MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/09.22.19.39-TDI

A HYBRID MACHINE LEARNING PROCESS FOR ANOMALOUS BEHAVIOR DETECTION ON SATELLITE TELEMETRY DATA

Marcio Waldir Silva Junior

Master's Dissertation of the Graduate Course in Engineering and Space Technology/Space Systems Engineering and Management, guided by Dr. Walter Abrahão dos Santos, approved in October 04, 2022.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34T/47LSQK2>

INPE
São José dos Campos
2022

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/09.22.19.39-TDI

A HYBRID MACHINE LEARNING PROCESS FOR ANOMALOUS BEHAVIOR DETECTION ON SATELLITE TELEMETRY DATA

Marcio Waldir Silva Junior

Master's Dissertation of the Graduate Course in Engineering and Space Technology/Space Systems Engineering and Management, guided by Dr. Walter Abrahão dos Santos, approved in October 04, 2022.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34T/47LSQK2>

INPE
São José dos Campos
2022

Cataloging in Publication Data

Silva Junior, Marcio Waldir.

Si38h A hybrid machine learning process for anomalous behavior detection on satellite telemetry data / Marcio Waldir Silva Junior. – São José dos Campos : INPE, 2022.

xxii + 115 p. ; (sid.inpe.br/mtc-m21d/2022/09.22.19.39-TDI)

Dissertation (Master in Engineering and Space Technology/Space Systems Engineering and Management) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2022.

Guiding : Dr. Walter Abrahão dos Santos.

1. Machine learning. 2. Satellite. 3. KPCA. 4. DBSCAN. 5. Anomalous behavior. I.Title.

CDU 629.78:004.82

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÕES

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

DEFESA FINAL DE DISSERTAÇÃO MARCIO WALDIR SILVA JUNIOR BANCA N°194/2022, REG. 134392/2019.

No dia 04 de outubro de 2022, as 09h, por teleconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Mestre em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais. O trabalho precisa da incorporação das correções sugeridas pela Banca e revisão final pelo(s) orientador(es).

Título: "A Hybrid Machine Learning Process for Anomalous Behavior Detection on Satellite Telemetry Data".

Membros da Banca:

Maurício Gonçalves Vieira Ferreira - Presidente - INPE
Walter Abrahão dos Santos - Orientador - INPE
Ana Maria Ambrósio - Membro Interno - INPE
András Vörös - Membro Externo - Budapest University of Technology and Economics (BME)

Declaração de aprovação do membro estrangeiro anexo ao processo.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Mauricio Goncalves Vieira Ferreira**, **Coordenador de Rastreio, Controle e Recepção de Satélites**, em 10/10/2022, às 13:37 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Walter Abrahão dos Santos**, **Tecnologista em Ciência e Tecnologia**, em 12/10/2022, às 13:33 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

sei! assinatura eletrônica

Documento assinado eletronicamente por **Ana Maria ambrosio (E)**, **Usuário Externo**, em 17/11/2022, às 10:27 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

“Se alguém já fez isso, meu filho, com certeza você também é capaz de fazer!
Pode ser que não seja tão fácil, mas com certeza você é capaz.”

Marcio Waldir Silva (Meu pai)

Á todos que compartilharam comigo o seu bem mais valioso, **o tempo**. Em especial à minha família.

ACKNOWLEDGEMENTS

Firstly, I would like to thank the Father, without him, nothing would be possible. I also would like to thank my family who were always there for me, holding my ground and pushing me forwards, with highlights to my parents who raised me believing that if I would like, I could achieve great things, that there are no limits for those ones who have power of will and persistence.

For my friends, colleagues and all those who were patience and comprehensive with my absence, thank you very much. I am grateful that you all could understand that if I was not there, it wasn't because I cared less about you, but because I was caring about the fulfillment of my dreams.

To Dr. Walter A. dos Santos for welcoming me to INPE, accepted as student, and trust me such a challenge, my sincere thanks for the opportunity and for all support that you provided to me.

To Drª Ana Maria Ambrosio, for accepting as a student and for all to knowledge kindly shared as all the head ups and bold reviews about my works, not forgetting to mention all the priceless support on reviewing my dissertation, my sincere thanks for all the support you provided to me.

To Drª Denise Rotondi Azevedo for sharing her knowledge, time and experience about her studies made over the CBERS1 and for opening the door that allowed me to continue her studies.

To INPE and all the employees who provided all the necessary infrastructure for this work, especially the postgraduate secretariats that are always available to answer questions.

To those ones who have doubt of my capacity, who tried to demote me from my ideas or who dare to say that I might not be capable of achieving my goals, thank you. Your thoughts and words were also a fuel for my power of will.

Finally, to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the scholarship that made me capable to perform this work.

ABSTRACT

In space missions, telemetry data is a key source towards systems health monitoring, and the lack of this may compromise the mission. Concerning the great number of functional service telemetries, there are some difficulties regarding the telemetry analysis. Some satellites have hundreds, even thousands of telemetry signals, and to an operator analyzing that to infer something about a system is quite laborious. In this scenario, it can be difficult to perform in advance the detection, diagnosis, and prevention of anomalies and failures, decreasing the reliability and availability of space systems. Thus, shortening the system life and service continuity. This research proposes a data-driven approach, composed of a hybrid Machine Learning process for automatically detecting anomalous behavior on satellites via telemetry data. Such approach aims to provide support in the satellites operations when comes to telemetry data analysis. Through statistics, data science processes, and Machine Learning algorithms, the proposed process was capable of identify original anomalies and injected failures in the behavior of the Power Supply subsystem of the CBERS1 satellite.

Keywords: Machine Learning. Satellite. KPCA. DBSCAN. KNN. SVM. Anomalous behavior.

.

UM PROCESSO DE APRENDIZADO DE MÁQUINA HIBRIDO PARA A DETEÇAO DE COMPORTAMENTOS ANOMALOS EM TELEMETRIA DE SATÉLITES

RESUMO

Em missões espaciais, os dados de telemetria são uma fonte fundamental para o monitoramento da integridade dos sistemas, e a falta dela pode comprometer a missão. Com relação ao grande número das telemetrias de serviço funcionais, existem algumas dificuldades em relação à análise da telemetria. Alguns satélites têm centenas, até milhares de telemetrias, e para um operador analisar isso para inferir algo sobre o sistema tende a ser bastante trabalhoso. Nesse cenário, pode ser difícil realizar antecipadamente a detecção, diagnóstico e prevenção de anomalias e falhas, diminuindo a confiabilidade e disponibilidade dos sistemas espaciais. Assim, encurtando a vida útil do sistema e a continuidade do serviço. Este estudo propõe uma abordagem orientada a dados, composta por um processo híbrido de Aprendizado de Máquina para detectar comportamentos anômalos em satélites por meio de dados de telemetria. Tal abordagem tem como por objetivo fornecer suporte nas operações de satélites quando se trata de análise de dados de telemetria. Por meio de estatísticas, processos de ciência de dados e algoritmos de Machine Learning, o processo proposto foi capaz de identificar anomalias originais e falhas injetadas no comportamento do Subsistema de *Power Supply* do satélite CBERS1.

Palavras-chave: Aprendizado de máquina. Satélites. KPCA. DBSCAN. KNN. SVM. Comportamento anômalo

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
BDR	Battery Discharge Regulator
CAST	Chinese Academy of Space Technology
CBERS	China-Brazil Earth Resource Satellites
CBM	Condition-based Monitoring system
CCD	Charge-Coupled Device camera
CCS	Centro de controle de satélites
CSV	Comma-Separated Values format file
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCP	Data Collecting Platform
DGRU	Deep Gated Recurrent Unit (DGRU RNN)
DLSTM	Deep Long Short-Term Memory (DLSTM RNNs)
ED	Euclidian Distance
EDA	Exploratory Data Analysis
EM	Expected Maximization
EPS	Electrical Power Subsystem
GRU	Gated Recurrent Unit (GRU RNN)
INPE	Instituto Nacional de Pesquisas Espaciais
JAXA	Japan Aerospace eXploration Agency
KNN	K-Nearest Neighbor
LADEE	Lunar Atmosphere and Dust Environment Explorer
LSTM	Long Short-Term Memory Recurrent Neural Network (LSTM RNN)
MD	Manhattan Distance
MLP	MultiLayer Perceptron
MPPCACD	Mixture of Probabilistic Principal Component Analysis and Categorical Distribution
NGS	Non-Gaussianity Score
NPP	Nuclear Power Plant

PCA	Principal Component Analysis
RBF	Radial Basis Function
RNN	Recurrent Neural Network
SCD-2	Satélite de Coleta de Dados -2
SDS-4	Small Demonstration Satellite 4
SID	Serviço de Informação e Documentação
SPG	Serviço de Pós-Graduação
SVM	Support Vector Machines
TC	Telecommand
TM	Telemetry
t-SNE	t-distributed Stochastic Neighbor Embedding Function

SUMMARY

Page.

1 INTRODUCTION

1.1 Context and motivation

The space environment is very harsh for spacecraft due to a variety of factors such as direct radiation, great temperature difference, risk of a clash with space debris, and so on. It is practically impossible to completely eliminate the possibility of anomalies or faults, even if we increase the reliability of the system component to the limit. In addition, the space is so distant from the earth that it is extremely difficult to directly inspect and repair a damaged component (IBRAHIM, 2018).

Artificial satellites usually provide important services in communication, remote sensing, scientific experiments, etc. Satellite damage entails not only a financial loss but also the loss of essential and sometimes strategic services. In this scenario, early detection, diagnosis, and prevention of anomalies and failures promote the reliability and availability of space systems, extending service life and long service continuity (AZEVEDO et al., 2012).

In space missions, telemetry is the only source of system status for the ground operations, and the lack of this can compromise the mission since the data set from Telemetry is usually the main source for the identification and prediction of anomalies on an artificial satellite. However, even with functional telemetry, there are some difficulties regarding the analysis of telemetry data. Some satellites have hundreds, even thousands of different telemetry signals, and for operators analyzing the entire mass of data from that telemetry to infer something about the system tends to be laborious.

Telemetry data is received in real-time and analyzed by specialists and operators, constituting the main source of identification and prediction of anomalies in artificial satellites (AZEVEDO et al., 2012). The complexity of these satellites with a big number of subsystems is reflected by the big number of TMs/TCs (WERTZ, 2011). Therefore, a large number of telemetries signals makes adequate telemetry analysis an extremely complex task.

