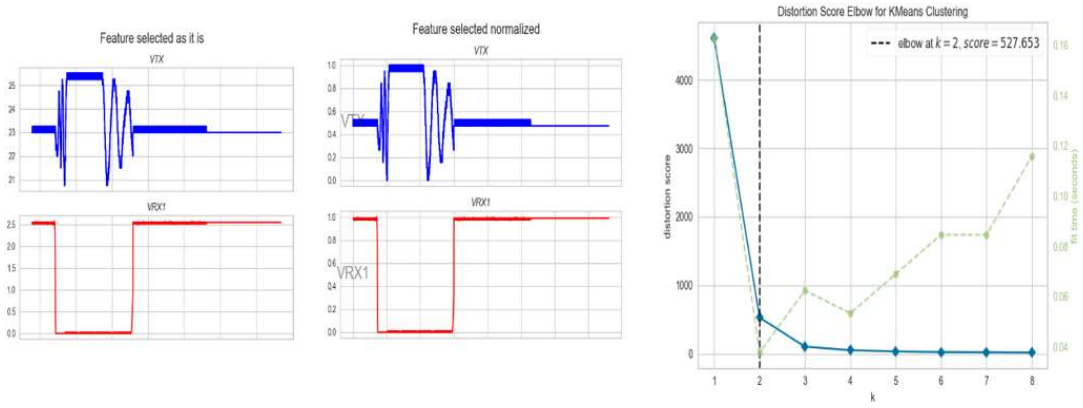


Figure 64 – Optimal k for UC1 state detection

The optimal value for the algorithm to converge was found to be k=2, indicating that the K-means algorithm should produce two clusters as its outcome.

The cluster separation is illustrated in two dimensions since the values are treated as individual data points for the two selected features. This confirms that the points are divided into two distinct groups based on their similarity, as shown below:

- ✓ Cluster 0: represented in blue and supposed to group occupancy events.
- ✓ Cluster 1: represented in orange and supposed to group clearance events.

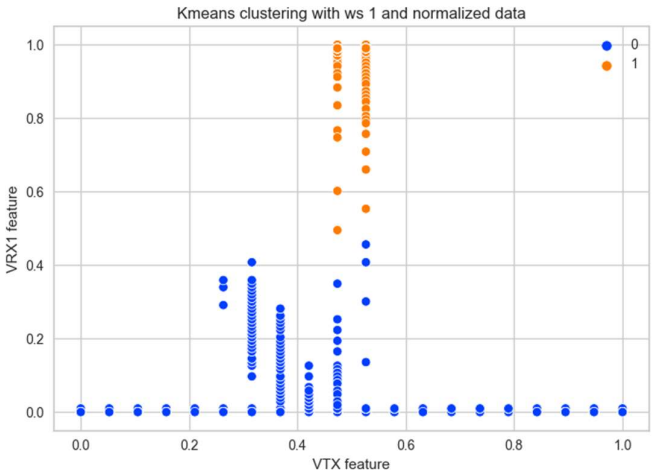


Figure 65 – Cluster representation UC1 state detection

The clustering outcome appears to group the two cases accurately based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcomes:

- ✓ 13283 events marked as "0" on input have been predicted as "1".
- ✓ 5687 events marked as "1" on input have been predicted as "0".
- ✓ 34 events marked as "1" on input have been predicted as "1".
- ✓ No events marked as "0" on input have been predicted as "0".

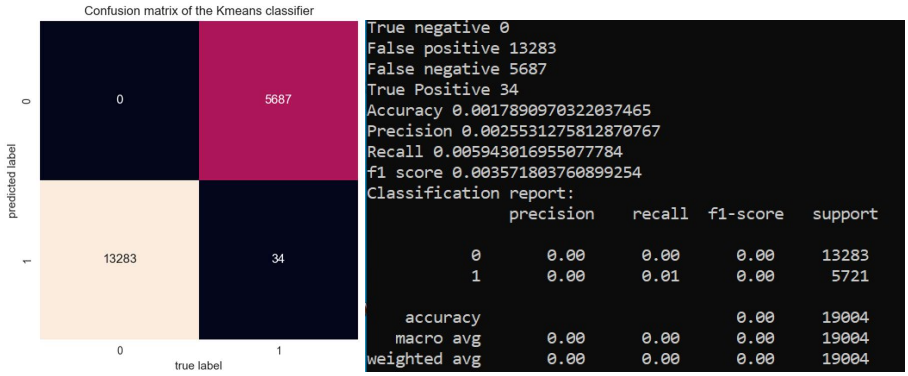


Figure 66 – Accuracy UC1 state detection

In conclusion, using a sampling rate of 100 milliseconds and evaluating the results with a confusion matrix, there was a complete misclassification in identifying the correct state, resulting in an accuracy of 0%.

No faults were detected by the algorithm as they were not injected in the dataset used, and instead, clearance events were identified as occupancy, and occupancy events were identified as clearance. This misclassification is illustrated in the image below, which shows the two selected features and the incorrect identification of the two states.

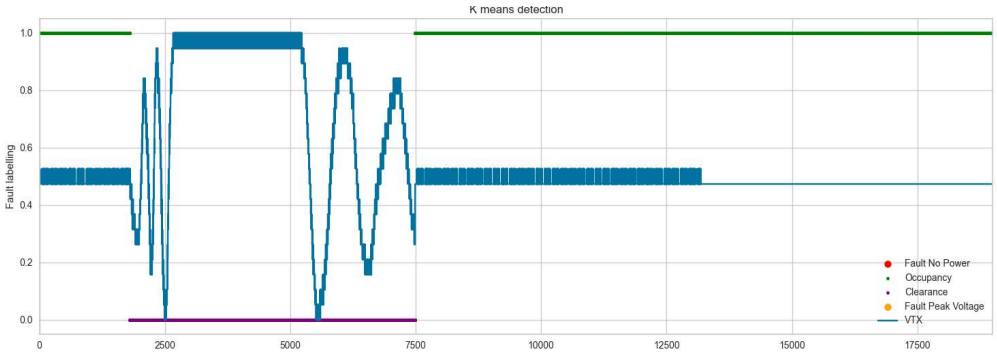


Figure 67 – Graphic state detection for UC1

5.6.2 State Detection Use Case 2

For this particular use case, a dataset consisting only of occupancy and clearance events was used. The K-means algorithm was applied to this dataset using a sample rate of 200 milliseconds that means a window size of 2, to estimate the events. The two features selected, VTX and VRX1, present in the available input data, were visualized to confirm the presence of two distinct states. They were then normalized to ensure that they were of the same order of magnitude. The Elbow method was used to determine the optimal value of k to be assigned to the K-means algorithm. This value was not affected by the new window size chosen.

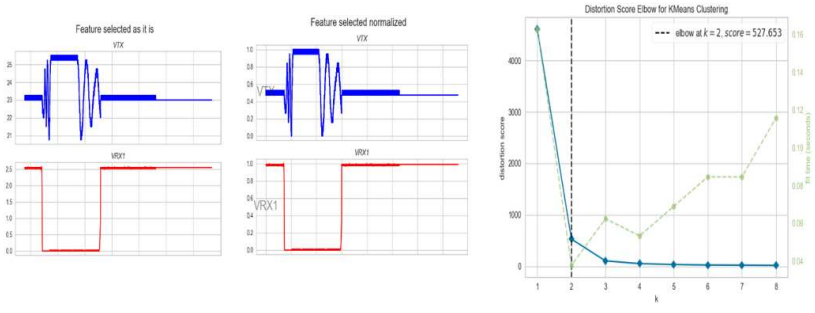


Figure 68 – Optimal k for UC2 state detection

The optimal value for k at which the algorithm converged remained the same at 2. As a result, assigning this value to the K-means algorithm, we can expect two clusters as the outcome.

The cluster separation is illustrated in three dimensions since the values are treated as 2x2 matrices for the two selected features. This confirms that the points are divided into two distinct groups based on their similarity, as shown below:

- ✓ Cluster 0: supposed to group occupancy events.
- ✓ Cluster 1: supposed to group clearance events.

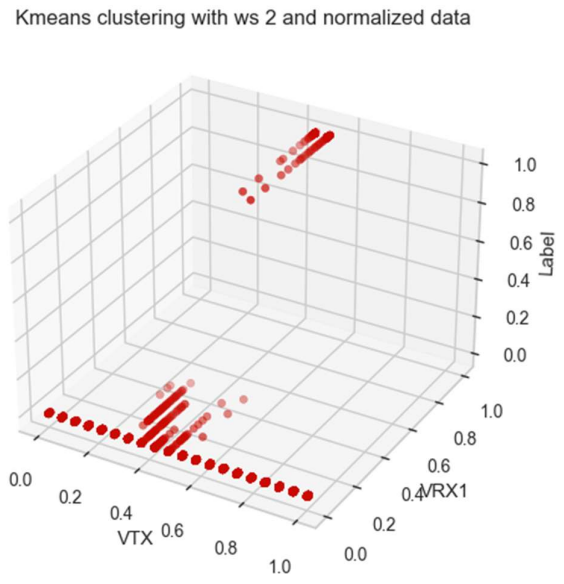


Figure 69 – Cluster representation UC2 state detection

The clustering outcome appears to group the two cases accurately based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcome:

- ✓ 6643 events marked as "0" on input have been predicted as "1".
- ✓ 2843 events marked as "1" on input have been predicted as "0".
- ✓ 17 events marked as "1" on input have been predicted as "1".
- ✓ No events marked as "0" on input have been predicted as "0".

Figure 70 – Accuracy UC2 state detection

In conclusion, even with an increased sampling rate of 200 milliseconds, there was no improvement in accuracy. Evaluating the results with a confusion matrix showed a complete misclassification in identifying the correct state, resulting in an accuracy of 0%.

No faults were detected by the algorithm as they were not present in the dataset used, and instead, clearance events were identified as occupancy, and occupancy events were identified as clearance. This misclassification is illustrated in the image below, which shows the two selected features and the incorrect identification of the two states.

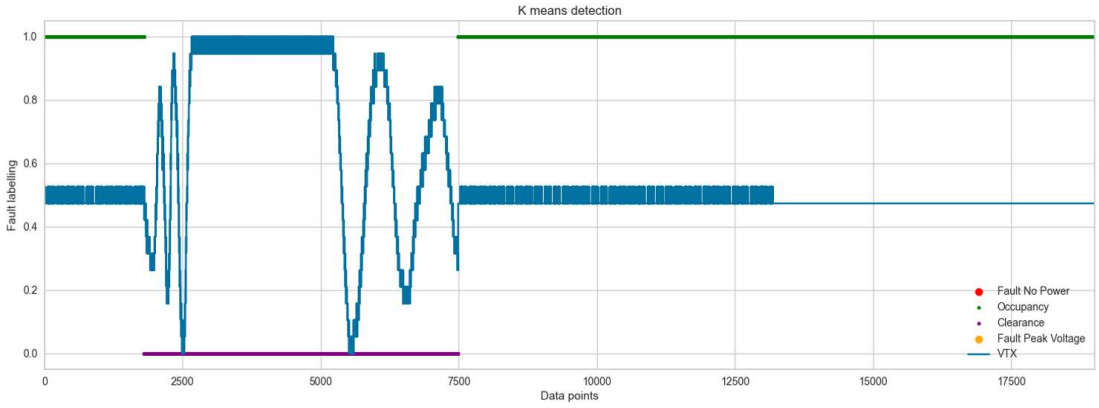


Figure 71 – Graphic state detection for UC2

5.6.3 State Detection Use Case 3

For this particular use case, a dataset consisting only of occupancy and clearance events was used. The K-means algorithm was applied to this dataset using a sample rate of 800 milliseconds that means a window size of 8, to estimate the events.

The two selected features, VTX and VRX1, present in the available input data, were visualized to confirm the presence of two distinct states. They were then normalized to ensure that they were of the same order of magnitude. The Elbow method was used to determine the optimal value of k to be assigned to the K-means algorithm, which was not affected by the new window size chosen.

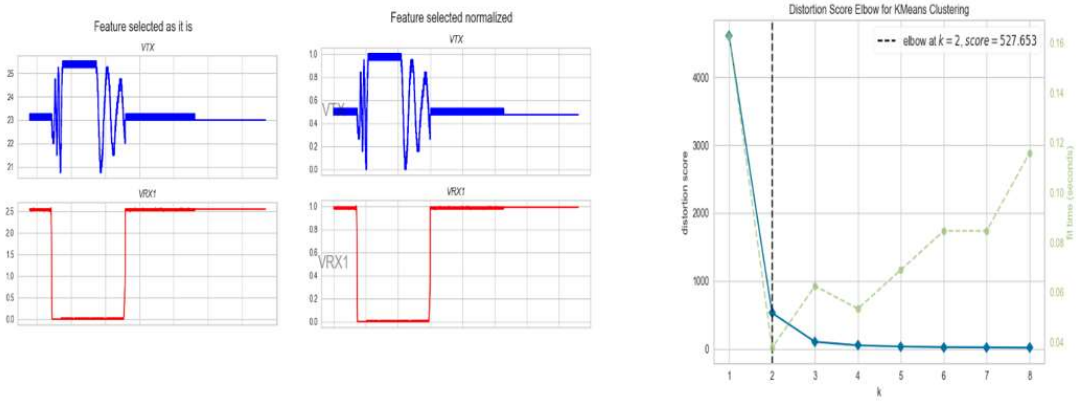


Figure 72 – Optimal k for UC3 state detection

The optimal value for k at which the algorithm converged remained the same at 2. As a result, assigning this value to the K-means algorithm, we can expect two clusters as the outcome. In this case, the values were treated as 8x8 matrices for the two selected features. Thus, working with a multidimensional array, the graphical representation of clusters cannot be represented in either 2-dimensional or 3-dimensional format.

For the purpose of cluster representation, dimensionality reduction techniques such as principal component analysis and t-distributed stochastic neighbour embedding were implemented in the Python code provided in the appendix. These techniques were used despite losing some information and resolution.