A satellite may be seen logically as a set of integrated subsystems (orbit and attitude control, thermal energy, power supply, structure, payload, a on-board computer, etc.). Each subsystem has a set of sensors (thermistors, switches, battery discharge, etc.) to measure the subsystem's condition and condition of the satellite. These measurements are present in the telemetry data and are the starting point of this work.

Larger satellites made by INPE have a large number of telemetry and remote controls and the usage of a Data Science approach may have a positive impact in the analysis effectiveness. For example, CBERS-4A, which was launched at the end of 2019, has more than one thousands of TMs and TCs, which makes the analysis of failures by Control Center operators more difficult and laborious.

As highlighted by Taburoğlu (2019), supervised learning techniques are no longer used because they need knowledge of expertise. There is limited unsupervised study, but the number of papers is increasing fast, and presenting quite satisfactory results for anomaly detection on systems.

The employment of data driven approaches as data science or data mining in the space systems related has also been studied at INPE. The prospect of multiple launches and increased on demand from orbiting satellites in operation according to the INPE's satellite program, raised the need of improving safety in the planning of routine operations that control the satellites in orbit. Souza (2011) proposed the usage of data mining concepts to analyze the data and predict the satellite operational states, assisting experts in evaluating the performance of the plan. This way improving safety in the planning of operations also ensures the integrity of satellites in orbit.

In this dissertation, the process of detecting data observation presenting unexpected behavior is called anomaly detection. Anomalies have an extended definition though. Every anomalous observation is considered an outlier, however, not all outliers are considered an anomaly on the given problem domain. This assumption is made over the fact that in the spacecraft anomaly detection domain, some outliers in the telemetry data may be caused by noise in the measurement, temporary errors in the data conversion or transmission process (YAIRI et al., 2017), and these cases will be treated as "trivial outliers".

1.2 Objective and research methodology

The objective of the research is to explore a process based on data science approach for space systems service telemetry data analysis. Such analysis aims of providing support to the operation of artificial satellites, being capable of detecting anomalous behavior based on the telemetry data. This study also has the goal of being a backbone for a machine learning process capable of performing not only detection but also, in the future, failure diagnosis.

A complete case study was performed with the China-Brazilian Earth Resource Satellite 1 (CBERS1) telemetry data, obtained by INPE's Satellite Control Center, to experimentally evaluate the proposed process. Another case study was partially performed with the Data Collection Satellite 2 (SCD2) also provided by INPE's Satellite Control Center. The SCD2 data comprises over 4 years of satellite telemetry data, totaling over 24GB of raw data and associated documentation, with 135 telemetries being tracked. However, due to limitations imposed by the available resources for performing the research, the SCD2 case study was only partially perform.

In order to evaluate whether the proposed process is effective from the anomaly detection point of view, first the available data were selected and "filtered" using rules based on the literature. The obtained data set was then divided into two parts. The first part was segmented into training, verification and tuning, being the source of information for problem domain characterization. The process refinement and insights raised in the first part were used as a knowledge base for the following step. The second part was then responsible for training and validating the process again the rest of the data. The data used in the second part contained a visual-identified anomalous behavior that was used as a first validation. A second validation was performed using an injected trend that simulated a failure reported in the literature. The validation criteria used was checking whether the proposed process, fed by input data, was capable of detecting the identified anomalous behavior, providing useful information that could be used as support information for a satellite operator.

The remainder of this documented is structured as follows:

- Chapter 2: Presents the theoretical background Space Mission, Anomaly detection, Data Science topics necessary for the understanding of this dissertation;

- Chapter 3: Related works in the literature are presented and reviewed, as well as anomaly detection approaches to problems that resemble the one approached in this dissertation, and how researchers on the anomaly detection field are handling such problems;

- Chapter 4: Presents the method, the case study planning with the CBERS1 and SCD2 satellites and showcases the mixed machine learning process proposed for enabling anomaly detection on such study cases;

- Chapter 5: Presents a critical analysis of the results, where the proposed approach is better suited and where they excelled or had drawbacks;

- Chapter 6: Presents the conclusions, summarizing the usefulness of the results and future work.

1.3 Contribution and limitations

The contribution of this work is the proposal of a machine learning process that can be not only functionally speaking adequate but also is expected to save money and time for space organizations that operate satellites and need to implement their own telemetry data analysis processes and methods to analyze data. This work is performed exclusively with open-source software, open literature and aims to publish the satellite telemetry data used as a case study for future uses. The dissertation in a certain sense is related to the space segment, addressing the satellite health monitoring through telemetry data analysis. However, it contributes to the ground segment since the proposed process would be implemented on the satellite control center, in order to contribute in a positive way with the telemetry data analysis conducted by the satellite operators.

2 THEORETICAL BACKGROUND

This section presents a common ground knowledge needed for the understanding of this dissertation.

2.1 Space mission

A space system may be seen as a system of systems, where each system is responsible for a different part of the space mission. From the interaction of these systems, an emerging desirable characteristic arose, fulfilling the given space mission purpose. A space system is "made of":

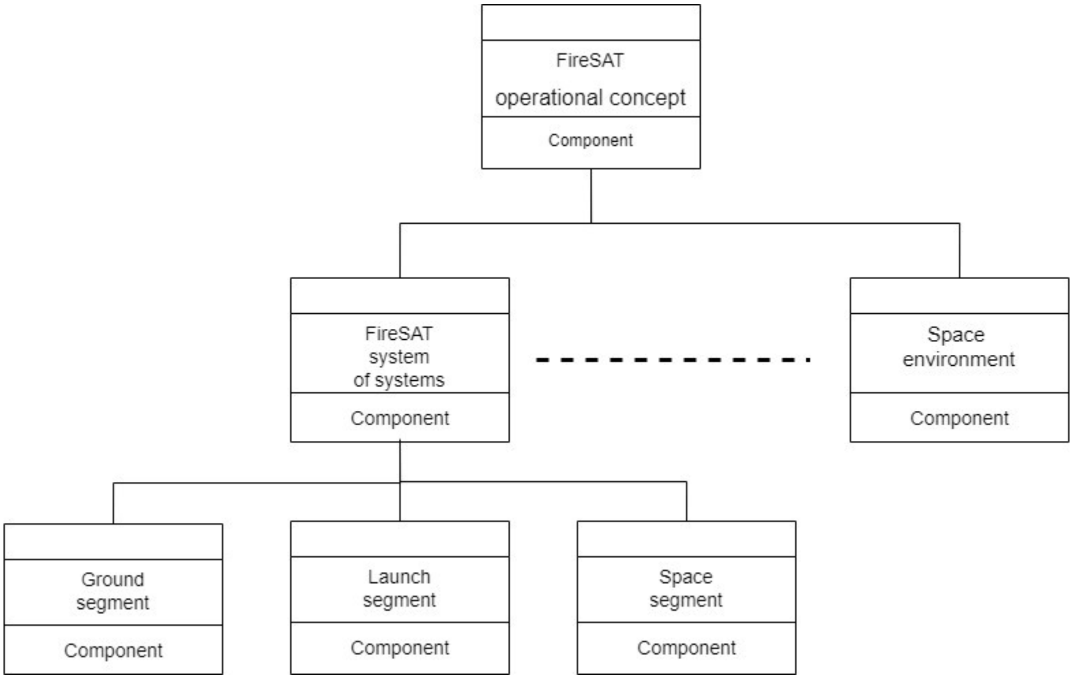
- A ground segment;
- A launching segment, translated into the whole launching mission framework, and;
- A space segment, which in the context of this dissertation is fulfilled by an artificial satellite, as the example depicted in Figure 2.1.

A Space mission can have different purposes, such as Earth monitoring, astronomy observation, communication, military and etc. The presented case study is found within the Earth monitoring category, since it uses telemetry data coming from earth resource monitoring satellites. However, the outcome of this dissertation intends to aggregate to the ground segment since the proposed process may be used on the satellite operations, as highlighted in Section 1.1.

The ground segment, available currently in Brazil, established for the Complete Brazilian Space Mission, can be divided into the mission control center and the tracking and control center, being this last one partitioned into the Satellite Control Center (CCS), and the ground stations from Cuiabá and Alcântara (ORLANDO; KUGA, 2022).

An artificial satellite (also referred to as a satellite) is partitioned into two distinct modules, the service module and the payload module. The service module is composed of everything necessary for determining and maintaining the operational conditions of the satellite. This module holds the on-board equipment, such as the *power supply subsystem*, the ground communication subsystem, the on-board computer, the attitude and control subsystem, and so on. The service module has the aim of keeping the payload healthy, happy and pointed in the right direction. The data related to the service module is referred to as housekeeping data. The payload is composed of all the necessary equipment to accomplish the mission objectives, in broader terms, it is a combination of hardware and software, these being sensors, cameras, etc (WERTZ; LARSON, 1992).

Figure 2.1 - Space mission example.



Source: Adapted from Wertz and Larson (1992).

Satellites telemetry data can be distinguished, in a first level, by their source. The payload telemetry data, which relates to the mission data, in other words, is the reason to be of the satellite. These data can be made up of photos taken for remote sensing (WERTZ; LARSON, 1992).

The housekeeping telemetry data is the data used to assess the current status of the satellite, not just from the satellite's health point of view but also information regarding the system's orbit, attitude, temperature, and other information related to the status and condition of the system. These data most of times consist of information relevant for the operation of the satellite, such as measurements from sensors spread through satellite's equipment or modes of operation of some instrument. Different from the mission data, the housekeeping data is usually continuously transmitted. These data are collected by the satellite’s on-board computer and are sent to the ground stations via the telecommunication sub-system.

Ultimately, the satellite telemetry data is the unique source of information regarding the satellite system situation when it is in orbit. In order to assess whether the satellite's health enables its mission, or if the satellite is performing the task it should relies on the satellite operation process. And the success of the operation of the satellite, relies on this data, and the information retrieved through telemetry data analysis performed by the satellite operators, responsible for monitoring and operating the satellite on the ground.

2.2 Anomaly detection

Anomaly detection techniques perform an important role in many different application domains, such as fraud detection over service providers, fault detection on spacecraft systems, or intrusion detection in computer networks (CHANDOLA et al., 2009). It can be applied to detect performance degradation on mechanical parts of a manufacturing system (PURARJOMANDLANGRUDI et al., 2014), supporting the maintenance of the system. Or used to perform a critical task in a safety-critical environment, detecting abnormal running conditions from an aircraft engine rotation (HODGE; AUSTIN, 2004).

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior (CHANDOLA et al., 2009). These patterns, consist of outliers, which can be defined as an observation, or a subset of them, which appears to be inconsistent with the remainder of that set of data (BARNETT; LEWIS, 1994).

In the literature, anomaly detection is treated by different terminologies by different authors, being called as outlier detection, novelty detection, noise detection, among others. The same can be seen when comes to observations called as outliers, which can be also referred to as exceptions, discordant observations, anomalies, and so on.

Anomaly detection techniques deals with unwanted patterns in the data, and even though it relates to noise removal, these are distinct when comes to each technique's goal. In the context of this work, noise is a phenomenon that is not an object of interest for the analysis and also imposes difficulties on anomalies detection way (CHANDOLA et al., 2009).

Different application domains present different characteristics that tell a lot about how the data could be handled or interpreted, and this aspect shall be taken into consideration when thinking about which anomaly detection technique should be used. Hence, a problem characterization can be driven by the following domain characteristics:

- Nature of the input data
- Anomaly type
- Label availability
- What kind of output

In this dissertation context, the input data is a data set containing a set of observation, or data instances (also called as point, event, sample, pattern among other) (TAN; STEINBACH; KUMAR, 2005). Each observation is described by a number of attributes. These attributes unravel the nature of the input data and may be found as a continuous value coming from a battery temperature sensor, or a binary attribute stating a state of a switch, or a categorical data representing the current mode of operation of a given equipment. Attributes are also referred as dimensions or features. Other relevant aspect of the nature of the input data are whether the data set is univariate or multivariate, the last one meaning that each data instance has multiple attributes. Not surprisingly, the type of data determines which tools and techniques can be used to analyze the data. (TAN; STEINBACH; KUMAR, 2005), i.e., for nearest-neighbor-based techniques, the nature of attributes would determine the distance measure to be used (CHANDOLA et al, 2009).

The type of anomaly, given a problem domain, relates to definition of what is the observed behavior which is considered as not conforming to the considered normal behavior, how this anomalous behavior looks like against the rest of the data set. In this matter, the type of anomaly can be divided into three categories (CHANDOLA et al, 2009):

- Point anomalies, which can be seen as an individual data observation laying outside of the regions of the considered normal behavior;
- Contextual anomalies, happens when an observation is considered to be anomalous in a given context, but not otherwise. The observations are classified against two attributes, a context attribute, and a behavioral attribute. This type of anomaly is determined using the value obtained from the behavioral attribute within a given context (defined by the context attribute);
- Collective anomalies, refers to the type of anomaly that is identified only when a collection of related observations is anomalous against the rest of the entire data set. An individual observation within a collective anomaly may not be considered as an anomaly by themselves, but the occurrence of these observations together as a collection is anomalous.

The problem of anomalous behavior detection is divided in three fundamental different approaches (HODGE; AUSTIN, 2004), that differs in respect to the availability of labels on the data set that is going to be used as input. Data sets containing labeled observation for anomaly as well as normal classes, are approached by techniques considered as "Supervised learning" techniques i.e., Nearest Neighbors algorithms or Support Vector Machines (SVM). However, usually these data sets present an imbalanced class distribution.

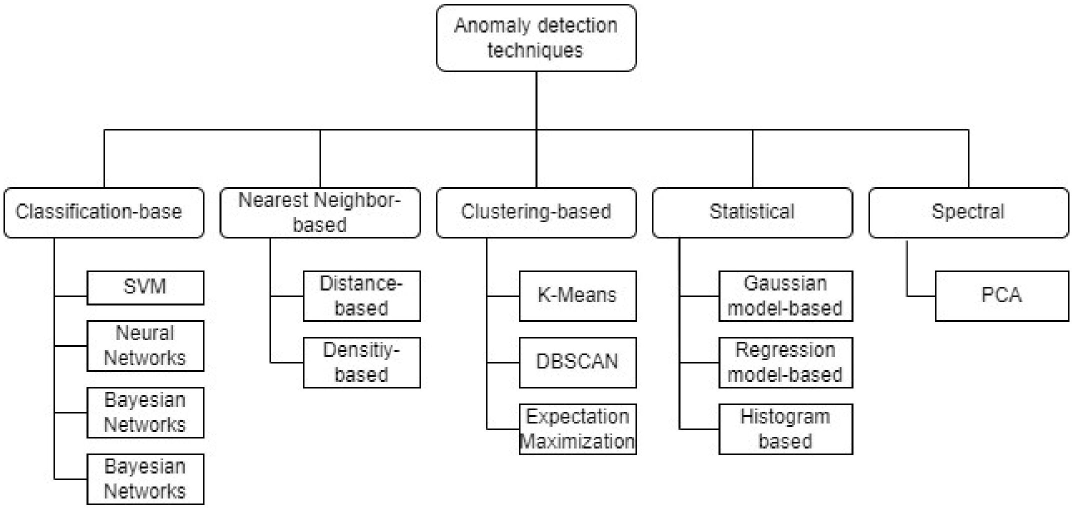
Semi supervised techniques are applied when only normal behavior classes are available as observation labels. In spacecrafts, an anomaly scenario would signify an accident which is not easy to model, therefore, building a model for the normal behavior class and consider anything else as anomalous is more typical (FUJIMAKI et.al, 2005).

When a data set presents no label at all, an unsupervised approach has to be implemented. Due the fact that such techniques do not rely on labels availability, they are widely applicable. In order to unsupervised techniques be capable of generating a model, an assumption that the frequency rate of normal observations and anomalous observations are so disparate, being the normal class far beyond more frequent, is made. Based in this assumption regarding the normal class frequency in a data set, semi supervised techniques can be also used for unsupervised anomaly detection, requiring just some adaptation (CHANDOLA et al., 2019). Two examples of unsupervised techniques are, K-Means algorithm, used for clustering, and Principal Component Analysis (PCA), performing dimensionality reduction.

Anomaly detection techniques, typically, provide outputs of two different types, scores and labels. When a technique provides a value, or index, which indicates how much of an anomaly was that observation considered, the output type is scores. On the other hand, when the approach provides an output that is an assignment to a specific class, then it is said that its outputs generate labels for the observations.

In a nutshell, the anomaly detection challenge comes from the fact that the problem domain, or application domain, has different aspects, such the nature of the input data, the notion of anomaly, the challenges associated with detecting those anomalies. Furthermore, to define which are the existing techniques of anomaly detection that can be used to the given problem domain, all these aspects shall be taken into consideration (CHANDOLA et al., 2019). Some examples of types of algorithms based on the anomaly detection technique are shown in Figure 2.2.

Figure 2.2 - Different anomaly detection techniques examples.



Source: Author's production.

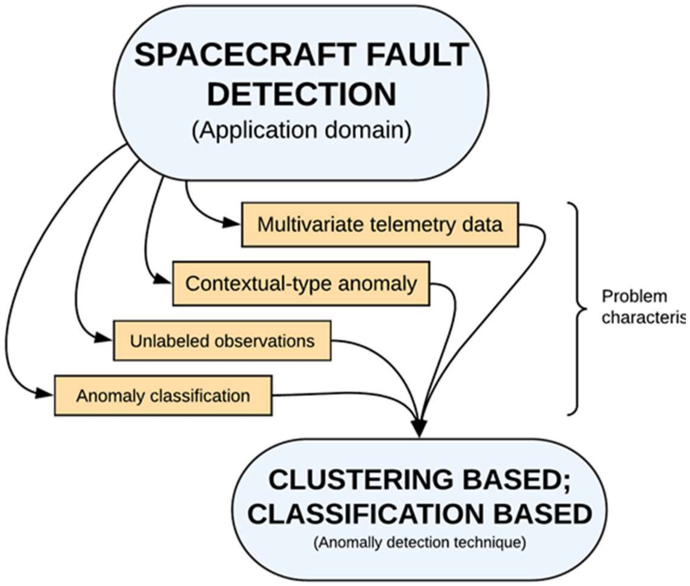
2.2.1 Space systems anomaly detection

Being capable of detecting failure and tracing diagnosis on a space system equipment has been important and necessary since the first space flight in the 50s and have gained even more importance in the late years with the beginning of manned flights. Independently if the system under observation is a spacecraft or a launcher vehicle or an artificial satellite, the fact that these systems are far beyond the reach to be physically probed and assessed. To be capable of failure detection and diagnosis is a fundamental key to keep a system alive and functional, from the mission point of view. In artificial satellites, the failure detection capability has to be on time and accurate in order to reduce the aggravation of its health. This health monitoring approach it is what determine whether the satellite can safety, reliably operate through a long-life.

In-space applications, as many other ones, impose challenges to perform autonomous anomaly detection which can be summarized as the problem domain characterization (AZEVEDO et al., 2012). Furthermore, in artificial satellites the availability of the data and its characteristics can impose even more difficulties to narrow down to an appropriate approach. To identify and apply the most appropriated techniques for automatic detection of anomalies in artificial satellite telemetry data, the problem domain should be characterized in regard to some key aspects (CHANDOLA et al., 2009), which are cited below and depicted in the Figure 2.3:

- Data type: Generally, telemetry data from artificial satellites carry out information regarding the different equipment and sensors on board, due that every time observation can present several different attributes, therefore, these data are multivariate, presenting categorical and continuous for the different attributes it contains.
- Anomaly type: artificial satellites operate in different scenario which present different circumstances, i.e., operating while hidden in the shadow of the earth, or operating having full sight to the sun light. It means that the telemetry data carries context attributes embedded. This way, point anomalies or collective anomalies have to be handled as a contextual anomaly detection problem.
- Label availability: Not having available label of the input dataset is the general situation in this space application. In this case, the decision on the anomaly detection approach to be taken relies on time, cost and expert knowledge availability (i.e., supervised, semi-supervised or unsupervised.
- Expected outcome: tells what kind of result would be more adequate for the given domain. The outcome can be classificatory, generating labels to the data observation i.e., instances anomalous or normal. The outcome can generate index values which will give a probabilistic value to the observations that will indicate how much anomalous an observation can be.

Figure 2.3 – Key aspects for Spacecraft fault detection.


SPACECRAFT FAULT DETECTION
(Application domain)
Multivariate telemetry data
Contextual-type anomaly
Unlabeled observations
Anomaly classification
Problem characteris
CLUSTERING BASED; CLASSIFICATION BASED
(Anomally detection technique)

Source: Author's production.

When comes to space applications anomaly detection, the limit-checking approach has been the most basic and common technique of detecting anomalies in spacecraft systems for a long time (FUJIMAKI et al., 2005). Basically, it monitors important attributes in the telemetry data and checks whether the value is within the pre-defined upper and lower limits for the given attribute, or measurements. Though the limit-checking has an advantage that it is simple enough to be applied to any types of spacecrafts, it can suffer from the problem of false alarms.