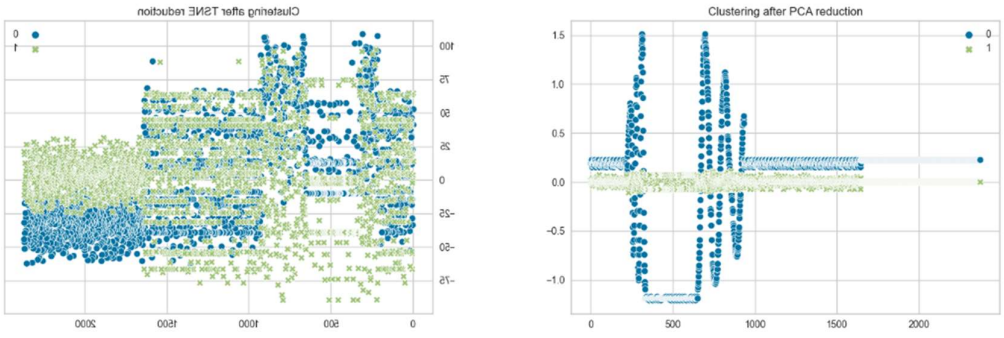
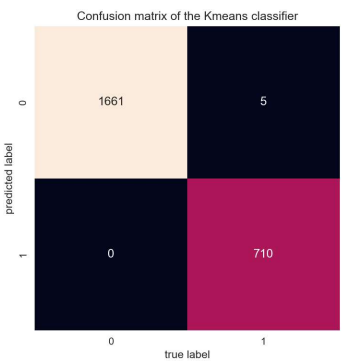


Figure 73 – Cluster representation UC3 state detection

The cluster representation obtained through dimensionality reduction techniques does not provide a clear indication of whether the model has created the clusters correctly. To evaluate the model's performance, true positive, true negative, false positive, and false negative metrics were computed from the confusion matrix, resulting in the following outcome:

- ✓ 1661 events marked as "0" on input have been predicted as "0".
- ✓ 710 events marked as "1" on input have been predicted as "1".
- ✓ 5 events marked as "1" on input have been predicted as "0".
- ✓ No events marked as "0" on input have been predicted as "1".




```
True negative 1661
False positive 0
False negative 5
True Positive 710
Accuracy 0.997895622895623
Precision 1.0
Recall 0.993006993006993
f1 score 0.9964912280701755
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1661
           1       1.00      0.99      1.00       715

    accuracy                           1.00      2376
   macro avg       1.00      1.00      1.00      2376
weighted avg       1.00      1.00      1.00      2376
```

Figure 74 – Accuracy UC3 state detection

In conclusion, increasing the sampling rate to 800 milliseconds resulted in a significant improvement in accuracy, with only a few events being misinterpreted. This led to a high level of precision in detecting the two states.

The algorithm did not detect any faults since not injected on the input data, and occupancy events were identified as occupancy, while clearance events were identified as clearance. This is illustrated in the image below, which shows the two selected features and the correct identification of the two states.

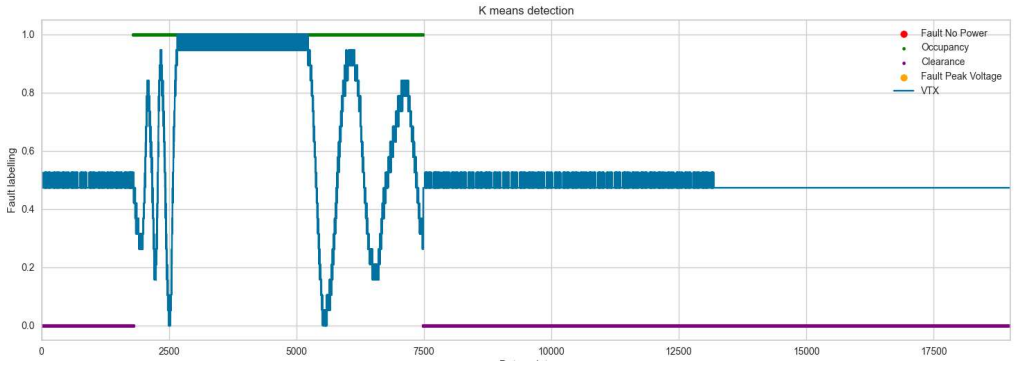


Figure 75 – Graphic state detection for UC3

5.6.4 State Detection Use Case 4

For this particular use case, a dataset consisting only of occupancy and clearance events was used. The K-means algorithm was applied to this dataset using a sample rate of 1 minute to estimate the events, resulting in a window size of 600.

The two selected features, VTX and VRX1, present in the available input data, were visualized to confirm the presence of two distinct states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the data was processed using the Elbow method to identify the optimal value of k to assign to the K-means algorithm. This value was not impacted by the new window size chosen.

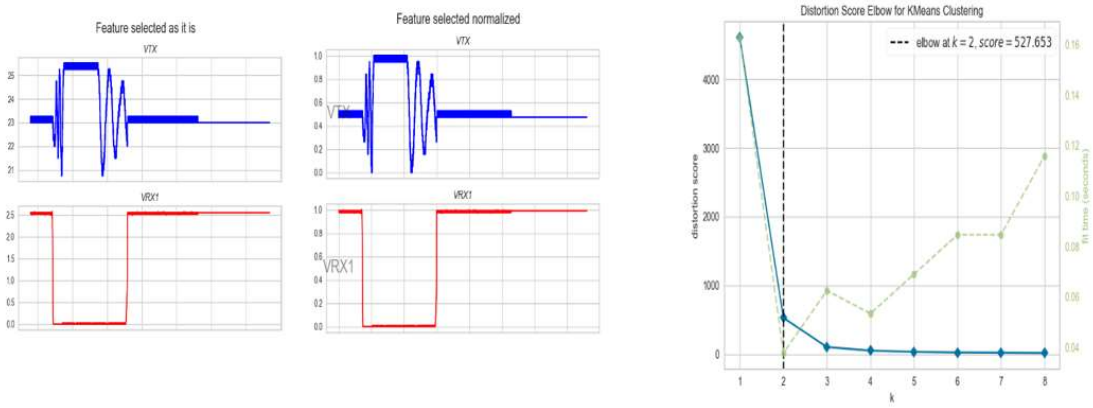


Figure 76 – Optimal k for UC4 state detection

The optimal value for k at which the algorithm converged remained the same at 2. As a result, assigning this value to the K-means algorithm, we can expect two clusters as the outcome.

In this case, the values were treated as 600x600 matrices for the two selected features. Thus, working with a multidimensional array, the graphical representation of clusters cannot be represented in either 2-dimensional or 3-dimensional format.

Dimensionality reduction techniques such as principal component analysis and t-distributed stochastic neighbour embedding were implemented in the Python code provided in the appendix for the purpose of cluster representation, despite the loss of information and resolution.

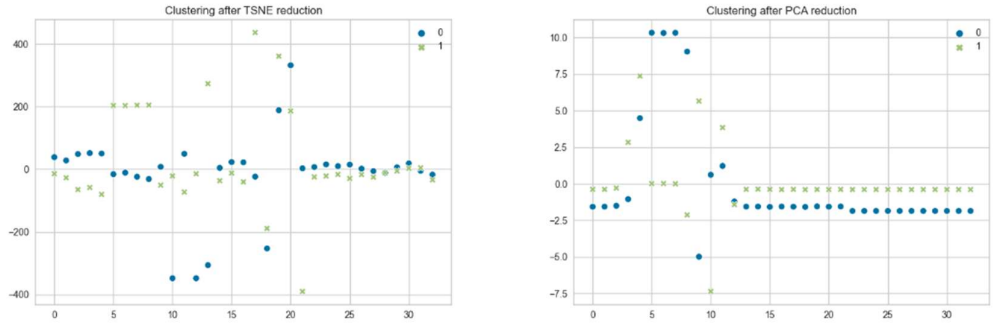
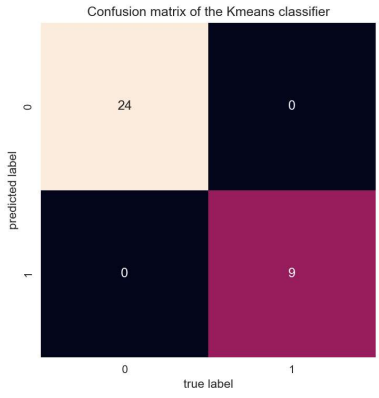


Figure 77 – Cluster representation UC4 state detection

With an increase in the sampling rate to 1 minute, a clear separation of the clusters was obtained because of the dimensionality reduction applied for the sake of representation.

To evaluate the model's performance, true positive, true negative, false positive, and false negative metrics were computed from the confusion matrix, resulting in the following outcome:

- ✓ 24 events marked as "0" on input have been predicted as "0".
- ✓ 9 events marked as "1" on input have been predicted as "1".
- ✓ No events marked as "1" on input have been predicted as "0".
- ✓ No events marked as "0" on input have been predicted as "1".




```
True negative 24
False positive 0
False negative 0
True Positive 9
Accuracy 1.0
Precision 1.0
Recall 1.0
f1 score 1.0
Classification report:
              precision    recall  f1-score   suppor

           0       1.00      1.00      1.00        24
           1       1.00      1.00      1.00         9

    accuracy                           1.00        33
   macro avg       1.00      1.00      1.00        33
weighted avg       1.00      1.00      1.00        33
```

Figure 78 – Accuracy UC4 state detection

In conclusion, increasing the sampling rate to 1 minute resulted in optimal accuracy, with the removal of misinterpreted events leading to a high level of precision in detecting the two states.

The algorithm did not detect any faults since not injected on the input data, and occupancy events were identified as occupancy, while clearance events were identified as clearance. This is illustrated in the image below, which shows the two selected features and the correct identification of the two states.

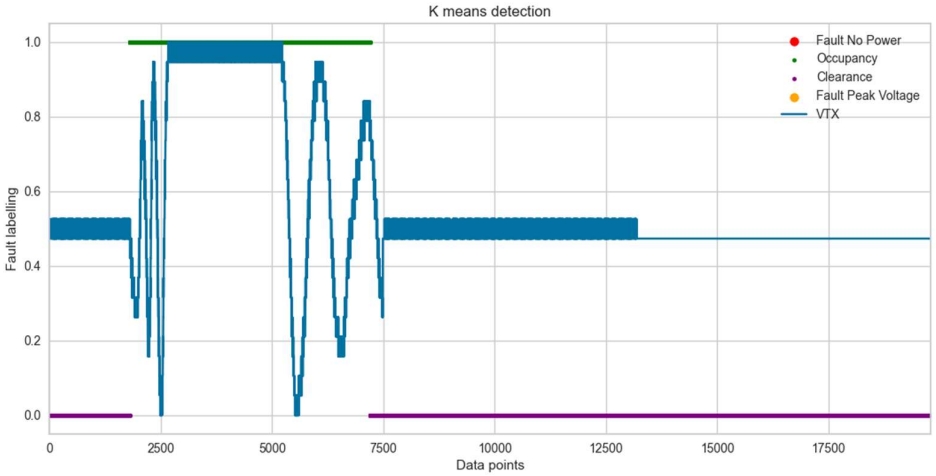


Figure 79 – Graphic state detection for UC4

5.6.5 One Fault Detection Use Case 1

For this particular use case, a dataset consisting of occupancy and clearance events with the injection of a no power failure in a synthesized manner was used. The K-means algorithm was applied to this dataset using a sample rate of 100 milliseconds meaning a window size of 1, to estimate the events.

The available input data was visualized using the two selected features, VTX and VRX1, which confirmed the presence of three distinct states. The data was then normalized to ensure that they were of the same order of magnitude.

Subsequently, the Elbow method was applied to identify the optimal value of k to assign to the K-means algorithm.

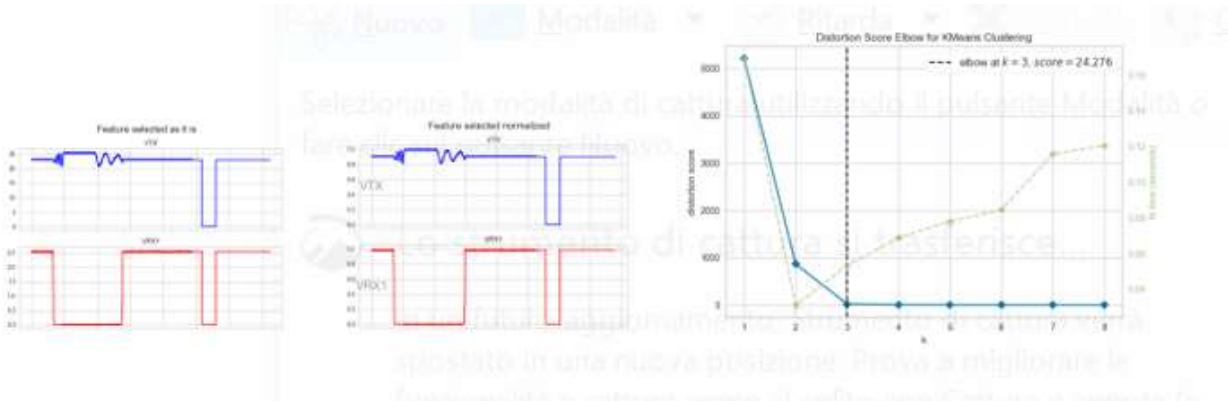


Figure 80 – Optimal k for UC1 one fault detection

The optimal value at which the algorithm converged was k equal to 3. Assigning this value to the K-means algorithm, we can expect three clusters as the outcome.

The cluster separation is depicted in 2-dimensional format as the values are treated as singular points of the two selected features. This confirms that the points are separated into three distinct ensembles based on their similarity, as shown below:

- ✓ Cluster 0: represented in blue and supposed to group occupancy events.
- ✓ Cluster 1: represented in orange and supposed to group clearance events.
- ✓ Cluster 2: represented in green and supposed to group fault events.

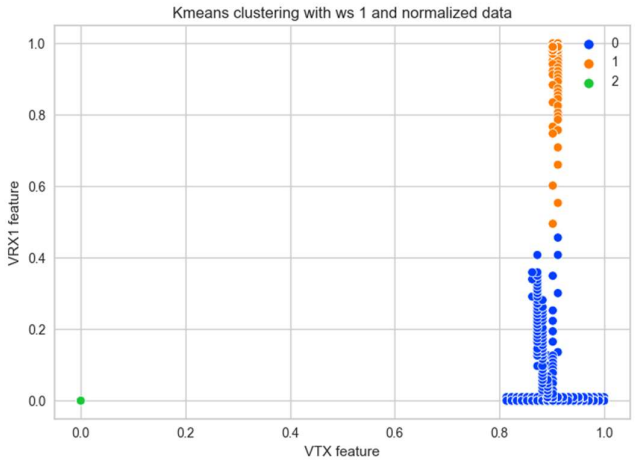
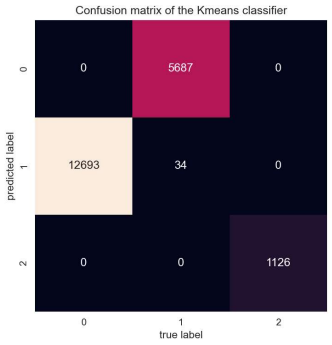


Figure 81 – Cluster representation UC1 one fault detection

The clustering outcome appears to group the three cases properly based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcome:

- ✓ No events marked as "0" on input have been predicted as "2".
- ✓ 12693 events marked as "0" on input have been predicted as "1".
- ✓ No events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "2".
- ✓ 34 events marked as "1" on input have been predicted as "1".
- ✓ 5687 events marked as "1" on input have been predicted as "0".
- ✓ 1126 events marked as "2" on input have been predicted as "2".
- ✓ No events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".




```
Accuracy 0.0593654042988741
Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     12693
           1       0.00      0.01      0.00      5721
           2       1.00      1.00      1.00      1126

    accuracy                           0.06     19540
   macro avg       0.33      0.34      0.33     19540
weighted avg       0.06      0.06      0.06     19540
```

Figure 82 – Accuracy UC1 one fault detection

In conclusion, with a sampling rate of 100 milliseconds, evaluating the model's performance with a confusion matrix revealed a complete misinterpretation in detecting the correct state between occupancy and clearance events, while the fault was identified correctly. This resulted in an accuracy of 6%.