Another approach used for such purposes is the knowledge-based, also known as expert systems. In these systems, human expert knowledge about the system, i.e., knowledge from satellite operators, is used to create a set of statements that together delimitate the boundaries of the satellite expected behavior. Such system is powerful and flexible. However, it suffers from the bottleneck of knowledge acquisition, which is the difficulty in achieve an accurate and complete knowledge base.

According to the literature, perform anomaly detection for health monitoring in space application can also be achieved through different ways. In order to detect an anomaly and predict a problem, a model of the satellite can be established based on the system design, also known as model-based approach, where the satellite system design is used to model the expected behaviors of the system. This approach can be considered a more sophisticated one when in contrast to the limit-checking method.

The above approaches of limit-checking, knowledge-based, and model-based, all share the characteristic of being very human experts dependent, what can be seen as a drawback in some situations. On the other hand, a model of the system behavior can also be obtained from the actual telemetry data coming from the in-orbit satellite, through data mining. Due the variety of elements in the satellites context that can cause performance degradation on the satellite, the changing in its behavior through the time in-orbit, especially for long-life systems is expected, and one significant way of tackling this effect is anomaly detection driven by telemetry data (YANG et al, 2013).

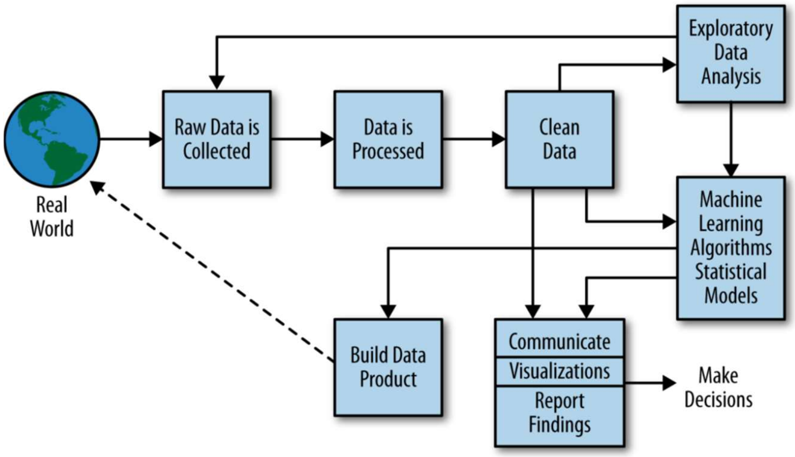
According to Fujimaki et al. (2005), a reasonable approach to this problem is the application of data mining and machine learning techniques to the spacecraft telemetry data. In this approach can be see the usage of different techniques such as classification-based, nearest neighbor-based, clustering, statistical, and mixed process techniques which are discussed in Section 3.

2.3 Data science

According to IBM (2022), Data Science is the combination of different areas of the knowledge, as math, statistics programming, analytics, artificial intelligence (AI), and machine learning, aggregated to a domain know-how to unravel under covered insights.

The data science process adopted in this dissertation is depicted in Figure 2.4. The steps of interest within this process are Clean Data, Exploratory Data Analysis (EDA), Machine Learning Algorithms Statistical Models, and the Communicate/Visualizations/Report Findings.

Figure 2.4 - The data science process according to Schutt and O'Neil.



Source: Schutt and O’Neil (2013).

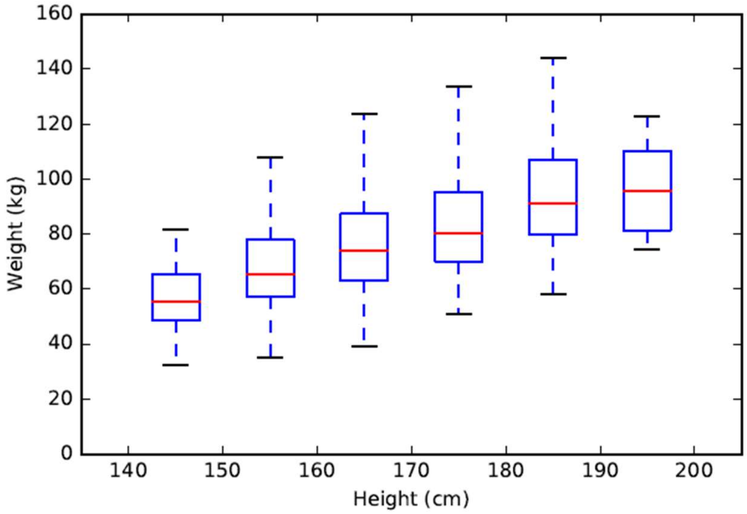
Briefly, Clean Data in this context means make the data tidy, in other words, apply a standard way of mapping the meaning of a dataset to its structure, obtaining at the end a data set which has each variable as a column, each observation as a row, and every cell is a single value (WICKHAM, 2014). Once the data set is clean, a data analysis shall be performed to summarize characteristics of the data, using statistical numbers and graphs, process also known as Exploratory Data Analysis

2.3.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a detective work, being this numerical, counting, or graphical. EDA, can never be the whole story, however, nothing else can serve as the foundation stone (TUKEY, 1977). Being the first step towards building a model, it is through EDA that an analyst can unraveled aspects of the data have insights and build intuitions. In EDA, there is no hypothesis and there is no model. The “exploratory” aspect means that, as the analyst systematically goes through the data, plotting distributions (box plots), transforming variables, analyzing pairwise relationship between these and generating summary statistics. The understanding of the problem ones is being solved, or might solve, is changing as ones go (SCHUTT; O’NEIL, 2013). At the end of the process, Parametric Statistical Modeling methods can be achieved for outlier removal or detection.

Chandola et al. (2009) describe that within the set statistical anomaly detection techniques, parametric techniques such as the box plot rule, makes usage of box plots (box-and-whisker plots) to identify anomalous data points in a given data sample. Such graphical representation depicts the data using summary attributes such as quartiles, smallest and largest observations representations, as depicted in the Figure 2.5. In the box plot, the box delineates the range of values from the quartiles Q1 and Q3 (25% and 75%) and is cut at the median (50th percentile). Typically, whiskers are added to show the range of the highest and lowest value (min and max).

Figure 2.5 - Box and whisker plots showing the quartiles of a distribution.



Box-and-whisker plot of weight as a function of height in a population sample. The median weight increases with height, but not the maximum, because fewer points in the tallest bucket reduces the chance for an outlier maximum value.
Source: Skiena (2017).

A method to create a model for outliers detection, having the box plots information as input, is making usage of the Tukey Fences (TUKEY, 1977). Here the Inter Quartile Range (IQR), associated with a "step", named as beta

(β), can be used to define upper and lower limits to data values when comes to what is considered to be a normal or expected value for a given feature. The IQR is obtained as $IQR = Q3 - Q1$, and the upper and lower limits, or Tukey Fences, can be obtained as the Equations (2.1)(2.2).

$$LoweLimit = Q1 - \ \beta * IQR \tag{2.1}$$

$$UpperLimit = Q3 + \ \beta * IQR \tag{2.2}$$

With this, a data point which lies over the limits, upper or lower, will be treated as an anomaly (SOLBERG; LAHTI, 2005). Modifications of such technique can also be found in the literature where instead of quartile representations, the study presented an approach using percentiles representations (YAIRI et al., 2017), in order to create a rule to preprocess the data before using it as input for a machine learning algorithm.

2.3.2 Machine learning

In Schutt and O'Neil (2013), machine learning algorithms are part of the data science process. The machine learning step is a key point in the process of obtaining a means of solving a given classification, prediction or basic description problem, as shown in Figure 2.4.

Machine learning, broadly defined, is a field of Data science, where computational learning techniques, driven by data, can improve performance of a given process or make accurate predictions combining computer science, statistics, probability and optimization concepts. The usage covers a broad spectrum of application, like text classification, computer vision, computational biology, until applications into the anomaly detection applications set (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018).

These learning techniques are implemented in form of algorithms capable of learning from past information in electronic data form. Machine learning algorithms, usually, provide as an output a model with which one can perform prediction over an input data. However, such model shall be trained with training data set. After the training phase, a machine learning model can be verified, or tested, using a testing data set, with which the model will make predictions upon, based on the training data set. Machine learning essentially creates analytics models from past data (HURWITZ; KIRSCH, 2018).

The decision regarding which machine learning algorithm can be use relies on which kind of task we have to perform, to solve the problem that we have, in other words, it is strongly connected to the problem domain characterization. And, when comes to the problem of detect anomalous behavior, most of the times, the applicable algorithms regarding the technique, are Classification-based, Nearest Neighbor-based, Clustering-based, Statistical, or Spectral, as depict in the Figure 2.2, and according to Mohri, Rostamizadeh and Talwalkar (2018). These techniques can be used to perform distinct standard machine learning task such as classification, regression, clustering, dimensionality reduction or manifold learning, and. ranking.

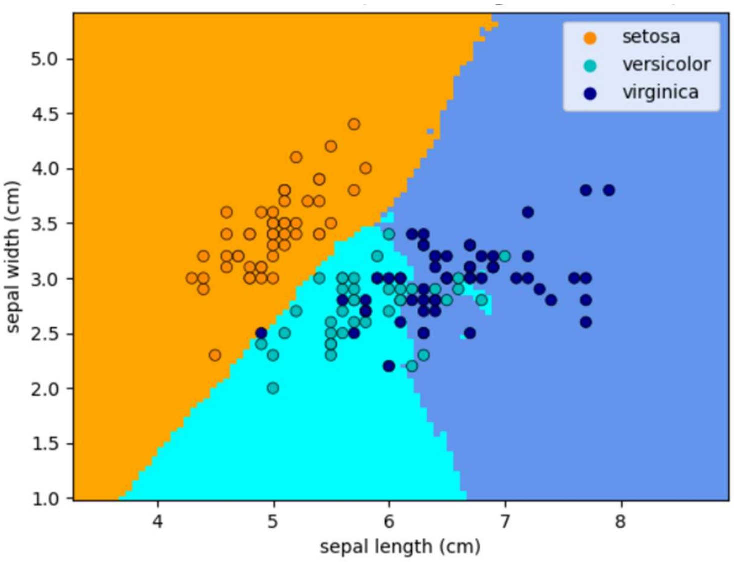
2.3.2.1 Classification

Classification is the problem of, given an input observation, assigning the right label or value to it. The classification model, or target function, is responsible for mapping each observation x to one of the predefined labels y (TAN; STEINBACH; KUMAR, 2005). The classifier (a.k.a. classification model) is obtained through a learning phase using a training data set. The obtained classifier is then verified using a testing data set. In anomaly detection techniques based on classification, the approach is to assume that a classifier that can distinguish between normal and anomalous classes can be learned in the given feature space (CHANDOLA et. al, 2009).

Classification problems, in Figure 2.6, can be approached by different techniques based in different algorithms as depict in Figure 2.2, and even though Classification and Nearest Neighbors techniques are placed in different groups, a Nearest Neighbor-based technique such as the k-Nearest Neighbors (kNN) algorithm, can be used for a classification task (TAN; STEINBACH; KUMAR, 2005).

The k-Nearest-Neighbors algorithm is feasible when it is assumed that similarities in the feature space imply into similarities in the label space. The similarity, or closeness of a given point *x* to a possible neighbor *w*, in the kNN algorithm, is obtained through a distance metric.

Figure 2.6 - 3-Class Nearest Neighbors classification with k equals to 15.



Fisher's Iris data set sample classification when comes to sepal length and width using nearest neighbors approach

Source: Scikit-Learn (2022).

Smaller the distance between the points more similar they are and the more likely the label from *w* will be assigned to *x*. However, the distance metric by itself is not enough since other neighbors with different attributes can interfere in the classification task. To tackle this, the number of neighbor points that shall be assessed before assigning a label to an *x* point shall be defined, this value is the *k* (SCHUTT; O'NEIL, 2013). How labeled points are arranged allied with the number of k, shapes the feature space into different regions, where all the points in a given region will have assigned to then the same label. The basic k-Nearest neighbor classification algorithm is the following:

a) Being *k* the number of nearest neighbors, *d* a test sample, and *D* the training data set;

b) Compute the distance between *d* and every sample in D;

c) Choose the *k* samples in *D* that are nearest to *d*; denote the set by points P (belonging to *D);*

d) Assign to *d* the most frequent class (or the majority class).

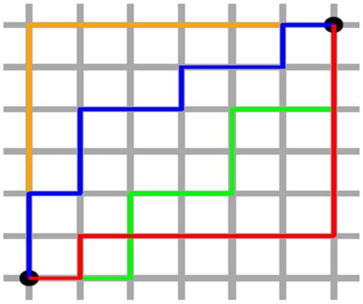
There are different ways to assign the proper value of *k.* According to Skiena (2017), the correct method for such is to assign a fraction of labeled training samples to perform an evaluation, and then perform experiments with different values of the parameter *k,* in order to assess which values achieves the best classification performance. These evaluation values can then be thrown back into the training/target set, once *k* has been selected.

Manhattan and Euclidean are very commonly distance metrics used for similarity calculation (BROWNLEE, 2020). The Manhattan distance (MD), known as City block distance, thought by Hermann Minkowski in 19th-century Germany, is a distance that represents the sum of the absolute differences between two real-valued vectors, as shown in Equation (2.3).

$$\mathrm{MD(x,y)} = \sum_{i=1}^{n} |x_i - y_i| \tag{2.3}$$

As depict in Figure 2.7, typically there are many possible shortest paths between two points. There is no possibility of taking advantage of a diagonal short cut, this way the distance between two points is then the sum of x-dimension differences and the y-dimension differences. Manhattan distance is the total sum of the deviations between the dimensions (SKIENA, 2017).

Figure 2.7 - The different pathways between two points in a Manhattan distance-fashion.

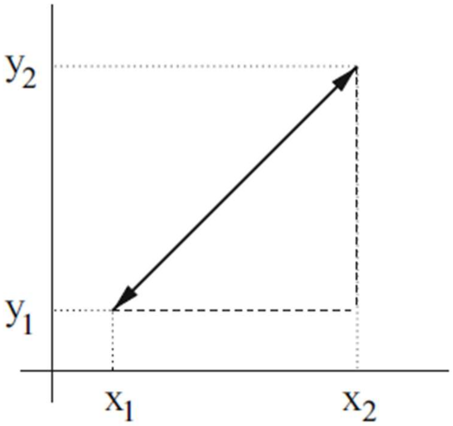


The image to have in mind is that of a taxi having to travel the city streets of Manhattan, which is laid out in a grid-like fashion
Source: Skiena (2017).

The Euclidean Distance (ED) calculates the geometrical distance that connects two data points, *x = (x1, x2)* and *y = (y1, y2),* as depict in Figure 2.8. Considered as the most popular distance metric, ED is based on the Pythagorean theorem, and it has a generalized formula for a domain with n continuous attributes, where the distance is defined in Equation (2.4). This distance represents the root of the sum of the squared difference between the opposite values in vectors.

$$\mathrm{ED}(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.4}$$

Figure 2.8 - The Euclidean distance between two points in a two-dimensional space.



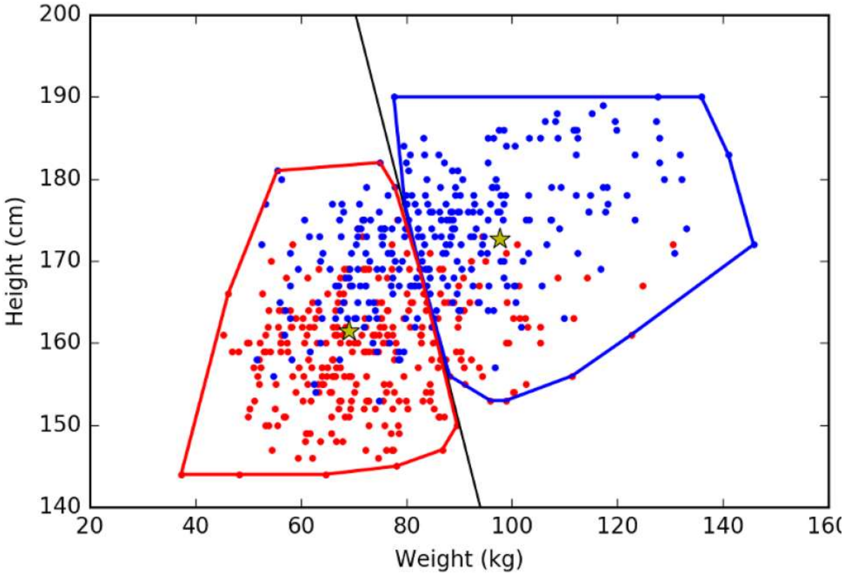
Source: Kubat (2017).

As concluded by Alfeilat et al. (2019), the kNN algorithm classifiers have their accuracy, precision, and recall, strongly dependent on the used distance metric, and as shown in the same study. There is no optimal distance metric option which fits all kinds of datasets, in other words, it the distance metric applied depends on the characteristics of the problem domain.

2.3.2.2 Clustering

Clustering is a machine learning task to solve problems which requires an unsupervised learning approach. Clustering is often used to analyze very large data set, partitioning a data set into homogeneous groups of data points (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018), which are considered to belong to a same group due some similarity they present. Cluster analysis can be understood as the art of finding groups (clusters) on the data (KAUFMAN; ROUSSEEUW, 1990). Cluster analysis separate the data into clusters that communicate, or depict, some aspect of the natural structure of the data that at first hand would not be so obvious. This approach plays an important role in domains like pattern recognition, biology, social sciences, anomaly detection, and so on (TAN; STEINBACH; KUMAR, 2005). An outcome of a clustering task is shown in the Figure 2.9 where two groups are depicted.

Figure 2.9 - Outcome of clustering over a data set of persons characteristics.

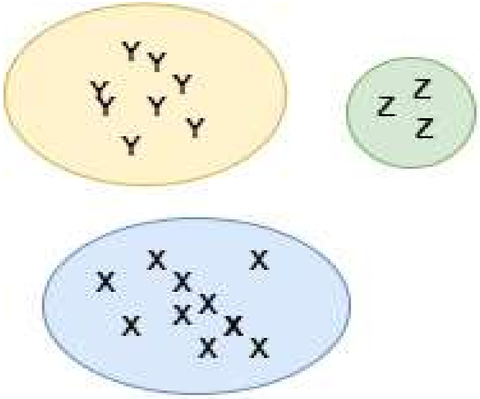


This data set contain measurements of Height and Weight from people and a clustering task was conducted using 2-means clustering through the K-Means algorithm
Source: Skiena (2017).

The clustering process can be divided into many algorithm types as overlapping, exclusive, fuzzy, complete, partial, and the two most commonly used (KAUFMAN; ROUSSEEUW, 1990; SKIENA, 2017), partitioning and hierarchical.

Partitional clustering, or unnested clustering, divides a data-set *D,* into *k* non-overlapping clusters, in a way that each data point belongs to only one cluster as depicted in Figure 2.10. In this kind of algorithms, the input parameter *k* is very important, therefore, some domain knowledge is required, which may not be available (ESTER et al., 1996). An example is the K-Means algorithm.

Figure 2.10 - Partitional-type clustering algorithm outcome with k value of 3.

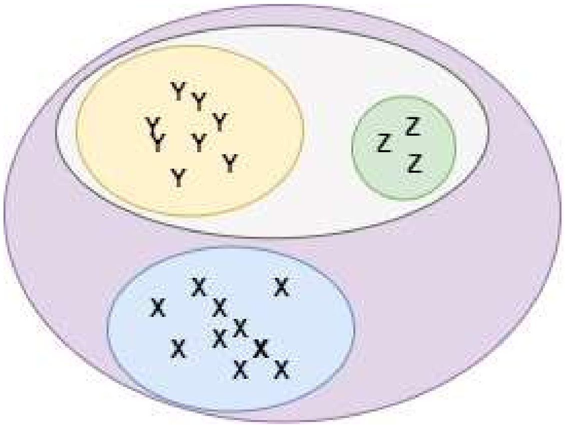


Source: Author's production.

Hierarchical clustering, or nested clustering, differently from partitional algorithms, does not need an *k* value as an input parameter. This defines a set of nested clusters organized as a tree (or a dendrogram), where apart from the leafs, each node is a cluster, each cluster is a union of its subclusters, and the tree root is a cluster which contains all the objects in the given data set, as depicted in the Figure 2.11.

.

Figure 2.11 - Hierarchical clustering algorithm output.



Source: Author's production.

The hierarchical clustering process in this case can also be seen working as a bottom-up clustering, or agglomerative clustering where at the beginning each data point is a cluster, then the points started to be merged and becoming a new cluster. The stop condition is when all data points found aggregated into one single cluster. The other way around, known as divisive clustering, starts with a cluster containing all the data points, which is divided into smaller cluster, till the point a cluster contains only one data point, or a singleton cluster (TAN; STEINBACH; KUMAR, 2005; GLEN, 2022). A hierarchical clustering can be seen as a sequence of partitional clustering.