The algorithm detected the fault declared as no power with 100% precision, while occupancy events were identified as clearance and clearance events as occupancy. This is illustrated in the image below, which shows the two selected features and the correct identification of the three different states.

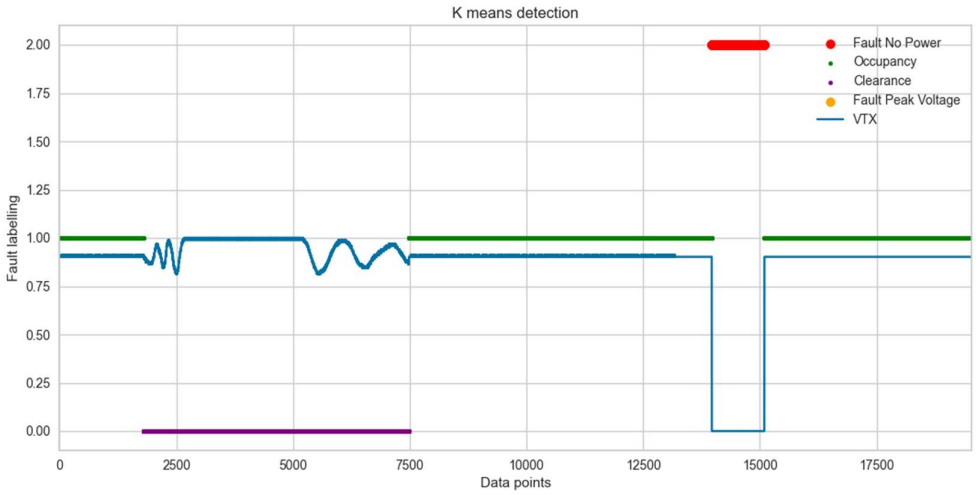


Figure 83 – Graphic one fault detection for UC1

5.6.6 One Fault Detection Use Case 2

For this use case, a dataset consisting of occupancy and clearance events with the injection of a synthesized no power failure was used. The K-means algorithm was applied to this dataset using a sample rate of 200 milliseconds that corresponds to a window size of 2, to estimate the events.

The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of three states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was used to identify the optimal value of k to assign to the K-means

algorithm.

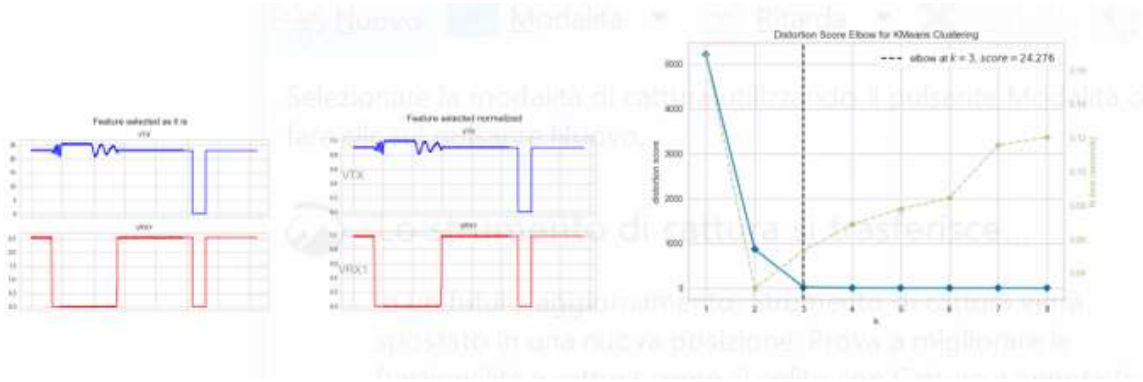


Figure 84 – Optimal k for UC2 one fault detection

The optimal value for k at which the algorithm converged remained the same at 3. As a result, assigning this value to the K-means algorithm, we can expect three clusters as the outcome.

The cluster separation is depicted in 3-dimensional format as the values are treated as 2x2 matrices for the two selected features. This confirms that the points are separated into three distinct ensembles based on their similarity, as shown below:

- ✓ Cluster 0: supposed to group occupancy events.
- ✓ Cluster 1: supposed to group clearance events.
- ✓ Cluster 2: supposed to group no power faults events.

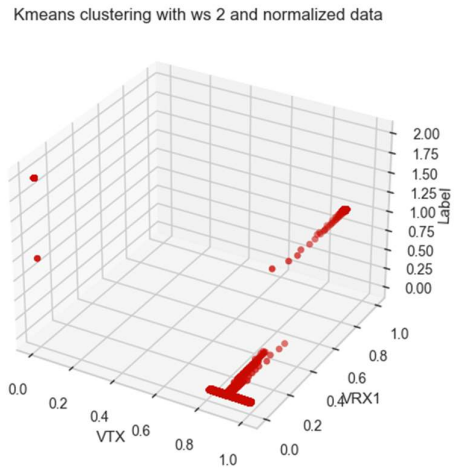
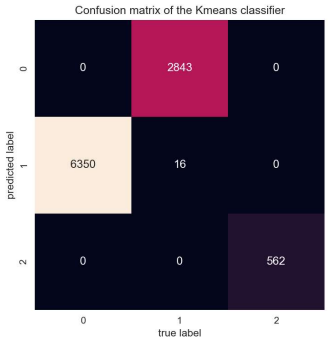


Figure 85 – Cluster representation UC2 one fault detection

The clustering outcome appears to group the three cases properly based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcome:

- ✓ No events marked as "0" on input have been predicted as "2".
- ✓ 6350 events marked as "0" on input have been predicted as "1".
- ✓ No events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "2".
- ✓ 16 events marked as "1" on input have been predicted as "1".
- ✓ 2843 events marked as "1" on input have been predicted as "0".
- ✓ 562 events marked as "2" on input have been predicted as "2".
- ✓ No events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".




```
Accuracy 0.059154641285436495
Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      6350
           1       0.00      0.01      0.00      2859
           2       1.00      1.00      1.00       562

    accuracy                           0.06      9771
   macro avg       0.33      0.34      0.33      9771
weighted avg       0.06      0.06      0.06      9771
```

Figure 86 – Accuracy UC2 one fault detection

In conclusion, even with an increased sampling rate of 200 milliseconds, there was no improvement in accuracy. Evaluating the performance of the model with a confusion matrix revealed a complete misinterpretation in detecting the correct state between occupancy and clearance events, while the no power fault was correctly clustered. This resulted in an accuracy of 6%.

The algorithm detected the fault declared as no power with 100% precision, while occupancy events were identified as clearance and clearance events as occupancy. This is illustrated in the image below, which shows the two selected features and the correct identification of the three different states.

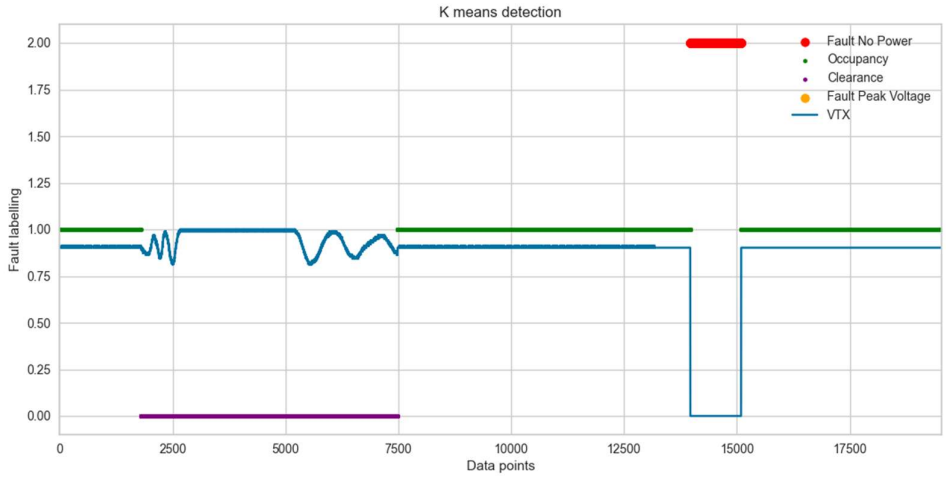


Figure 87 – Graphic one fault detection for UC2

5.6.7 One Fault Detection Use Case 3

For this use case, a dataset consisting of occupancy and clearance events with the injection of a synthesized no power failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 800 milliseconds that corresponds to a window size of 8, to estimate the events.

The available input data was visualized using the two selected features, VTX and VRX1, which confirmed the presence of three distinct states. The data was then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was applied to identify the optimal value of k to assign to the K-means algorithm.

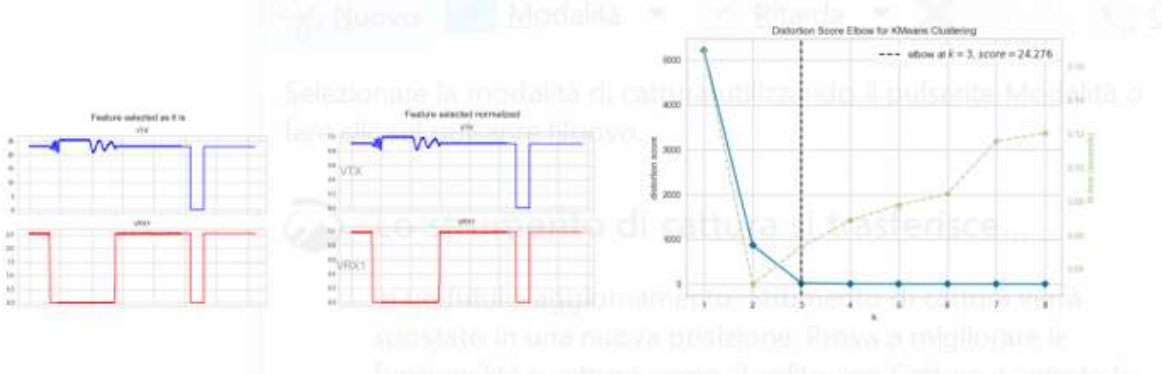


Figure 88 – Optimal k for UC3 one fault detection

The algorithm continued to converge at an optimal k value of 3. As a result, assigning this value to the K-means algorithm would result in three clusters as the outcome.

In this case, the values are treated as 8x8 matrices for the two selected features. As a result, when working with multidimensional arrays, the graphical cluster representation cannot be depicted in either 2-dimensional or 3-dimensional format. For the purpose of cluster representation, the Python code provided in the appendix implemented dimensionality reduction techniques such as principal component analysis and t-distributed stochastic neighbour embedding. However, this resulted in a loss of information and resolution.

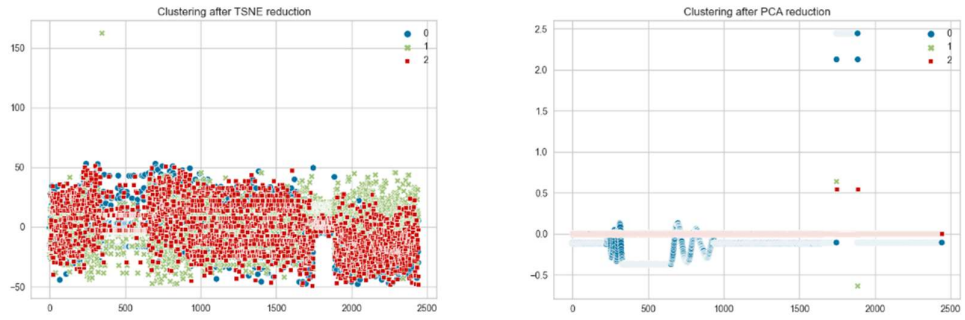


Figure 89 – Cluster representation UC3 one fault detection

The use of dimensionality reduction techniques for cluster representation did not provide a clear indication of whether the model had created the clusters properly. Therefore, the model's performance was evaluated by computing the true positive, true negative, false positive, and false negative metrics from the confusion matrix, which resulted in the following outcome:

- ✓ No events marked as "0" on input have been predicted as "2".
- ✓ No events marked as "0" on input have been predicted as "1".
- ✓ 1587 marked as "0" on input have been predicted as "0".
- ✓ 711 events marked as "1" on input have been predicted as "2".
- ✓ No events marked as "1" on input have been predicted as "1".
- ✓ 4 events marked as "1" on input have been predicted as "0".
- ✓ No events marked as "2" on input have been predicted as "2".
- ✓ 141 events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".

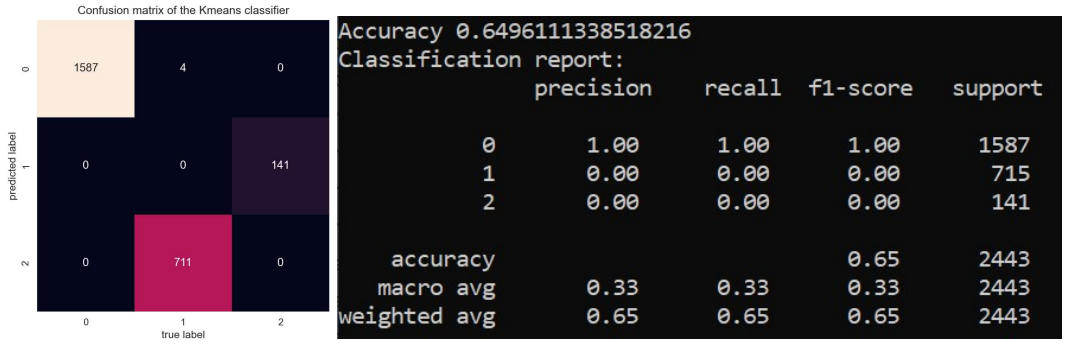


Figure 90 – Accuracy UC3 one fault detection

In conclusion, increasing the sampling rate to 800 milliseconds resulted in a significant improvement in accuracy, which reached 65%. However, there was still a misinterpretation in the clustering between no power fault and occupancy events.

The image below illustrates how the algorithm misinterpreted the no power fault and occupancy events, while the clearance events were correctly clustered.