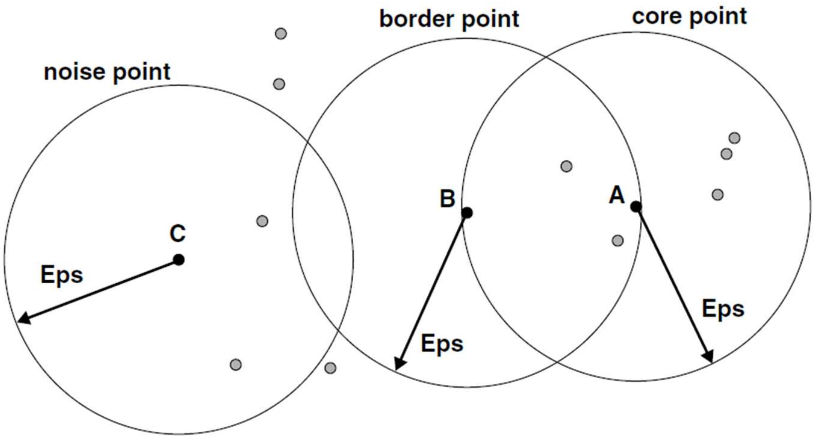
Another important aspect regarding the clustering task is the type of cluster, which can be defined in several different ways, and among then the following types are more relevant for this dissertation:

- Well-separated cluster, where each data point within the cluster is more similar to every other in the cluster than to any data point out of the cluster. The distance between any two points in different clusters is larger than the distance between any two points within a cluster (TAN; STEINBACH; KUMAR, 2005).
- Prototype-based, sometimes referred as center-based clusters are formed by a group of objects which are more similar to the prototype that defines the cluster than to the prototype of any other cluster, i.e., for data with continuous attributes, the prototype of a cluster is often a centroid.
- Density-based clusters are formed because there is a considerable higher density of point within the cluster than outside it (ESTER et al., 1996). An example is the Density-based spatial clustering of applications with noise (DBSCAN). Such algorithm is capable of work with data with noise also because the density within the areas of noise is lower than the density in any of the clusters.

Among a variety of clustering techniques available, two simple important techniques are the K-means and the DBSCAN. The K-means is a partitional, prototype-based, clustering technique that attempts to find a k-number of clusters (user input) represented by their centroids. The DBSCAN is, a density-based, partitional clustering technique, in which the number of clusters is automatically determined by the algorithm (TAN; STEINBACH; KUMAR, 2005).

Within a DBSCAN cluster, data points can be classified in two different ways, those inside of the cluster, therefore named core points, and those ones on the borders of the cluster, name border points. Those points, found in a sparsely occupied region, out of a cluster, are considered as noise points. An example is depicted in Figure 2.12.

Figure 2.12 - DBSCAN clusters.



Source: Tan, Steinbach and Kumar (2005).

The DBSCAN algorithm requires two inputs from the user to be capable of delineate the clusters. The first one is an Epsilon (Eps) value used to determine the Eps-neighborhood value, which defines the size of a given point neighborhood $N_{Eps}(p)$. A point p neighborhood is defined by $N_{Eps}(p) = \{q \in D | \; dist(p, q) \leq Eps\}$. In simple terms, Eps specifies a radius size around a point under assessment. The second input parameter, called minimum number of points (MinPts), is a value that together with the Eps-neighborhood of a point value, can determine whether a point within a cluster is a core point or a border point. A data point p within a cluster is a core point. When within is neighborhood there are a number of points equal or higher to the value define by the MinPts value, also see as $| N_{Eps}(p)| \geq MinPts$. However, when a data point within the cluster cannot fulfill the core point condition, but fall within the core point neighborhood, $q \in N_{Eps}(p)$, then the data point q is considered as a border point (TAN; STEINBACH; KUMAR, 2005).

Formally, according to Ester et al. (1996), given a data set *D,* taking into account the Eps and MinPts input values, a cluster *C* is a non-empty subset of *D* satisfying the following conditions:

- $\forall\, p, q :$ if $p \in C$ and q is density $-$ reachable from p then $q \in C$.
- $\forall\, p, q \in C : p$ is density $-$ connected to q.

While noise can be defined as a set of points in data set *D* not belonging to any cluster *C* of a set of clusters*,* i.e., $noise = \{p \in D | \forall\, i{:}\, p \notin C_i\}$.

A point *p* is density-reachable from a point *q* with regards to. Eps and MinPts if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} , is an element of the $N_{Eps}(p_i)$, and p_i is a core point (p_{i+1} is directly density-reachable from p_i). A point *p* is density-connected to a point *q* with regards to Eps and MinPts if there is a point *o* such that both, *p* and *q* are density-reachable from *o* with regards to Eps and MinPts (ESTER et al, 1996).

Given the definitions of a cluster, a core point, a border points, and noise point the DBSCAN algorithm can be roughly defined as follow:

1. Labeling of all data point according to the core, border or noise point;
2. Eliminate noise points;
3. Put an edge between all core points that are within Eps of each other;
4. Make each group of connected core points into a separate cluster;
5. Assign each border point to one of the clusters of its associated core.

DBSCAN is a clustering technique that can find many clusters that cannot be found using K-means, however, DBSCAN present drawbacks when comes to widely varying cluster-densities and high-dimensional data set due difficulties in defining the density for such data (TAN; STEINBACH; KUMAR, 2005).

2.3.2.3 Dimensionality reduction

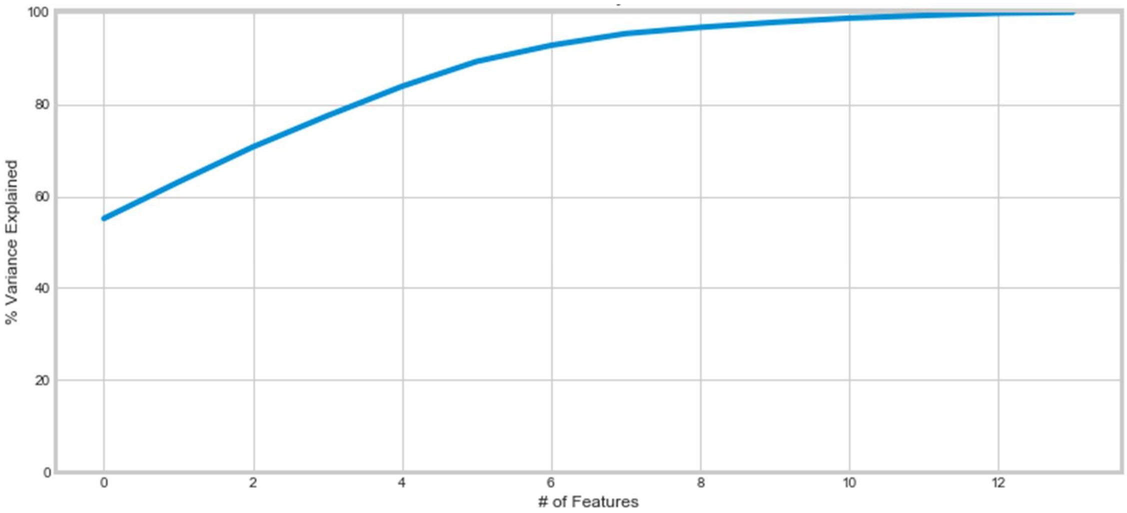
According to Mohri, Rostamizadeh and Talwalkar (2018), dimensionality reduction, in a data set containing a large number of features, is the act of transforming its initial representation into a lower-dimensional representation, while preserving some of its properties. This technique helps in the removal of data that might not be useful for the analysis, as redundant data, outliers, and other non-useful data. Reduce the dimension of data set also can be used as a support for analysts to visualize the data (HURWITZ; KIRSCH, 2018). The main arguments to use dimensionality reduction techniques are computational in order to compress data as a preprocessing step, visualization in order to make the exploratory data analysis possible in two or three dimensions, and finally for feature extraction seeking a smaller and more useful set of features.

Two main groups of algorithms for dimensionality reduction can be highlighted, the first one is driven by linear algebra, comprising the Principal Component Analysis (PCA) algorithm, the Singular Value Decomposition and the Non-Negative Matrix Factorization. The second group is the one from the Manifold learning methods, comprising the algorithms, t-distributed Stochastic Neighbor Embedding (t-SNE), and Kernel-Principal Component Analysis (KPCA) (BROWNLEE, 2020).

The objective of the PCA algorithm is to obtain a new set of dimensions (features). The new set is the most meaningful in representing the variability of the data. The output of the algorithm provides components (dimensions or features), where the first component usually has the responsibility for capturing as much of the variability of the data as possible. The second component is orthogonal to the first and subject to that constraint, captures as much of the remaining variability as possible and so on (TAN; STEINBACH; KUMAR, 2005). This is depicted in Figure 2.13, where a curve of number of components used versus percentage of absolute variance captured can be seen.

In order to obtain the components that represent the data the most, the PCA algorithm has to find the eigenvectors which have the bigger associated eigenvalues, because these will be the principal components.

Figure 2.13 - Number of components vs. amount of data variance explained.



The first component (feature) is responsible for explaining more than 60% of the variance of the data. As more components are added, more of the variance is explained, however, the aggregated value per each new added component starts to be irrelevant from the fifth component on.
Source: Author's production.