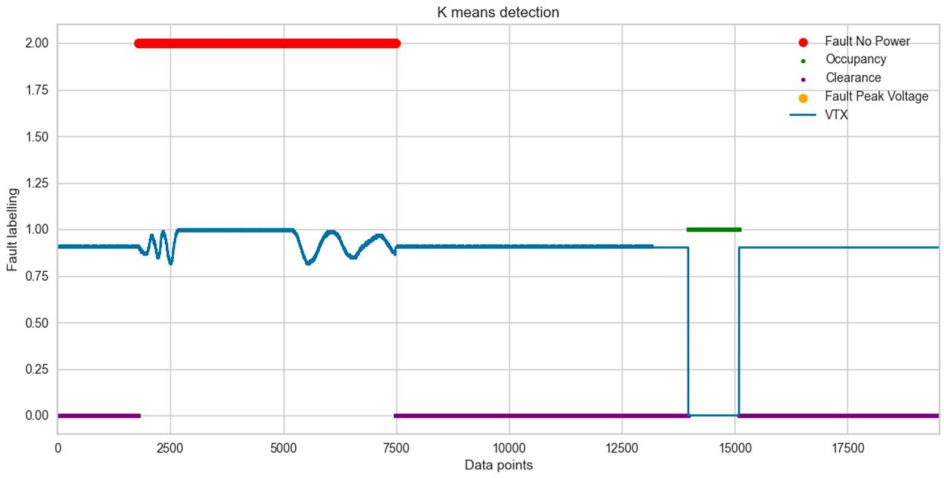


Figure 91 – Graphic one fault detection for UC3

5.6.8 One Fault Detection Use Case 4

For this use case, a dataset consisting of occupancy and clearance events with the injection of a synthesized no power failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 1 minute that corresponds to a window size of 600, to estimate the events.

The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of three distinct states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was utilized to identify the optimal value of k to assign to the K-means algorithm.

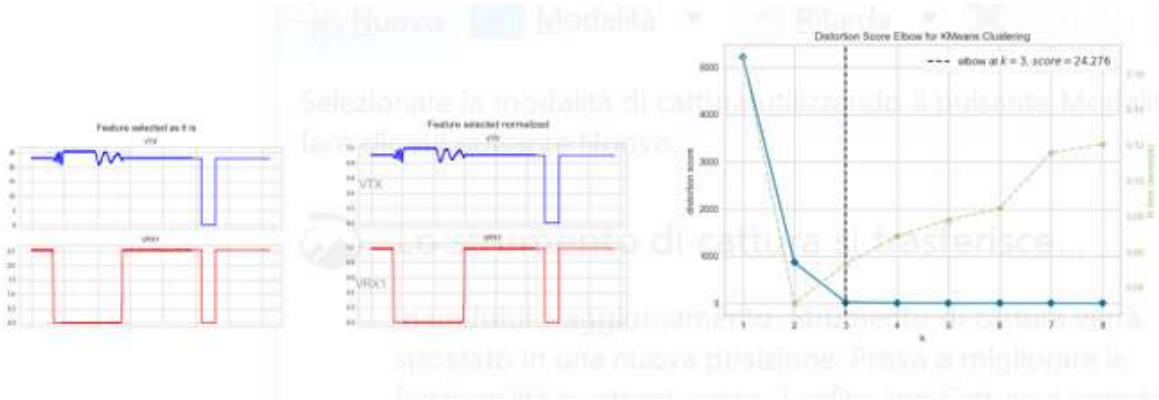


Figure 92 – Optimal k for UC4 one fault detection

The algorithm continued to converge at an optimal k value of 3. Hence, assigning this value to the K-means algorithm would result in three clusters as the outcome.

In this case, the values are treated as 600x600 matrices for the two selected features. As a result, when working with multidimensional arrays, the graphical cluster representation cannot be depicted in either 2-dimensional or 3-dimensional format.

For the purpose of cluster representation, the Python code provided in the appendix implemented dimensionality reduction techniques such as principal component analysis and t-distributed stochastic neighbour embedding. However, this resulted in a loss of information and resolution.

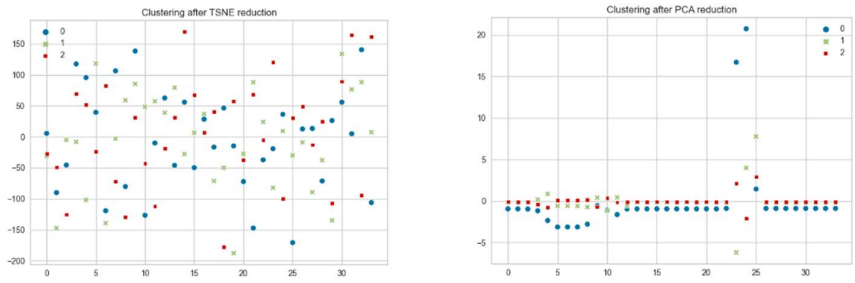
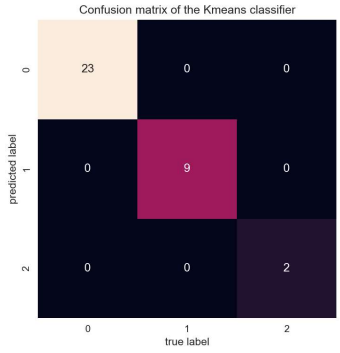


Figure 93 – Cluster representation UC4 one fault detection

With an increased window size of 1 minute, a clear separation of the clusters was obtained because of the dimensionality reduction applied for the sake of representation. The model's performance was evaluated by computing the true positive, true negative, false positive, and false negative metrics from the confusion matrix, which resulted in the following outcome:

- ✓ No events marked as "0" on input have been predicted as "2".
- ✓ No events marked as "0" on input have been predicted as "1".
- ✓ 23 events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "2".
- ✓ 9 events marked as "1" on input have been predicted as "1".
- ✓ No events marked as "1" on input have been predicted as "0".
- ✓ 2 events marked as "2" on input have been predicted as "2".
- ✓ No events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".



Accuracy 1.0
Classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23
1	1.00	1.00	1.00	9
2	1.00	1.00	1.00	2
accuracy			1.00	34
macro avg	1.00	1.00	1.00	34
weighted avg	1.00	1.00	1.00	34

Figure 94 – Accuracy UC4 one fault detection

In conclusion, increasing the sampling rate to 1 minute resulted in optimal accuracy by removing all misinterpreted events, leading to a total precision in detecting the three states. The algorithm was able to discriminate with high precision between the three different states: occupancy, clearance, and the synthesized no power fault, as illustrated in the image below.

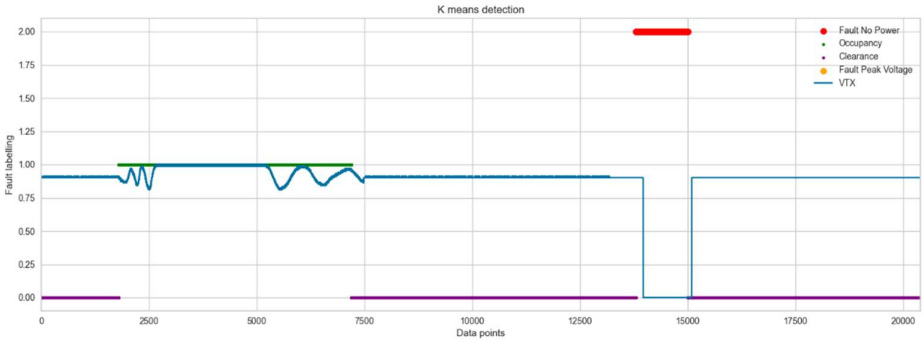


Figure 95 – Graphic one fault detection for UC4

5.6.9 Two Faults Detection Use Case 1

For this use case, a dataset consisting of occupancy and clearance events with the injection of synthesized no power and peak voltage failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 100 milliseconds resulting in a window size of 1, to estimate the events.

The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of four distinct states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was utilized to identify the optimal value of k to assign to the K-means algorithm.

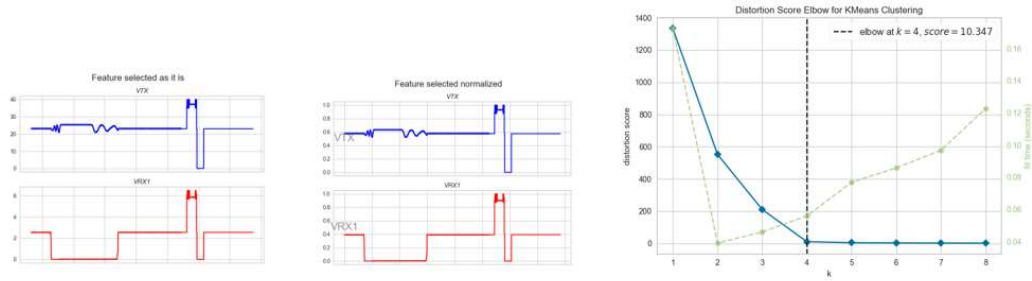


Figure 96 – Optimal k for UC1 two faults detection

The algorithm converged at an optimal value of k equal to 4. Hence, assigning this value to the K-means algorithm would result in four clusters as the outcome.

The 2-dimensional cluster separation was depicted since the values were treated as singular points for the two selected features. This confirmed that the points were separated into four ensembles depending on their similarity, as shown below:

- ✓ Cluster 0: represented in blue and supposed to group occupancy events.
- ✓ Cluster 1: represented in orange and supposed to group clearance events.
- ✓ Cluster 2: represented in green and supposed to group peak voltage fault events.
- ✓ Cluster 3: represented in red and supposed to group no power fault events.

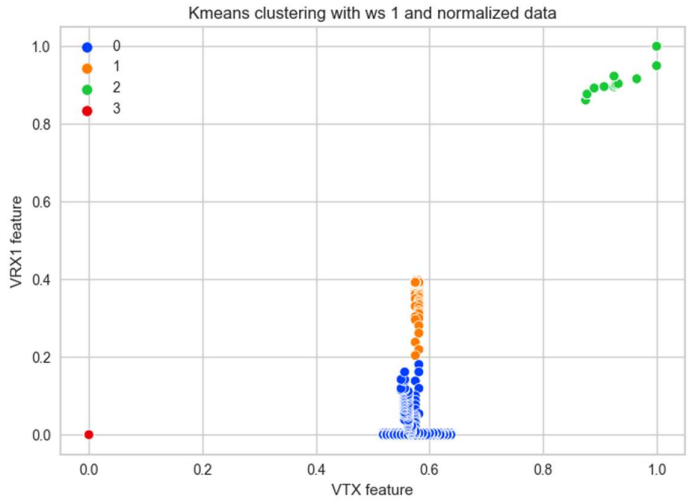
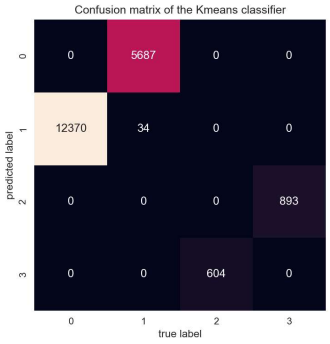


Figure 97 – Cluster representation UC1 two fault detection

The clustering outcome appears to properly group the four cases based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcome:

- ✓ No events marked as "0" on input have been predicted as "3".
- ✓ No events marked as "0" on input have been predicted as "2".
- ✓ 12370 events marked as "0" on input have been predicted as "1".
- ✓ No events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "3".
- ✓ No events marked as "1" on input have been predicted as "2".
- ✓ 34 events marked as "1" on input have been predicted as "1".
- ✓ 5687 events marked as "1" on input have been predicted as "0".
- ✓ 604 events marked as "2" on input have been predicted as "3".
- ✓ No events marked as "2" on input have been predicted as "2".
- ✓ No events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".
- ✓ No events marked as "3" on input have been predicted as "3".
- ✓ 893 events marked as "3" on input have been predicted as "2".
- ✓ No events marked as "3" on input have been predicted as "1".
- ✓ No events marked as "3" on input have been predicted as "0".




```
Accuracy 0.0017357565856646927
Classification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     12370
           1       0.00      0.01      0.00      5721
           2       0.00      0.00      0.00       604
           3       0.00      0.00      0.00       893

    accuracy                           0.00     19588
   macro avg       0.00      0.00      0.00     19588
weighted avg       0.00      0.00      0.00     19588
```

Figure 98 – Accuracy UC1 two faults detection

In conclusion, with a sampling rate of 100 milliseconds and evaluating it with the confusion matrix, there was a complete misinterpretation in detecting the correct state between occupancy, clearance, no power, and peak voltage fault. This resulted in an accuracy of 0%. The image below illustrates the complete misinterpretation in the grouping between no fault and peak voltage failures, as well as between occupancy and clearance events.

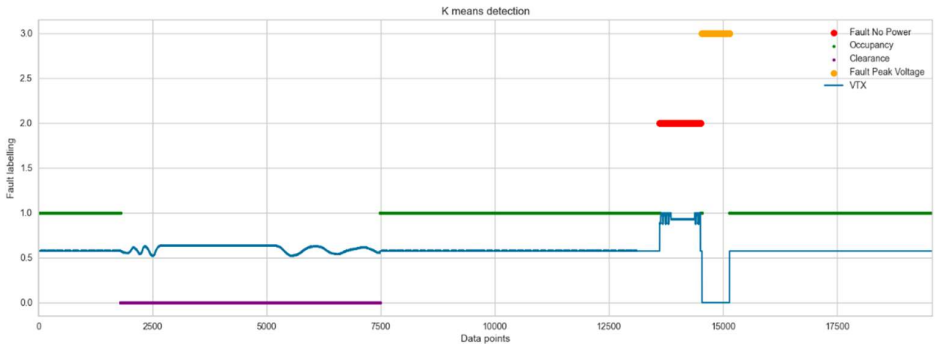


Figure 99 – Graphic two faults detection for UC1

5.6.10 Two Faults Detection Use Case 2

For this use case, a dataset consisting of occupancy and clearance events with the injection of synthesized no power and peak voltage failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 200 milliseconds that means a window size of 2, to estimate the events.

The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of four distinct states. They were then normalized to ensure that they were of the same order of

magnitude. Subsequently, the Elbow method was utilized to identify the optimal value of k to assign to the K-means algorithm.

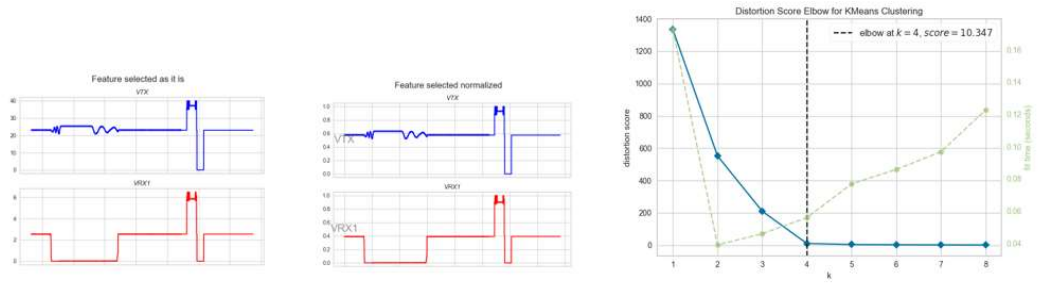


Figure 100 – Optimal k for UC2 two faults detection

The algorithm continued to converge at an optimal k value of 4. Hence, assigning this value to the K-means algorithm would result in four clusters as the outcome. The 3-dimensional cluster separation was depicted since the values were treated as 2x2 matrices for the two selected features. This confirmed that the points were separated into four ensembles depending on their similarity, as shown below:

- ✓ Cluster 0: supposed to group occupancy events.
- ✓ Cluster 1: supposed to group clearance events.
- ✓ Cluster 2: supposed to group peak voltage faults events.
- ✓ Cluster 3: supposed to group no power faults events.

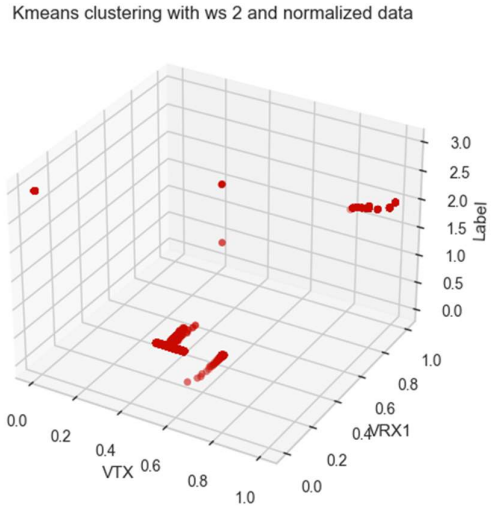
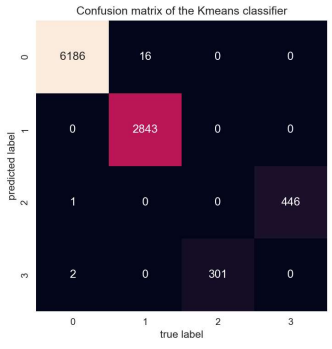


Figure 101 – Cluster representation UC2 two faults detection

The clustering outcome appears to properly group the four cases based on the similarity of the available input data. However, the evaluation of the model's performance resulted in the following outcome:

- ✓ 2 events marked as “0” on input have been predicted as “3”.
- ✓ 1 event marked as “0” on input have been predicted as “2”.
- ✓ No events marked as “0” on input have been predicted as “1”.
- ✓ 6186 events marked as “0” on input have been predicted as “0”.
- ✓ No events marked as “1” on input have been predicted as “3”.
- ✓ No events marked as “1” on input have been predicted as “2”.
- ✓ 2843 events marked as “1” on input have been predicted as “1”.
- ✓ 16 events marked as “1” on input have been predicted as “0”.
- ✓ 301 events marked as “2” on input have been predicted as “3”.
- ✓ No events marked as “2” on input have been predicted as “2”.

  No events marked as “2” on input have been predicted as “1”.
- ✓ No events marked as “2” on input have been predicted as “0”.
- ✓ No events marked as “3” on input have been predicted as “3”.
- ✓ 446 events marked as “3” on input have been predicted as “2”.
- ✓ No events marked as “3” on input have been predicted as “1”.
- ✓ No events marked as “3” on input have been predicted as “0”.




```
Accuracy 0.9217968351199591
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      6189
           1       1.00      0.99      1.00      2859
           2       0.00      0.00      0.00       301
           3       0.00      0.00      0.00       446

    accuracy                           0.92      9795
   macro avg       0.50      0.50      0.50      9795
weighted avg       0.92      0.92      0.92      9795
```

Figure 102 – Accuracy UC2 two faults detection

In conclusion, with an increased sampling rate of 200 milliseconds, there was a significant improvement in accuracy, reaching a value of 92%. The occupancy and clearance events were clustered with almost perfect

precision, with only a few events being misinterpreted. However, there was a complete misunderstanding between the no power and peak voltage failures.

The image below illustrates the complete misinterpretation in grouping between the no fault and peak voltage failures, while the clearance and occupancy events were almost properly discriminated by the model.

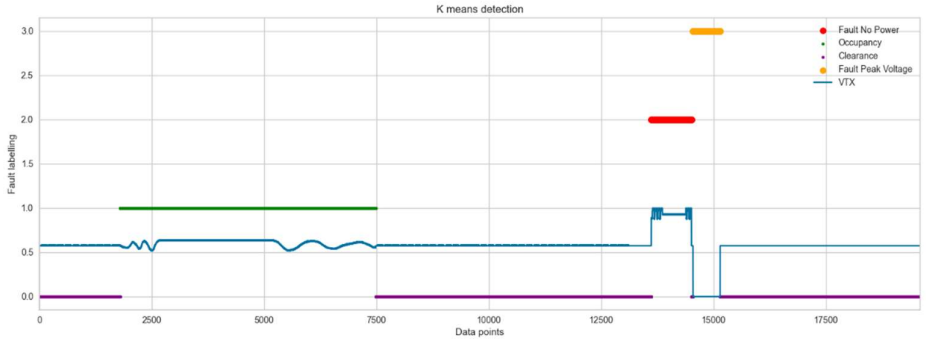


Figure 103 – Graphic two faults detection for UC2

5.6.11 Two Faults Detection Use Case 3

For this use case, a dataset consisting of occupancy and clearance events with the injection of synthesized no power and peak voltage failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 800 milliseconds resulting in a window size of 8, to estimate the events. The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of four distinct states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was utilized to identify the optimal value of k to assign to the K-means algorithm.

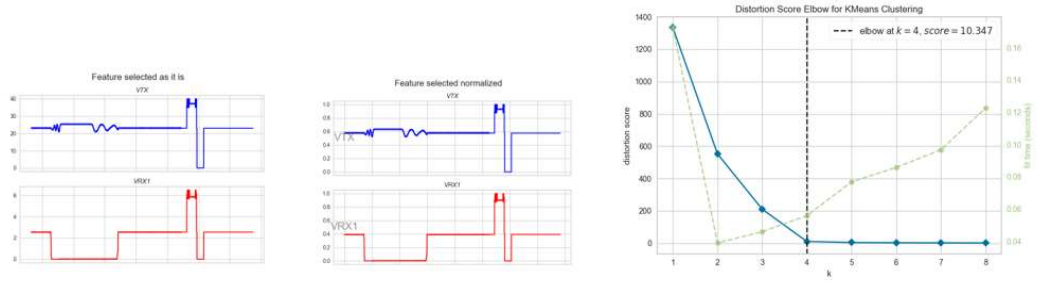


Figure 104 – Optimal k for UC4 one fault detection

The algorithm continued to converge at an optimal k value of 4. Hence, assigning this value to the K-means algorithm would result in four clusters as the outcome.

In this case, the values were treated as 8x8 matrices for the two selected features. Therefore, working with a multidimensional array, the graphical cluster representation cannot be represented in either a 2-dimensional or a 3-dimensional format. To overcome this challenge, dimensionality reduction techniques such as principal component analysis were implemented in the Python code provided in the appendix for the purpose of cluster representation, despite losing some information and resolution.

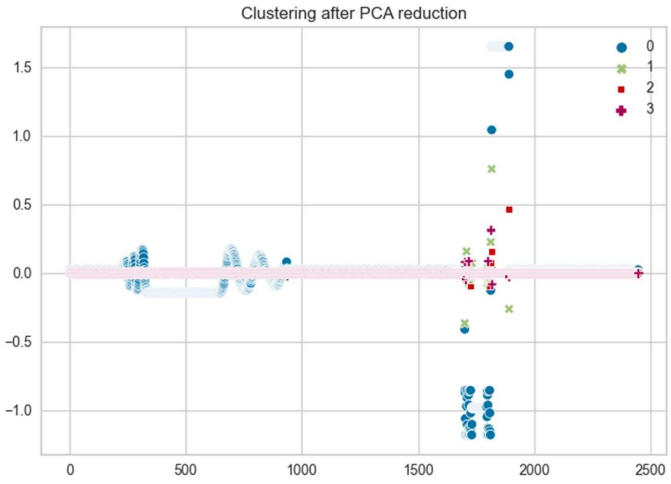
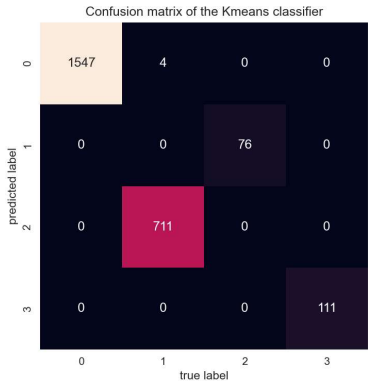


Figure 105 – Cluster representation UC3 two faults detection

The use of dimensionality reduction techniques for cluster representation does not provide a clear indication of whether the clusters have been properly created by the model. Therefore, the model's performance was evaluated by computing the true positive, true negative, false positive, and false negative metrics contained within the confusion matrix, resulting in the following outcome:

- ✓ No event marked as "0" on input have been predicted as "3".
- ✓ No event marked as "0" on input have been predicted as "2".
- ✓ No events marked as "0" on input have been predicted as "1".
- ✓ 1547 events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "3".
- ✓ 711 events marked as "1" on input have been predicted as "2".
- ✓ No events marked as "1" on input have been predicted as "1".

- ✓ 4 events marked as "1" on input have been predicted as "0".
- ✓ No events marked as "2" on input have been predicted as "3".
- ✓ No events marked as "2" on input have been predicted as "2".
- ✓ 76 events marked as "2" on input have been predicted as "1".
- ✓ No events marked as "2" on input have been predicted as "0".
- ✓ 111 events marked as "3" on input have been predicted as "3".
- ✓ No events marked as "3" on input have been predicted as "2".
- ✓ No events marked as "3" on input have been predicted as "1".
- ✓ No events marked as "3" on input have been predicted as "0".




```
Accuracy 0.6770110249081258
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1547
           1       0.00      0.00      0.00       715
           2       0.00      0.00      0.00        76
           3       1.00      1.00      1.00       111

    accuracy                           0.68      2449
   macro avg       0.50      0.50      0.50      2449
weighted avg       0.68      0.68      0.68      2449
```

Figure 106 – Accuracy UC3 two faults detection

In conclusion, increasing the sampling rate up to 800 milliseconds resulted in a decrease in the accuracy value, which reached 68%. The clearance and peak voltage failure events were clustered with almost perfect precision, with only a few events being misinterpreted. However, there was a complete misunderstanding between the occupancy and no power failure events.

The image below illustrates the complete misinterpretation in grouping between occupancy and no power failure events, while clearance and peak voltage failure events were almost properly discriminated by the model.

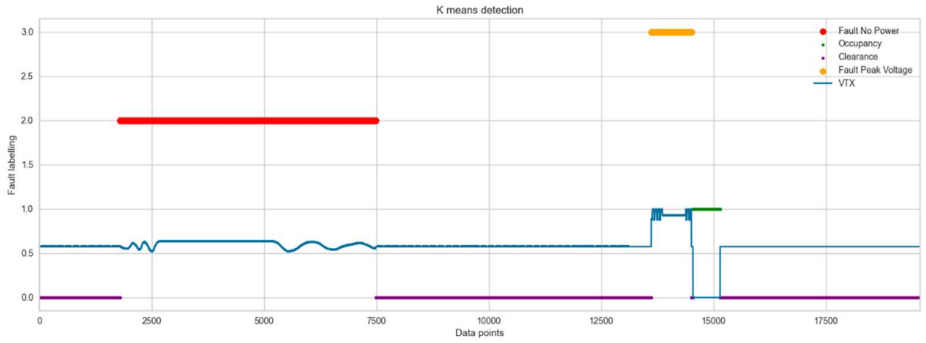


Figure 107 – Graphic two faults detection for UC3

5.6.12 Two Faults Detection Use Case 4

For this use case, a dataset consisting of occupancy and clearance events with the injection of synthesized no power and peak voltage failure was utilized. The K-means algorithm was applied to this dataset using a sample rate of 1 minute corresponding to a window size of 600, to estimate the events.

The two selected features from the available input data, VTX and VRX1, were visualized to confirm the presence of four distinct states. They were then normalized to ensure that they were of the same order of magnitude. Subsequently, the Elbow method was utilized to identify the optimal value of k to assign to the K-means algorithm.

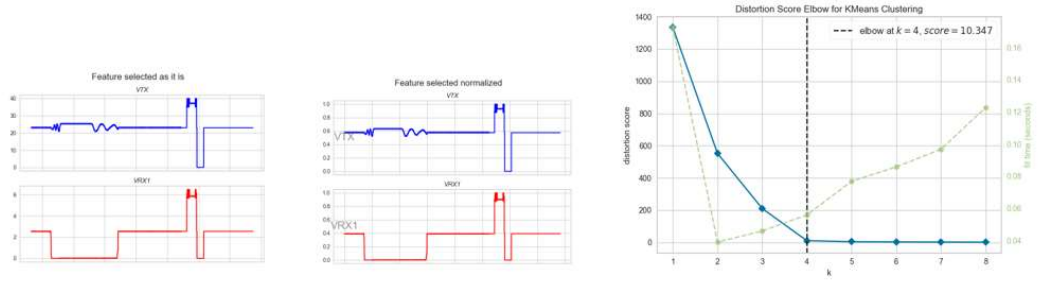


Figure 108 – Optimal k for UC4 two faults detection

The algorithm continued to converge at an optimal k value of 4. Hence, assigning this value to the K-means algorithm would result in four clusters as the outcome.

In this case, the values were treated as 600x600 matrices for the two selected features. Therefore, working with a multidimensional array, the graphical cluster representation cannot be displayed in either a 2-dimensional or a 3-dimensional format.

To overcome this challenge, dimensionality reduction techniques such as principal component analysis were implemented in the Python code provided in the appendix for the purpose of cluster representation, despite losing some information and resolution.

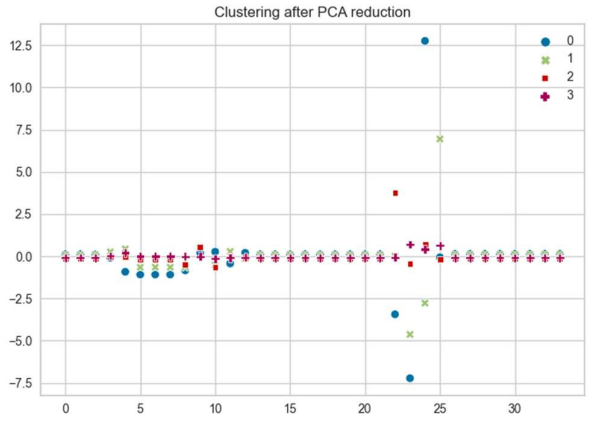


Figure 109 – Cluster representation UC4 two faults detection

Increasing the sampling rate up to 1 minute resulted in a clear separation of the clusters as an outcome of the dimensionality reduction applied for the sake of representation. Therefore, the model's performance was evaluated by computing the true positive, true negative, false positive, and false negative metrics contained within the confusion matrix, resulting in the following outcome:

- ✓ No event marked as "0" on input have been predicted as "3".
- ✓ No event marked as "0" on input have been predicted as "2".
- ✓ No events marked as "0" on input have been predicted as "1".
- ✓ 23 events marked as "0" on input have been predicted as "0".
- ✓ No events marked as "1" on input have been predicted as "3".
- ✓ No events marked as "1" on input have been predicted as "2".
- ✓ 9 events marked as "1" on input have been predicted as "1".
- ✓ No events marked as "1" on input have been predicted as "0".
- ✓ No events marked as "2" on input have been predicted as "3".