In a multiple dimensions data set (many features), the most important *k* features from the original *m* features are going to be represented by the eigenvectors associated with the *k* largest eigenvalues. The eigenvectors and eigenvalues can be obtained through the Equation (2.5) :

$$Se_i = \lambda_i e_i \tag{2.5}$$

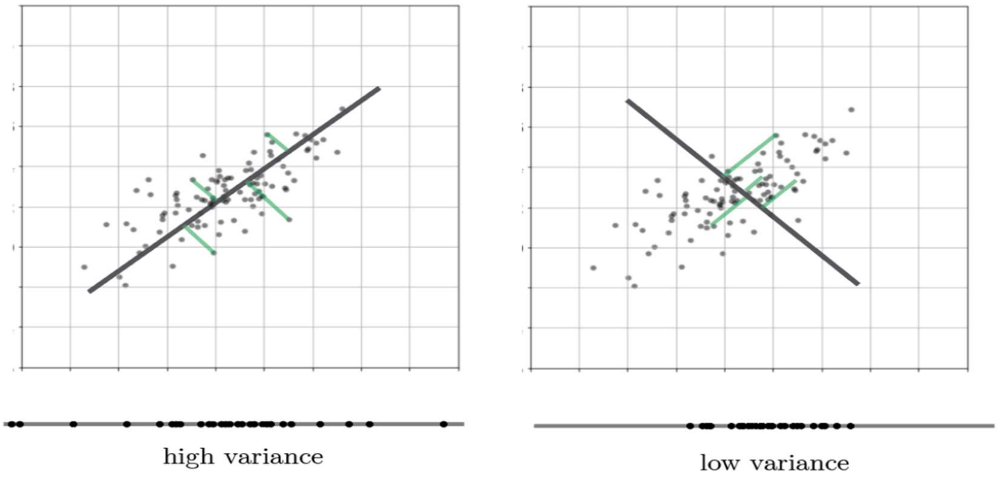
Where *S* is a matrix, *e* is an eigenvector, and λ denotes the eigenvalue. For each eigenvalue in *S,* $(\lambda_1, \lambda_2, \dots, \lambda_p)$ correspond to the variance associated to each principal component $(PC_1, PC_2, PC_3, \dots PC_P)$ (RIBEIRO, 2022). The used matrix is a covariance matrix, which represent the covariance measurement between the variables (features in the data), given on its principal diagonal. The Equation (2.6) demonstrates how to calculate the covariance between two variables (two dimension) for all the observations on a matrix that is representing a data set of two dimensions.

$$\boldsymbol{C_{x,y} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})} \tag{2.6}$$

$C_{x,y}$	covariance of features *x* and *y*
x_i	specific value from feature *x*
y_i	specific value from feature *y*
$\bar{x}$	mean over all observations of feature *x*
$\bar{y}$	mean over all observations of feature *y*
n	number of observations

The left side of Figure 2.14 shows what is achieved when the eigenvector whose eigenvalues is the biggest is found, for a data set input of two dimensions. There, a component traced through the data points can represent as much as possible of the variance of the data points, it is the vector through the dataset that maximizes the variance. On the right side, the opposite happens with an orthogonal component not being capable of represent as much as variance as the left example depicts.

Figure 2.14 - Example of components and the variance they can represent.



Source: Dhalla (2022).

In simplified terms, the PCA algorithm can be seen as an unsupervised learning technique and its process of obtaining principal components can be summarized as follows, given a dataset *D* composed of *n dimensions:*

- Calculate the mean of the *n-dimensions;*
- Compute the covariance matrix of all *n*-dimensions of *D;*
- From the covariance matrix, obtain the eigenvectors and the associated eigenvalues;
- Choose k eigenvectors with the largest eigenvalues to form a d X k dimensional matrix W;
- Use the W-transposed matrix to transform the original matrix onto the new subspace, or:

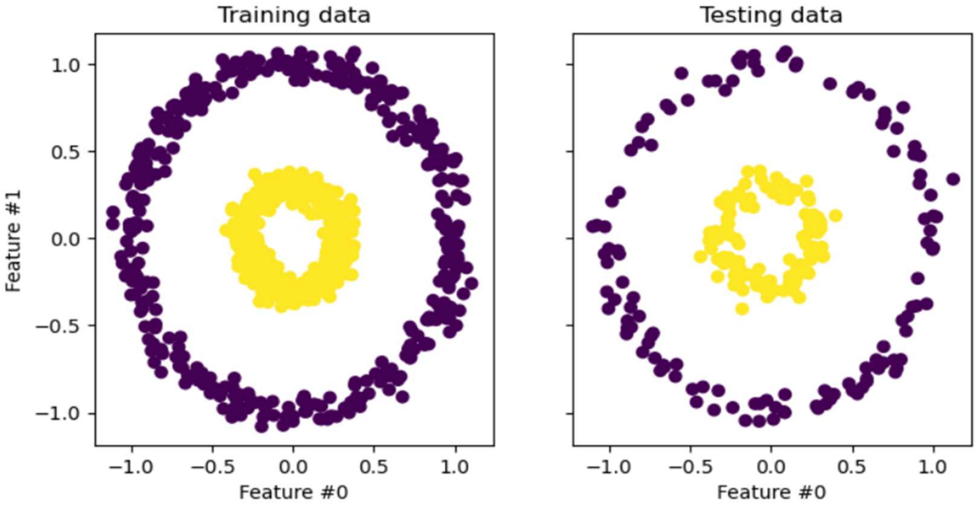
$$FinalDataSet = eigenvectorMatrix^T * originalDataMatrix \qquad (2.7)$$

Among the several appealing characteristics of the PCA algorithm, the most important for this dissertation is the capability of reducing high-dimensional data, into a relatively low-dimensional data, enabling the usage of the given data set as input of other techniques that might not work well with high-dimensional data, as DBSCAN clustering algorithm.

Even though the PCA algorithm can perform dimensionality over data set providing good and adequate results, it has some limitation since it is validated for linearly separable data, however, many real-world data require nonlinear methods in order to perform tasks that involve the analysis and discovery of patterns adequately (RASCHKA, 2014). In cases involving such kind of data, to achieve non-linear dimensionality reduction, a variation of the PCA algorithm can be used. This variation makes usage of positive definite kernel functions to efficiently compute principal components in high-dimensional feature spaces. This kernel-based method for performing a nonlinear Principal Component Analysis is called Kernel Principal Component Analysis (KPCA) (SCHÖLKOPF; SMOLA, 2002).

In Figure 2.15 the data points are from the different features and cannot be linearly separated by a straight line.

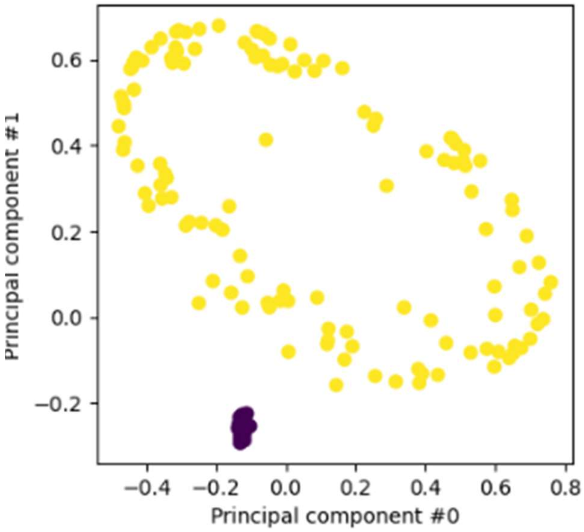
Figure 2.15 – Non-linearly separable data (on the left) to be used as input to an PCA algorithm and its outcome (on the right).



Source: Scikit-Learn (2022).

To be capable of dealing with linearly inseparable data, the idea is to project the data onto a higher dimensional space where it becomes linearly separable, as depict in the Figure 2.16.

Figure 2.16 - Projection of the non-linear separable data set with KPCA.



Source: Scikit-Learn (2022).

This process of high dimensional mapping of the data coming from lower dimensions in a high dimension space is described as:

$$\phi: R^2 \rightarrow R^3$$
$$(X_1, X_2)| \rightarrow (X_1, X_2, X_1^2 + X_2^2)$$

This is done making usage of a nonlinear mapping function ϕ which will map the data points χ of the original data set as $\chi \rightarrow \ \phi(\chi)$, called the kernel function. Assuming this mapping, a covariance matrix, can be represented in the new feature space as the Kernel matrix $\boldsymbol{K}$, see Equation (2.8) (SCHÖLKOPF et al., 1997).

$$\boldsymbol{K} = \Phi(\mathrm{X})\Phi(X)^T \tag{2.8}$$

The dot product $\phi(x_i) \cdot \phi(x_j)$ in the Equation (2.8) regards to the measure of similarity between the instances $x_i and\ x_j$, in the transformed space. It calculates this similarity using the original attribute set without really mapping each data point to a high-dimensional space which is a very costly operation the algorithm makes usage of the so called "kernel trick". The "kernel trick" implies to use a defined kernel function, such as Gaussian, or RBF, or Sigmoid to express the similarity between instances in the original space (SHAWE-TAYLOR; CRISTIANINI, 2004). The dot product between two input vector **v** and **w** in the transformed space, expressed in terms of the original space, after some algebraic simplification, is denoted as the Equation (2.9):

$$K(\mathbf{v}, \mathbf{w}) = \ \Phi(\mathbf{v}) \cdot \Phi(\mathbf{w}) = (\mathbf{v} \cdot \mathbf{w} + 1)^2 \tag{2.9}$$

Considering a mapping function $\Phi: (x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2x_1}, \sqrt{2x_2}, 1)$

This way, the Kernel matrix ***K,*** scaled with the number of data points***,*** takes the form of following representation:

$$\boldsymbol{K_{ij}} \coloneqq \left(\phi(x_i) \cdot \phi(x_j)\right) = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_d) \\ \vdots & \ddots & \vdots \\ K(x_d, x_1) & \cdots & K(x_d, x_d) \end{bmatrix} \tag{2.10}$$

In summary, the KPCA algorithm can be seen as the following (SHAWE-TAYLOR; 2004):

a) Choose a Kernel function $k(x_i, x_j)$, calculate the kernel matrix K;

b) Center the data in feature space by computing the centered kernel matrix $\widetilde{K} = K - \mathbb{1}_n K - K\mathbb{1}_n + \mathbb{1}_n K \mathbb{1}_n$;

c) Solve the eigenproblem $[V, \Lambda] = eig(K)$. Where $\alpha^j = {\lambda_j}^{-1/2} v_j, j = 1, \dots, k$;

d) Compute the low dimensional representation points y_i with the following formula:

$$\widetilde{x_\iota} = \left(\sum\nolimits_{i=1}^{n} \alpha_i^j k(x_i, x)\right)_{j=1}^{k} \tag{2.11}$$

Having as an output the transformed data $\tilde{S} = \{\widetilde{x_1}, \dots\dots \widetilde{x_n}\}$.

One of the main advantages of the Kernel PCA method is that its process is essentially linear algebra, having no nonlinear optimization involved, and as the standard PCA algorithm, only Eigenvalue problem has to be solved. Another important aspect is that the algorithm does not require the number of components of the output space as an input (SCHÖLKOPF; SMOLA, 2002). PCA feature extraction has found application in many areas, including noise reduction, pattern recognition, regression estimation, and image indexing. In all cases were taking into account nonlinearities might be beneficial, kernel PCA provides a new tool which can be applied with little computational cost and possibly substantial performance gains (SCHÖLKOPF et al., 1997).

3 RELATED WORKS

This Chapter presents the literature related to this dissertation topic. It covers the usage of machine learning algorithms for failures detection and health monitoring, those being applied to space systems or not. A summary of the reviewed methods is given in Table 3.1.