- ✓ 1 event marked as “2” on input have been predicted as “2”.
- ✓ No events marked as “2” on input have been predicted as “1”.
- ✓ No events marked as “2” on input have been predicted as “0”.
- ✓ 1 event marked as “3” on input have been predicted as “3”.
- ✓ No events marked as “3” on input have been predicted as “2”.
- ✓ No events marked as “3” on input have been predicted as “1”.
- ✓ No events marked as “3” on input have been predicted as “0”.

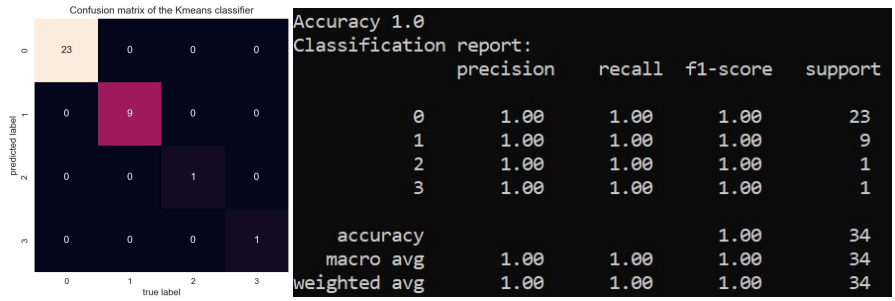


Figure 110 – Accuracy UC4 two faults detection

In conclusion, increasing the sampling rate up to 1 minute resulted in the accuracy reaching the optimal value, thereby removing all misinterpreted events, and leading to a total precision in the detection of the four states. The algorithm demonstrated high precision in discriminating between the four different states, namely occupancy, clearance, no power fault, and peak voltage failure, all of which were synthetically injected, as illustrated in the picture below.

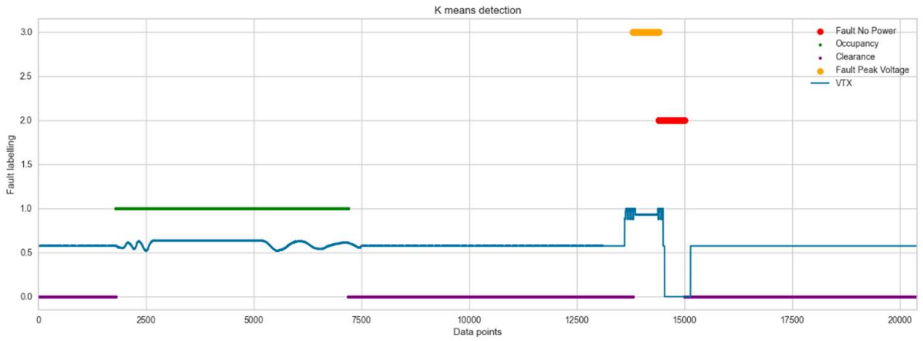


Figure 111 – Graphic two faults detection for UC4

6 Chapter six: Conclusion and future developments

The first topic of industrial research is a crucial topic that every railway company worldwide is currently exploring due to the upcoming switch-off of GSM-R in 2030, which is presently the standard adopted globally for railway communication. The Future Railway Mobile Communication Standard (FRMCS) is already in the specification and development phase, but the 5G technology needed for it is not yet standalone everywhere. The implementation of this new railway standard will entail additional costs for the various railway operators, including the construction of a private backbone, like what was done in the past for GSM-R, in infrastructural terms. Additionally, there will be a need to purchase a dedicated frequency spectrum to ensure that the system remains private and to limit any ever-evolving cyber attacks that could disrupt daily operations.

This research is positioned in the context of finding a solution based on the existing public networks of telecommunications providers, which serve railway lines with low traffic density. The choice of such lines was made because the public network would have to manage a smaller amount of data flow exchange between the railway subsystems, and it would be adopted by railway entities that do not have a large financial sum for the realization of an ad-hoc network based on FRMCS. The research conducted and the tests evaluated in a real environment have demonstrated the feasibility of leveraging the existing 4G public telecom network if there is a sufficient level of coverage. This would involve realizing a redundant communication system based on the adoption of different telecom provider networks to reduce communication loss and consequent service interruption, which can be a challenge for radio frequency technology.

From a legislative standpoint, the current LTE standard cannot assign dedicated radio resources or prioritize them, as it is a public network where resources are shared. Therefore, it is essential to explore techniques to reduce access to railway transmission by introducing tunnelling, encryption, and the creation of a private Access Point Name (APN), as requested to telecom providers. If the solution developed in this research is adopted for commercial service, it is likely that greater requirements will be demanded of telecom operators to ensure a continuous and more reliable service in terms of guaranteed bandwidth and quality of service. This solution will continue to be studied within the company in the coming years.

The same radio frequency link that is used for signalling communication will also serve the transmission of non-vital maintenance data. Therefore, the second part of the industrial research focused on developing an algorithm based on machine learning techniques, realized through the Python script attached in the Appendix. The objective was to discriminate between nominal and anomalous states characterizing the STDS-AF railway equipment chosen for diagnosis, as it will be installed on a large scale in the next few years. The constraint of the absence of fault events coming from the field was overcome by inserting synthetic failures in the available input data, allowing for the evaluation of the model in different operating scenarios. Since the available input data was composed of structured and unlabelled format, an unsupervised machine learning technique was adopted to solve the clustering problem. The model was fed with different datasets in the previous chapter and evaluated in terms of performance, considering various sampling rates that were progressively increased in the estimation of events. The aim was to obtain a unique window size that could be applied and compatible in terms of duration with a hypothetical occupancy/clearance event or fault as well.

Input data	Accuracy ws = 1	Accuracy ws = 2	Accuracy ws = 8	Accuracy ws = 60	Accuracy ws = 100	Accuracy ws = 600
No Fault	0%	0%	100%	100%	100%	**100%**
Power Fault	6%	6%	65%	100%	100%	**100%**
Power Fault + Peak Fault	0%	0%	68%	63%	63%	**100%**

Table 9- Accuracy of the Fault detection model

To determine the accuracy of the model, a recursive Python script code was employed, which is attached in the Appendix. The code was used to compute accuracy for all the datasets fed to the model, over a range of sampling rates configured from the minimum window size of 100 milliseconds up to 1 minute. The 1-minute window size is more compatible with occupancy produced by the movement of a wagon within the track circuit, for instance.

As shown in the table above, increasing the window size for the estimation of events resulted in an improvement in the precision and accuracy of the model. This achievement met the intended objective of discriminating between nominal and anomalous behaviour of the STDS-AF railway equipment with good

performance. When treating the data with a sampling rate of 1 minute, which is compatible in terms of duration with a real occupancy event, the model achieved optimal performance for all the cases in which it was fed.

The results were obtained by feeding the algorithm with synthesized data for the failure events part, as no faults had been experienced on the currently installed railway component. A future improvement that needs to be achieved is the evaluation of the algorithm's behaviour as soon as data containing field failures become available.

Additionally, continuously collecting logs with real faults coming from the field should enable the implementation of a predictive diagnosis of the device, taking into consideration the time and historical detection events.

Bibliography

REFERENCES

[1] A.K.Jain, "Data Clustering: 50 Years Beyond K-means", Pattern Recognition Letters, Vol 31 pp651-650, 2010.

[2] Laurens van der Maaten, Geoffrey Hinton: “Visualizing Data using t-SNE”, Journal of Machine Learning Research 9 (2008)

[3] Claudio Gallicchio: “Reservoir Recurrent Neural Networks” SSIE “Silvano Pupolin” lecture, July 2021

[4] Karen Hao: “Training a single AI model can emit as much carbon as five cars in their lifetimes”, June 2019

[5] Sumi Ghosh, Sanjay Jumar Dubey: “Comparative Analysis of K-Means and Fuzzy C-Means Algorithms", International Journal of Advanced Computer Science and Applications, Vol 4, No 4, 2013

[6] Manuel Roveri: “Intelligent Embedded and Edge Computing Systems” SSIE “Silvano Pupolin” lecture, July 2021

[7] Manuel Roveri: “Learning in nonstationary environments” SSIE “Silvano Pupolin” lecture, July 2021

[8] B.S. Dhillon: “Engineering Maintenance: A Modern Approach” book

[9] IEC 62443-3-3 International Standard – Industrial communication networks – Part 3.3 – System security requirements and security levels

[10] Qingsong Han1, Huifang Li1, Wei Dong1, Yafei Luo2, Yuanqing Xia1: "On Fault Prediction Based on Industrial Big Data", 36th Chinese Control Conference, July 2017

[11] Yuxin Gao, Yong Yang, Yuan Ma, Weixiang Xu: "Research on Intelligent Diagnosis of Railway Turnout Based on Fast-DTW under Big Data Monitoring" International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT), 2021

[12] Yubo Zhang: “Research on Sound Signal Recognition and Its Application in Mechanical Operation State Prediction” Hunan University, Master's Thesis, 2004

[13] Zhizhe Zhang, Bo Li, Tianhua Xu, Cong Chen, Feng Wang: "Deep Forest based Fault Diagnosis for Railway Turnout Systems in The Case of Limited Fault Data" IEEE, 2019

[14] Rahman Shafique, Hafeez-Ur-Rehman Siddiqui, Furqan Rustam, Saleem Ullah, Muhammad Abubakar Siddique, Ernesto Lee, Imran Ashraf, Sandra Dudley: "A Novel Approach to Railway Track Faults Detection Using Acoustic Analysis" MDPI Article, 2021

[15] Iwo Doboszewski, Simon Fossier, Christophe Marsala: "Data Driven Detection of Railway Point Machines Failures" IEEE Symposium Series of Computational Intelligence (SSCI), December 2019

[16] Zaharah A. Bukhsh, Irina Stipanovic, Aaqib Saeed: "A machine learning approach for maintenance prediction of railway assets" 7th Transport Research Arena TRA 2018

[17] Hamad Alawad, Sakdirat Kaewunruen, Min An: "Learning from Accidents: Machine Learning for Safety at Railway Stations" IEEE 2019

[18] Tim de Bruin, Kim Verbert, Robert Babuska: "Railway Track Circuit Fault Diagnosis Using Recurrent Neural Networks" IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 28, NO. 3, MARCH 2017

[19] Qin Liu, Tian Liang, Venkata Dinavahi: "Real-Time Hierarchical Neural Network Based Fault Detection and Isolation for High-Speed Railway System Under Hybrid AC/DC Grid" IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 35, NO. 6, DECEMBER 2020

[20] Zijian Guo, Yiming Wan, Hao Ye: "An Unsupervised Fault-Detection Method for Railway Turnouts" IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 69, NO. 11, NOVEMBER 2020

[21] Macilio da Silva Ferreira, Lucio F. Vismari, Paulo S. Cugnasca, Jorge R. de Almeida Jr, João B. Camargo Jr, Guilherme Kallemback: "A comparative analysis of unsupervised learning techniques for anomaly detection in railway systems 18th IEEE International Conference on Machine Learning and Applications (ICMLA), 2019

[22] Nagdev Amruthnath, Tarun Gupta: "A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance" 5th International Conference on Industrial Engineering and Applications, 2018

[23] Dong Yan, Xiukun Wei, Guorui Zhai: "RUL Prediction for Railway Vehicle Bearings Based on Fault Diagnosis" IEEE, 2017

[24] Tamer S. Abdelgayed, Walid G. Morsi, Tarlochan S. Sidhu: "Fault Detection and Classification Based on Co-training of Semi-supervised Machine Learning" IEEE TRANSACTIONS ON INDUSTRIAL

ELECTRONICS, VOL. 65, NO. 2, FEBRUARY 2018

[25] Yiwei Guo: "A Reinforcement Learning Approach to Train Timetabling for Inter-City High Speed Railway Lines" 5th International Conference on Intelligent Transportation Engineering, 2020

Appendix

This section includes the Python code developed with the objective of creating an algorithm to detect nominal and anomalous states of the STDS-AF railway equipment referred to in this thesis.

```
#import of the libraries
from ctypes import sizeof
from datetime import datetime, timedelta
from pandas import DataFrame
import numpy as np
import pandas as pd
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
from collections import Counter
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.spatial import distance_matrix
from mpl_toolkits.mplot3d import Axes3D
from numpy import array
from numpy import size
from scipy.cluster.vq import whiten, kmeans, vq, kmeans2
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from itertools import cycle
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score, accuracy_score, precision_score, recall_score
from sklearn.metrics import classification_report
from yellowbrick.classifier import ClassificationReport
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
from sklearn import decomposition
import collections

#open the file and normalize the values and then selecting the features of interest
rowlist = list()
fname = input("Enter the file wanted to be processed: ")
df = pd.read_csv(fname, sep=";", decimal=",")
#Loop to have a row every 100 ms that is not the case of the input data
#convert the Timestamp coloumn from dtype object to dtype datetime
fmt ='%Y-%m-%d %H:%M:%S.%f'
df.rename(columns = {'Timestamp [+02:00]':'Timestamp'}, inplace = True)
df["Timestamp"] = pd.to_datetime(df["Timestamp"], format = fmt)
```

```
#count the number of rows in the input excel file
total_rows = len(df.axes[0]) + 1
#subtract all the data to check the index of the row where the difference is higher than 100 ms
i = 1
for i in range(len(df)-2):
    start_time = df.at[i, "Timestamp"]
    end_time = df.at[i+1, "Timestamp"]
    delta_raw = (end_time - start_time)
    delta = int((delta_raw.seconds * 1000) + (delta_raw.microseconds / 1000))
    n_rows_add = int(round(delta/100))
    rowlist.append(n_rows_add)
#Start of the part of code to have the lenght of the list concerning the rows to add equal to the number of
#lines of the file opened
for element in range((total_rows-len(rowlist))-1):
    rowlist.append(0)
#creation of a Dataframe related to the rows to add which has been added to the starting dataframe
#in the last coloumn
    dfrows = pd.DataFrame(rowlist)
    df["Repeat"] = dfrows
# replication of the row considering the value reported on the new coloumn "Repeat" created
out = df.iloc[np.repeat(np.arange(len(df)), df["Repeat"])].drop("Repeat", axis=1).reset_index(drop=True)
# replaced of the first timestamp of the file for all the lines and then resampled adding t*100ms
# each loop
for t in range(len(df)-2):
        out.at[t, "Timestamp"] = out.at[1, "Timestamp"]
for l in range(len(df)-2):
        out.at[l, "Timestamp"] = out.at[l, "Timestamp"] + timedelta(milliseconds=l*100)
#Closure of the part of code to have a row every 100 ms that is not the case of the input data
#take a look at the dataset not normalized with time not processed and time processed to have constant
sampling rate
dataset_view=df.iloc[:,0:30]
print(dataset_view)
dataset_view_time=out.iloc[:,0:30]
print(dataset_view_time)
cdf = dataset_view_time[['Timestamp','VTX','ITX','VRX1','VRX2','ExtPS','IntPS']]
cdf.head(7)
fig, axs = plt.subplots(2,2)
fig.suptitle('Feature selected as it is', fontsize=16)
axs[0,0]=plt.subplot(2,1,1)
axs[0,0].text(0.5, 0.5, 'VTX', ha='center', va='center',size=20, alpha=.5)
axs[0,0].set_title('VTX', fontstyle='italic')
axs[0,0]=plt.plot(cdf.Timestamp, cdf.VTX,  color="blue")
plt.xticks(color='w')
axs[1,1]=plt.subplot(2,1,2)
axs[1,1].text(0.5, 0.5, 'VRX1', ha='center', va='center',size=20, alpha=.5)
axs[1,1].set_title('VRX1', fontstyle='italic')
axs[1,1] = plt.plot(cdf.Timestamp, cdf.VRX1,  color="red")
```

```
plt.xticks(color='w')
plt.show()
#Normalization of the input data once selected the features, to have all the data with the same magniture order
x = df.iloc[:,1:5]
x_features = x[["VTX","VRX1"]]
normalized_df=(x-x.min())/(x.max()-x.min())
norm_df = normalized_df[["VTX","VRX1"]]
#take a look at the dataset once normalized
fig, axs = plt.subplots(2,2)
fig.suptitle('Feature selected normalized', fontsize=16)
axs[0,0]=plt.subplot(2,1,1)
axs[0,0].text(0.5, 0.5,'VTX', ha='center', va='center',size=20, alpha=.5)
axs[0,0].set_title('VTX', fontstyle='italic')
axs[0,0]=plt.plot(norm_df.index, norm_df.VTX,  color="blue")
plt.xticks(color='w')
axs[1,1]=plt.subplot(2,1,2)
axs[1,1].text(0.5, 0.5,'VRX1', ha='center', va='center',size=20, alpha=.5)
axs[1,1].set_title('VRX1', fontstyle='italic')
axs[1,1] = plt.plot(norm_df.index, norm_df.VRX1,  color="red")
plt.xticks(color='w')
plt.show()
#determine the best value for the clustering through the Elbow method for Kmeans
x_array=norm_df["VTX"].to_numpy().reshape(-1,1*1)
y_array=norm_df["VRX1"].to_numpy().reshape(-1,1*1)
data = np.concatenate((x_array,y_array),axis =1)
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,9)).fit(data)
visualizer.show()
#Choose the best k value aligned with the outcome of the Elbow method
KV= input("Enter the size of the best cluster values : ")
KV=int(KV)
if KV ==2:
   #Loop within the window size "ws" to be used for the process (ws=1 means 100ms, ws=2 means 200ms and so on)
   accuracy_list=list()
   for ws in range(1,600,10):
      ws=int(ws)
   # part of the code to verify if dataframe lenght and window size chosed are divisible by giving an integer as output
      r_add=list()
      if len(df)/ws is not int:
         #print("ws not supported with the size of the input data.Input data modified appending rows at the EOD")
         div_value=int(round((len(df)/ws)))+1
         newlendf = div_value * ws
         r_add= newlendf - len(df)
```

```
        last_row= x.tail(r_add)
        df_lastrows = pd.DataFrame(last_row)
        df_reworked = pd.concat([x, df_lastrows], ignore_index=True)
        #Normalization of the input data to have all the data with the same magniture order
        normalized_df=(df_reworked-df_reworked.min())/(df_reworked.max()-df_reworked.min())
    else:
        normalized_df=(x-x.min())/(x.max()-x.min())
    norm_df = normalized_df[["VTX","VRX1"]]
    #norm_df.to_csv('out_balanced_plus_fault.csv', index=False)
    #norm_df = x[["VTX","VRX1"]]
    #norm_df.head(100)
    x_array1D=norm_df["VTX"].to_numpy().reshape(-1,1*1)
    y_array1D=norm_df["VRX1"].to_numpy().reshape(-1,1*1)
    data1D=np.concatenate((x_array1D,y_array1D),axis =1)
    x_array=norm_df["VTX"].to_numpy().reshape(-1,ws*1)
    y_array=norm_df["VRX1"].to_numpy().reshape(-1,ws*1)
    #print("this is converted Vtx array", x_array)
    #print("this is converted VRx1 array", y_array)
    data = np.concatenate((x_array,y_array),axis =1)
    #print("this is the concatenation od the two array", data)
    #Definition f an empty list to collect the labels of the input data
    act_label=list()
    #part of code to determine the distribution of the events in the input data in case of k=2
    #part of code to determine the distribution of the events with a granularity of 100ms
    diff_list= list()
    for j in range(0,len(y_array1D)):
        if ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and ((x_array1D[j] <= 0.89  and x_array1D[j]
>= 0.8)) or ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and x_array1D[j] >= 0.9120)):
                diff_list.append(1)
        elif (y_array1D[j] > 0.96 and x_array1D[j] < 0.9120) or (y_array1D[j] > 0.96 and x_array1D[j] >
0.9019):
            diff_list.append(0)
        else:
            diff_list.append(1)
    df_labelled = norm_df.assign(Trackside_status = diff_list)
    df_labelled.to_csv('K2_LABELLED.csv', index=False)
    diff_array=np.asarray(diff_list)
    #print(diff_list)
    #part of code to determine the distribution of the events with grouping the granularity depending on
the window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    #print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        if c1 > c0:
        #if np.count_nonzero(newdiff_array[v] == 1) > np.count_zero(newdiff_array[v]) :
```

```
            act_label.append(1)
        else:
            act_label.append(0)
    #determine the following attributes to get the return values
    #labels_ : gives predicted class labels (cluster) for each data point
    #cluster_centers_ : Location of the centroids on each cluster. The data point in a cluster will be close to the centroid of that cluster. As we have two features and four clusters, we should get four locations.
    #inertia_ : gives within-cluster sum of squares. This is a total of the within-cluster sum of squares for all c
    kmeans = KMeans(n_clusters=KV, init='k-means++',max_iter=300,  n_init=10, random_state=0).fit(data)
    #print(kmeans.inertia_)
    #print(kmeans.n_iter_)
    #print(kmeans.cluster_centers_)
    centroids = kmeans.cluster_centers_
    #print(act_label)
    #print(kmeans.labels_)
    #Get each cluster size
    #print(Counter(kmeans.labels_))
    acs=accuracy_score(act_label, kmeans.labels_)
    accuracy_list.append(acs)
  print(accuracy_list)
  ws_list=list(range(60))
  plt.plot(ws_list, accuracy_list)
  plt.title('Accuracy Vs Window Size')
  plt.xlabel('Window Size')
  plt.ylabel('Accuracy')
  plt.show()

if KV ==3:
  #Loop within the window size "ws" to be used for the process (ws=1 means 100ms, ws=2 means 200ms and so on)
  accuracy_list=list()
  for ws in range(1,600,10):
    ws=int(ws)
  # part of the code to verify if dataframe lenght and window size chosed are divisible by giving an integer as output
    r_add=list()
    if len(df)/ws is not int:
      #print("ws not supported with the size of the input data.Input data modified appending rows at the EOD")
      div_value=int(round((len(df)/ws)))+1
      newlendf = div_value * ws
      r_add= newlendf - len(df)
      last_row= x.tail(r_add)
      df_lastrows = pd.DataFrame(last_row)
      df_reworked = pd.concat([x, df_lastrows], ignore_index=True)
```

```
        #Normalization of the input data to have all the data with the same magniture order
        normalized_df=(df_reworked-df_reworked.min())/(df_reworked.max()-df_reworked.min())
    else:
        normalized_df=(x-x.min())/(x.max()-x.min())
    norm_df = normalized_df[["VTX","VRX1"]]
    #norm_df.to_csv('out_balanced_plus_fault.csv', index=False)
    #norm_df = x[["VTX","VRX1"]]
    #norm_df.head(100)
    x_array1D=norm_df["VTX"].to_numpy().reshape(-1,1*1)
    y_array1D=norm_df["VRX1"].to_numpy().reshape(-1,1*1)
    data1D=np.concatenate((x_array1D,y_array1D),axis =1)
    x_array=norm_df["VTX"].to_numpy().reshape(-1,ws*1)
    y_array=norm_df["VRX1"].to_numpy().reshape(-1,ws*1)
    data = np.concatenate((x_array,y_array),axis =1)
    #Definition f an empty list to collect the labels of the input data
    act_label=list()
    #part of code to determine the distribution of the events in the input data in case of k=3
    #part of code to determine the distribution of the events with a granularity of 100ms
    diff_list= list()
    for j in range(0,len(y_array1D)):
        if x_array1D[j] == 0 and y_array1D[j] == 0 :
            diff_list.append(2)
        elif ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and ((x_array1D[j] <= 0.89  and x_array1D[j]
>= 0.8)) or ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and x_array1D[j] >= 0.9120)):
            diff_list.append(1)
        elif (y_array1D[j] > 0.96 and x_array1D[j] < 0.9120) or (y_array1D[j] > 0.96 and x_array1D[j] >
0.9019):
            diff_list.append(0)
        else:
            diff_list.append(1)
    df_labelled = norm_df.assign(Trackside_status = diff_list)
    df_labelled.to_csv('K3_LABELLED.csv', index=False)
    diff_array=np.asarray(diff_list)
    #print(diff_list)
    #part of code to determine the distribution of the events with grouping the granularity depending on
the window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    #print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        c2 = collections.Counter(newdiff_array[v])[2]
        if c1 > c0 and c1 > c2:
            act_label.append(1)
        elif c2 > c0 and c2 > c1:
            act_label.append(2)
        else:
```

```
            act_label.append(0)
        #determine the following attributes to get the return values
        #labels_ : gives predicted class labels (cluster) for each data point
        #cluster_centers_ : Location of the centroids on each cluster. The data point in a cluster will be close
to the centroid of that cluster. As we have two features and four clusters, we should get four locations.
        #inertia_ : gives within-cluster sum of squares. This is a total of the within-cluster sum of squares for
all c
        kmeans = KMeans(n_clusters=KV, init='k-means++',max_iter=300,  n_init=10,
random_state=0).fit(data)
        #print(kmeans.inertia_)
        #print(kmeans.n_iter_)
        #print(kmeans.cluster_centers_)
        centroids = kmeans.cluster_centers_
        #print(act_label)
        #print(kmeans.labels_)
        #Get each cluster size
        #print(Counter(kmeans.labels_))
        acs=accuracy_score(act_label, kmeans.labels_)
        accuracy_list.append(acs)
    print(accuracy_list)
    ws_list=list(range(60))
    plt.plot(ws_list, accuracy_list)
    plt.title('Accuracy Vs Window Size')
    plt.xlabel('Window Size')
    plt.ylabel('Accuracy')
    plt.show()

if KV ==4:
    #Loop within the window size "ws" to be used for the process (ws=1 means 100ms, ws=2 means
200ms and so on)
    accuracy_list=list()
    for ws in range(1,600,10):
        ws=int(ws)
    # part of the code to verify if dataframe lenght and window size chosed are divisible by giving an integer
as output
        r_add=list()
        if len(df)/ws is not int:
            #print("ws not supported with the size of the input data.Input data modified appending rows at the
EOD")
            div_value=int(round((len(df)/ws)))+1
            newlendf = div_value * ws
            r_add= newlendf - len(df)
            last_row= x.tail(r_add)
            df_lastrows = pd.DataFrame(last_row)
            df_reworked = pd.concat([x, df_lastrows], ignore_index=True)
            #Normalization of the input data to have all the data with the same magniture order
            normalized_df=(df_reworked-df_reworked.min())/(df_reworked.max()-df_reworked.min())
```

```
    else:
        normalized_df=(x-x.min())/(x.max()-x.min())
    norm_df = normalized_df[["VTX","VRX1"]]
    #norm_df.to_csv('out_balanced_plus_fault.csv', index=False)
    #norm_df = x[["VTX","VRX1"]]
    #norm_df.head(100)
    x_array1D=norm_df["VTX"].to_numpy().reshape(-1,1*1)
    y_array1D=norm_df["VRX1"].to_numpy().reshape(-1,1*1)
    data1D=np.concatenate((x_array1D,y_array1D),axis =1)
    x_array=norm_df["VTX"].to_numpy().reshape(-1,ws*1)
    y_array=norm_df["VRX1"].to_numpy().reshape(-1,ws*1)
    #print("this is converted Vtx array", x_array)
    #print("this is converted VRx1 array", y_array)
    data = np.concatenate((x_array,y_array),axis =1)
    #print("this is the concatenation od the two array", data)
    #Definition f an empty list to collect the labels of the input data
    act_label=list()
    #part of code to determine the distribution of the events in the input data in case of k=4
    #part of code to determine the distribution of the events with a granularity of 100ms
    diff_list= list()
    for j in range(0,len(y_array1D)):
        if x_array1D[j] == 0 and y_array1D[j] == 0 :
            diff_list.append(2)
        elif (y_array1D[j] < 0.4 and y_array1D[j] > 0.378) and (x_array1D[j] < 0.59  or x_array1D[j] >
0.575):
            diff_list.append(0)
        elif (y_array1D[j] < 0.09 and (x_array1D[j] <= 0.55  and x_array1D[j] >= 0.5)) or (y_array1D[j]
< 0.09 and (x_array1D[j] <= 0.6375 and x_array1D[j] >= 0.575)) :
            diff_list.append(1)
        elif (y_array1D[j] > 0.45 and x_array1D[j] > 0.64):
            diff_list.append(3)
        else:
         diff_list.append(1)
    df_labelled = norm_df.assign(Trackside_status = diff_list)
    df_labelled.to_csv('K4_LABELLED.csv', index=False)
    diff_array=np.asarray(diff_list)
    #print(diff_list)
    #part of code to determine the distribution of the events with grouping the granularity depending on
the window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    #print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        c2 = collections.Counter(newdiff_array[v])[2]
        c3 = collections.Counter(newdiff_array[v])[3]
        if c1 > c0 and c1 > c2 and c1 > c3:
```

```
            act_label.append(1)
        elif c2 > c0 and c2 > c1 and c2 > c3:
            act_label.append(2)
        elif c3 > c0 and c3 > c1 and c3 > c2:
            act_label.append(3)
        else:
            act_label.append(0)
    #determine the following attributes to get the return values
    #labels_ : gives predicted class labels (cluster) for each data point
    #cluster_centers_ : Location of the centroids on each cluster. The data point in a cluster will be close to the centroid of that cluster. As we have two features and four clusters, we should get four locations.
    #inertia_ : gives within-cluster sum of squares. This is a total of the within-cluster sum of squares for all c
    kmeans = KMeans(n_clusters=KV, init='k-means++',max_iter=300,  n_init=10, random_state=0).fit(data)
    #print(kmeans.inertia_)
    #print(kmeans.n_iter_)
    #print(kmeans.cluster_centers_)
    centroids = kmeans.cluster_centers_
    #print(act_label)
    #print(kmeans.labels_)
    #Get each cluster size
    #print(Counter(kmeans.labels_))
    acs=accuracy_score(act_label, kmeans.labels_)
    accuracy_list.append(acs)
print(accuracy_list)
ws_list=list(range(60))
plt.plot(ws_list, accuracy_list)
plt.title('Accuracy Vs Window Size')
plt.xlabel('Window Size')
plt.ylabel('Accuracy')
plt.show()

#Chose the window size "ws" to be used for the process (ws=1 means 100ms, ws=2 means 200ms and so on)
ws= input("Enter the window size to be used for the process : ")
ws=int(ws)
# part of the code to verify if dataframe lenght and window size chosed are divisible by giving an integer as output
r_add=list()
if len(df)/ws is not int:
    print("Lenght of dataframe might be processed appending row at the end of the file to outcome an integer from the division")
    div_value=int(round((len(df)/ws)))+1
    newlendf = div_value * ws
    r_add= newlendf - len(df)
    last_row= x.tail(r_add)
```

```
    df_lastrows = pd.DataFrame(last_row)
    df_reworked = pd.concat([x, df_lastrows], ignore_index=True)
    #Normalization of the input data to have all the data with the same magniture order
    normalized_df=(df_reworked-df_reworked.min())/(df_reworked.max()-df_reworked.min())
else:
    normalized_df=(x-x.min())/(x.max()-x.min())
norm_df = normalized_df[["VTX","VRX1"]]
norm_df.to_csv('out_balanced_plus_fault.csv', index=False)
#norm_df = x[["VTX","VRX1"]]
#norm_df.head(100)
x_array1D=norm_df["VTX"].to_numpy().reshape(-1,1*1)
y_array1D=norm_df["VRX1"].to_numpy().reshape(-1,1*1)
data1D=np.concatenate((x_array1D,y_array1D),axis =1)
x_array=norm_df["VTX"].to_numpy().reshape(-1,ws*1)
y_array=norm_df["VRX1"].to_numpy().reshape(-1,ws*1)
print("this is converted Vtx array", x_array)
print("this is converted VRx1 array", y_array)
data = np.concatenate((x_array,y_array),axis =1)
print("this is the concatenation od the two array", data)

#Definition of an empty list to collect the labels of the input data
act_label=list()
#part of code to determine the distribution of the events in the input data in case of k=2
#part of code to determine the distribution of the events with a granularity of 100ms
diff_list= list()
if KV == 2:
    for j in range(0,len(y_array1D)):
        if ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and ((x_array1D[j] <= 0.89  and x_array1D[j] >=
0.8)) or ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and x_array1D[j] >= 0.9120)):
            diff_list.append(1)
        elif (y_array1D[j] > 0.96 and x_array1D[j] < 0.9120) or (y_array1D[j] > 0.96 and x_array1D[j] >
0.9019):
          diff_list.append(0)
        else:
           diff_list.append(1)
    diff_array=np.asarray(diff_list)
    print(diff_list)
#part of code to determine the distribution of the events with grouping the granularity depending on the
window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        if c1 > c0:
        #if np.count_nonzero(newdiff_array[v] == 1) > np.count_zero(newdiff_array[v]) :
            act_label.append(1)
```

```
        else:
            act_label.append(0)

# Part of code to determine the distribution of the events in the input data in case of k=3
if KV ==3:
    for j in range(0,len(y_array1D)):
        if x_array1D[j] == 0 and y_array1D[j] == 0 :
            diff_list.append(2)
        elif ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and ((x_array1D[j] <= 0.89  and x_array1D[j]
>= 0.8)) or ((y_array1D[j] <= 0.96 and y_array1D[j] >= 0) and x_array1D[j] >= 0.9120)):
            diff_list.append(1)
        elif (y_array1D[j] > 0.96 and x_array1D[j] < 0.9120) or (y_array1D[j] > 0.96 and x_array1D[j] >
0.9019):
            diff_list.append(0)
        else:
            diff_list.append(1)
    diff_array=np.asarray(diff_list)
    print(diff_list)
#part of code to determine the distribution of the events with grouping the granularity depending on the
window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        c2 = collections.Counter(newdiff_array[v])[2]
        if c1 > c0 and c1 > c2:
        #if np.count_nonzero(newdiff_array[v] == 1) > np.count_zero(newdiff_array[v]) :
            act_label.append(1)
        elif c2 > c0 and c2 > c1:
            act_label.append(2)
        else:
            act_label.append(0)

# Part of code to determine the distribution of the events in the input data in case of k=4
if KV ==4:
    for j in range(0,len(y_array1D)):
            if x_array1D[j] == 0 and y_array1D[j] == 0 :
                diff_list.append(2)
            elif (y_array1D[j] < 0.4 and y_array1D[j] > 0.378) and (x_array1D[j] < 0.59  or x_array1D[j] >
0.575):
                diff_list.append(0)
            elif (y_array1D[j] < 0.09 and (x_array1D[j] <= 0.55  and x_array1D[j] >= 0.5)) or (y_array1D[j]
< 0.09 and (x_array1D[j] <= 0.6375 and x_array1D[j] >= 0.575)) :
                diff_list.append(1)
            elif (y_array1D[j] > 0.45 and x_array1D[j] > 0.64):
                diff_list.append(3)
```

```
            else:
             diff_list.append(1)
    diff_array=np.asarray(diff_list)
    print(diff_list)
#part of code to determine the distribution of the events with grouping the granularity depending on the
window size
    newdiff_array = diff_array.reshape(-1,ws*1)
    print(newdiff_array)
    for v in range(int(len(norm_df.index)/ws)):
        c0 = collections.Counter(newdiff_array[v])[0]
        c1 = collections.Counter(newdiff_array[v])[1]
        c2 = collections.Counter(newdiff_array[v])[2]
        c3 = collections.Counter(newdiff_array[v])[3]
        if c1 > c0 and c1 > c2 and c1 > c3:
            act_label.append(1)
        elif c2 > c0 and c2 > c1 and c2 > c3:
            act_label.append(2)
        elif c3 > c0 and c3 > c1 and c3 > c2:
            act_label.append(3)
        else:
            act_label.append(0)

#determine the following attributes to get the return values
#labels_ : gives predicted class labels (cluster) for each data point
#cluster_centers_ : Location of the centroids on each cluster. The data point in a cluster will be close to
the centroid of that cluster. As we have two features and four clusters, we should get four locations.
#inertia_ : gives within-cluster sum of squares. This is a total of the within-cluster sum of squares for all c
kmeans = KMeans(n_clusters=KV, init='k-means++',max_iter=300,  n_init=10,
random_state=0).fit(data)
print(kmeans.inertia_)
#print(kmeans.n_iter_)
print(kmeans.cluster_centers_)
centroids = kmeans.cluster_centers_
print(act_label)
print(kmeans.labels_)

#Get each cluster size
print(Counter(kmeans.labels_))

#Visualization of k-means clustering in case the ws is equal to 1 (2D representation)
if ws == 1:
    xdata = data[:,0]
    ydata = data[:,ws]
    sns.scatterplot(data=data, x=xdata, y=ydata, hue=kmeans.labels_, palette="bright")
    plt.xlabel("VTX feature")
    plt.ylabel("VRX1 feature")
    plt.title("Kmeans clustering with ws 1 and normalized data")
```

```
    plt.show()
#Visualization of k-means clustering in case the ws is equal to 2 (3D representation)
elif ws ==2:
    xdata=norm_df["VTX"].to_numpy().reshape(-1,2*1)
    ydata=norm_df["VRX1"].to_numpy().reshape(-1,2*1)
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    label_ws = kmeans.labels_
    df_label_ws = pd.Series(label_ws)
#replicate each row ws times
    df_label_ws_new = pd.Series(np.repeat(df_label_ws.values, 2, axis=0))
    z =df_label_ws_new.to_numpy()
    ax.scatter(xdata, ydata, z, c='r', marker='o')
    ax.set_xlabel('VTX')
    ax.set_ylabel('VRX1')
    ax.set_zlabel('Label')
    plt.title("Kmeans clustering with ws 2 and normalized data")
    plt.show()
else:
#Visualization of k-means clustering with ws higher than 2
# Necessary to reduce the dimensionality even losing information since nparray built has a higher
dimensionality
#Reduce dimensionality for clustering image purpose with TSNE
    if KV==2 or KV==3:
        X_TSNE = TSNE(n_components=KV, learning_rate='auto',init='random',
perplexity=1).fit_transform(x_array,y_array)
        #print(X_embedded)
        sns.scatterplot(data=X_TSNE)
        plt.title('Clustering after TSNE reduction')
        plt.show()
    #Reduce dimensionality for clustering image purpose with PCA
        pca = decomposition.PCA(n_components=KV)
        X_PCA=pca.fit_transform(x_array,y_array)
        sns.scatterplot(data=X_PCA)
        plt.title('Clustering after PCA reduction')
        plt.show()
    elif KV==4:
        #Reduce dimensionality for clustering image purpose with PCA
        pca = decomposition.PCA(n_components=KV)
        X_PCA=pca.fit_transform(x_array,y_array)
        sns.scatterplot(data=X_PCA)
        plt.title('Clustering after PCA reduction')
        plt.show()

#Compute Confusion matrix and metrics in case of K=2
if KV ==2:
    CM= confusion_matrix(act_label, kmeans.labels_, labels=[0,1])
```

```
    print("Confusion matrix: \n", CM)
    plt.matshow(CM)
    plt.title('Confusion matrix of the Kmeans classifier')
    plt.colorbar()
    plt.show()
    #outcome values order
    tn, fp, fn, tp = confusion_matrix(act_label, kmeans.labels_).ravel()
    print("Outcome values: \n", tn, fp, fn, tp)
    print("True negative",tn)
    print("False positive", fp)
    print("False negative", fn)
    print("True Positive", tp)
    #evaluations metrics
    #Accuracy
    acs=accuracy_score(act_label, kmeans.labels_)
    print("Accuracy",acs)
    #Precision
    prs=precision_score(act_label, kmeans.labels_)
    print("Precision",prs)
    #Recall
    rcs=recall_score(act_label, kmeans.labels_)
    print("Recall",rcs)
    #F1 score
    f1s=f1_score(act_label, kmeans.labels_)
    print("f1 score",f1s)
    # classification report for precision, recall f1-score and accuracy
    CM = classification_report(act_label, kmeans.labels_, labels=[0,1])
    print("Classification report: \n", CM)
    mat = confusion_matrix(act_label, kmeans.labels_)
    sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
    plt.title('Confusion matrix of the Kmeans classifier')
    plt.xlabel('true label')
    plt.ylabel('predicted label')
    plt.show()
#Compute Confusion matrix anc metrics in case of K=3
if KV ==3:
    CM= confusion_matrix(act_label, kmeans.labels_, labels=[0,1,2])
    print("Confusion matrix: \n", CM)
    plt.matshow(CM)
    plt.title('Confusion matrix of the Kmeans classifier')
    plt.colorbar()
    plt.show()
# #outcome values order
#evaluations metrics
# #Accuracy
    acs=accuracy_score(act_label, kmeans.labels_)
    print("Accuracy",acs)
```

```
# classification report for precision, recall f1-score and accuracy
   CM = classification_report(act_label, kmeans.labels_, labels=[0,1,2])
   print("Classification report: \n", CM)
   mat = confusion_matrix(act_label, kmeans.labels_)
   sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
   plt.title('Confusion matrix of the Kmeans classifier')
   plt.xlabel('true label')
   plt.ylabel('predicted label')
   plt.show()
#Compute Confusion matrix anc metrics in case of K=4
if KV ==4:
   CM= confusion_matrix(act_label, kmeans.labels_, labels=[0,1,2,3])
   print("Confusion matrix: \n", CM)
   plt.matshow(CM)
   plt.title('Confusion matrix of the Kmeans classifier')
   plt.colorbar()
   plt.show()
# #outcome values order
#evaluations metrics
# #Accuracy
   acs=accuracy_score(act_label, kmeans.labels_)
   print("Accuracy",acs)
# classification report for precision, recall f1-score and accuracy
   CM = classification_report(act_label, kmeans.labels_, labels=[0,1,2,3])
   print("Classification report: \n", CM)
   mat = confusion_matrix(act_label, kmeans.labels_)
   sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
   plt.title('Confusion matrix of the Kmeans classifier')
   plt.xlabel('true label')
   plt.ylabel('predicted label')
   plt.show()

x_array_ws=norm_df["VTX"].to_numpy().reshape(-1,ws*1)
y_array_ws=norm_df["VRX1"].to_numpy().reshape(-1,ws*1)
data_ws = np.concatenate((x_array_ws,y_array_ws),axis =1)
kmeans= KMeans(n_clusters=KV, init='k-means++',max_iter=300,  n_init=10,
random_state=0).fit(data_ws)
label_ws = kmeans.labels_
df_label_ws = pd.Series(label_ws)
#replicate each row ws times
df_label_ws_new = pd.Series(np.repeat(df_label_ws.values, ws, axis=0))
label_ws_updated =df_label_ws_new.to_numpy()
norm_df['labels_ws0'] = label_ws_updated.tolist()
norm_df['labels_ws1'] = label_ws_updated.tolist()
norm_df['labels_ws2'] = label_ws_updated.tolist()
norm_df['labels_ws3'] = label_ws_updated.tolist()
norm_df['labels_ws3'] = norm_df['labels_ws3'].replace(0, np.nan)
```

```
norm_df['labels_ws3'] = norm_df['labels_ws3'].replace(1, np.nan)
norm_df['labels_ws3'] = norm_df['labels_ws3'].replace(2, np.nan)
norm_df['labels_ws2'] = norm_df['labels_ws2'].replace(1, np.nan)
norm_df['labels_ws2'] = norm_df['labels_ws2'].replace(2, np.nan)
norm_df['labels_ws2'] = norm_df['labels_ws2'].replace(3, np.nan)
norm_df['labels_ws1'] = norm_df['labels_ws1'].replace(0, np.nan)
norm_df['labels_ws1'] = norm_df['labels_ws1'].replace(2, np.nan)
norm_df['labels_ws1'] = norm_df['labels_ws1'].replace(3, np.nan)
norm_df['labels_ws0'] = norm_df['labels_ws0'].replace(0, np.nan)
norm_df['labels_ws0'] = norm_df['labels_ws0'].replace(1, np.nan)
norm_df['labels_ws0'] = norm_df['labels_ws0'].replace(3, np.nan)
f, (ax2) = plt.subplots(figsize=(18, 6))
ax2.scatter(norm_df.index, norm_df.labels_ws0, label="Fault No Power", color='red', s=50)
ax2.scatter(norm_df.index, norm_df.labels_ws1, label="Occupancy", color='green', s=10)
ax2.scatter(norm_df.index, norm_df.labels_ws2, label="Clearance", color='purple', s=10)
ax2.scatter(norm_df.index, norm_df.labels_ws3, label="Fault Peak Voltage", color='orange', s=50)
ax2.plot(norm_df.index, norm_df.VTX, label='VTX')
plt.xlim((0,len(norm_df.index)))
plt.title('K means detection')
plt.xlabel('Data points')
plt.ylabel('Fault labelling')
plt.legend()
plt.show()
```