Machine learning algorithms are known by presenting a very good accuracy to prediction and usage of Data mining techniques for analyzing telemetry data to anomaly detection applied on the space. In one of these studies, Azevedo et al. (2012) presented an effectiveness comparison between two clustering algorithms, K-means and Expectation Maximization (EM), applied in two different study cases from real satellites. K-means divides the data set with the number of groups represented by the variable K. EM is an iterative algorithm for learning probabilistic categorization model from unlabeled data. They have calculated the dissimilarity indexes using both Euclidean and Manhattan distances, then they have used clustering algorithms. The study showed the clustering algorithms approach can be good to anticipate anomalies given some constraints on the problem domain, more precisely, when several telemetries behave abnormally and at least one of them tends to go out of limit. The advantages of the usage of clustering approaches relies on the fact that these clustering algorithms work with unsupervised learning, being capable of use unlabeled data, working in an automated way.

Gao et al. (2012) proposed a method based on Principal Component Analysis (PCA) and Support Vector Machine (SVM). The PCA is used to extract feature vectors reducing the complexity and dimensionality, which is an inherent feature in telemetry data coming from space system applications. While the Binary SVM, which can be seen as a field of pattern recognition that classifies data into two classes, was used to detect faults. In order to tailor the binary aspect of it and be able to identify fault types, a Multi-class SVM approach was used. The result shows that the method is efficient and practical for fault detection and diagnosis of spacecraft systems.

However, hybrid approaches, combining supervised and unsupervised methods, are researched as seen in the usage of unsupervised learning algorithm to cluster time series data by (BISWAS et al., 2016). This study revealed the hypothesis that the larger groups of clusters unveil normal behavior, but smaller groups show anomalous situations. They have used hierarchical clustering method followed by a verification of the outcome by consulting domain experts. Although they achieve good results, this method can be seen as not automate and cheap.

In a more recent study, Ibrahim et al. (2019) presented an approach chain ''K-means – LAD – FTA". The telemetry data is clustered, and classified in failure and non-failure, using K-means clustering algorithm based on t-SNE function which is used for dimensionality reduction. The clustered data is used as an input to LAD. As a result of classification, LAD generated positive patterns that are unique for each class and indicate conditional parameters values. Probabilities are estimated from FTA to get the most probable cause that lead to the satellite failure estimating the probability and dependencies of each parameter range with respect to the top event. LAD discovers the general patterns of failure; while FTA defines which basic event (corresponds to certain pattern) causes satellite failure.

In Yairi et al. (2017) the authors proposed a new health-monitoring and anomy detection method for artificial satellites, where a mixture of probabilistic principal component analyzers and categorical distributions, where, real-valued continuous variables are handled by the mixture of probabilistic principal component analyzers, while categorical discrete values are modeled by a categorical mixture distribution. Being able this way of performing dimensionality reduction and clustering over the satellite housekeeping data. Apart from the algorithm used to generate their model, this study also speaks about data processing, an important step to be applied before performing the model generation.

Another way of providing a reliable health monitoring system, as far as satellites goes aging, is based on the use of operational satellite simulators to support their operational procedures. But, how to make aging the models of an operational satellite simulator? The framework proposed by Rodrigues et al. (2021) uses an Artificial Neural Network (ANN) to reproduce the battery voltage behavior of a large sun-synchronous remote sensing satellite, the CBERS-4, considering its aging. The study makes usage the genetic algorithm to find the best network architecture of ANN, allowing then to obtain a model which presented less than 1% of error. Such simulators, due their high fidelity can be used to generate the data of satellites failure scenarios which could be used to train failure diagnostics algorithms, overtaking this difficulty associated with the lack of available data when comes to system abnormal behavior.

The research on the usage of data science and artificial intelligence on the forecast of anomalous behavior on a system is wide-spread and applied, not only to space systems and their telemetries, but to several different areas of science and technology as well (ZHANG, 2016).

Li et al. (2018) present a study case of a Nuclear Power Plant (NPP), applying a method based on Principal Component Analysis (PCA) for condition monitoring of the sensors used in the NPP. The goals of the study are fault detection, identification, and reconstruction of sensors data. The authors combine various methods, like statistics-based ones, with PCA method to optimize the model performance. Their approach has led them to reach valuable achievements.

Another application out of the space field, makes usage of machine learning to perform condition-based maintenance and repair as demonstrated in Purarjomandlangrudi, Ghapanchi and Esmalifalak (2014). The authors apply classification to distinguish among defect examples from rolling-element bearing in rotative machinery. The proposed method extracts from the given data two different features, kurtosis and Non-Gaussianity Score (NGS) in order to develop the anomaly detection algorithm.

Another application on bearing fault diagnosis is performed by Safizadeh (2014), where a novel Condition-based Monitoring system (CBM) consisting of sensing, signal processing, feature extraction, classification, high-level fusion and decision-making module has been proposed. The study applies a machine learning algorithm on the feature extraction module, through Principal Component Analysis (PCA), and later on in the fourth module. K-Nearest Neighbor (KNN) classifier has been used in order to identify the condition of the ball bearing based on vibration signal and load signal. Finally, in the last module of the Waterfall Fusion Process Model, a logical program is used to decide about the condition of the ball bearing. According to the authors, experimental results show the effectiveness of this method.

Wang et al. (2021) proposes on a technique aiming fault diagnosis on Nuclear Power Plants. In this paper, the Kernel Principal Component Analysis (KPCA) is primarily presented for fault detection and feature extraction. After that, support vector machine is carried out for fault diagnosis. Subsequently, a similarity clustering algorithm is used for analyzing degree. The authors claim that opposed to other ‘black box’ data-driven methods; this technique allows the results to be illustrated in a visual form in order to be assessed by operators.

In the literature it may be found works combining, in different manners, tools, techniques and algorithms, to deal with the complex inherent characteristic of the data under analysis, to make the process more automated and reliable, but above all to approach more efficiently the fault analysis and diagnoses problem when comes to detection of a real problem.

The approach in this dissertation proposes a new data-driven health monitoring and anomaly detection method for artificial satellites based on probabilistic dimensionality reduction and clustering, taking into consideration the miscellaneous characteristics of the spacecraft housekeeping data.

Table 3.1 summarizes all the approaches and applications in the papers discussed in this Chapter. The columns 3 and 4 highlight the used methods and the application of each work respectively.

Table 3.1 – Summary of the discussed Articles.

Author,date	Approach	Method	Application
(AZEVEDO et al., 2012)	K-means algorithm and Expectation Maximization are evaluated for anomaly detection in telemetry data. The results are compared to each other. In these experiments, an anomaly could be detected 6 hours in advance.	Clustering (Unsupervised Learning)	Telemetry data of two real cases of satellite anomalies in Brazilian space mission
(GAO et al.,2012)	Firstly, dataset is limited with Euclidean distance. Then, k-Nearest Neighbors (kNN) algorithm is used to anomaly detection.	Unsupervised learning, Nearest Neighbor Algorithm	TM data of the power subsystem of in-orbit satellite.
(SAFIZADEH, 2014)	Processed data has its dimensionality reduced by PCA. Then kNN algorithm is used over the two principal component data to decide which maintenance action should be taken.	Unsupervised learning, Nearest Neighbor Algorithm	Data from two different sensors are placed on the same mounting
(PURARJOMANDLANGRUDI et al., 2014)	An Anomaly Detection (AD) algorithm defined by the author and a Support Vector Machine (SVM) are applied. The results are compared to each other	Supervised learning, Support Vector Machine.	Data from different accelerometer sensors placed on the bearing
(BISWAS et al., 2016)	Unsupervised learning is used to cluster large telemetry database of time series data. Expert opinion is used for some inputs (supervised method).	Mixed Method (Unsupervised and Supervised methods)	Telemetry data from the EPS of the LADEE spacecraft.

continue

Table 3.1 – Continuation.

Author,date	Approach	Method	Application
(YAIRI et al., 2017).	New data-driven health monitoring and anomaly detection method based on probabilistic dimensionality reduction and clustering. Comparisons were made against one-class support vector machine and support vector data description algorithms.	Unsupervised method, Mixture of Probabilistic Principal Component Analysis and Categorical Distribution (MPPCACD)	Telemetry data from the Small Demonstration Satellite 4 (SDS-4) of JAXA.
(LI et al., 2018)	PCA is applied for fault detection allied to statistic-based methods to reduce the false alarms. Sensor measurement from a real NPP is used to evaluate the optimization of the PCA performed.	Unsupervised method, Principal Component Analysis	Data coming from sensors in a Nuclear Power Plant (NPP)
(IBRAHIM et al., 2019)	Summary of the performance of processing telemetry data using AutoRegressive Integrated Moving Average (ARIMA), Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), Long Short-Term Memory Recurrent Neural Network (LSTM RNN), Deep Long Short-Term Memory Recurrent Neural Networks (DLSTM RNNs), Gated Recurrent Unit Recurrent Neural Network (GRU RNN), and Deep Gated Recurrent Unit Recurrent Neural Networks (DGRU RNNs).	Mostly supervised method, MLP, RNN, LSTM RNN, DLSTM RNNs, GRU RNN, DGRU RNNs	Telemetry data of battery temperature, power bus voltage and load current received from Egyptsat-1 satellite.

continue

Table 3.1 – Conclusion.

Author,date	**Approach**	**Method**	**Application**
(WANG *et al.*, 2020)	KPCA is applied in two stages, at the beginning to reduce false alarms, later, for feature extraction. At the end, a similarity clustering is combined to verify results coming from the SVM algorithm	Unsupervised method, Kernel Principal Component Analysis (KPCA), SVM	Data coming from a reactor coolant system of pressurized water reactor
(RODRIGUES e t al., 2021)	An ANN is used to obtain accurate models for operational satellites simulators that enable analyses of satellite behavior over time.	Supervised method, Artificial Neural Network (ANN)	Telemetry data from CBERS 4 satellite

Source: Author's production.

4 THE MACHINE LEARNING PROCESS

This Chapter presents the proposed machine learning process in a break down structure manner, starting from the high overview and descend to detailing each step. The steps are described and illustrated with information of a case study. Two case studies were developed. Section 4.1 introduces the case studies planning, made with telemetry data from two satellites, and some of its characteristics. Section 4.2 describes the proposed process.

4.1 Case study planning

This Section presents the two satellites, the China-Brazil Earth Resources Satellite One (CBERS-1) and the Data Collection Satellite Two (SCD2), that were taken as case study to experimentally demonstrate the capabilities and constraints of the proposed process. Both systems are briefly presented in the following. Considerations regarding the experiment conductions are presented in Section 4.1.3.

4.1.1 CBERS-1 satellite

The CBERS-1 satellite was used as study case in this dissertation for the design phase of the machine learning process. CBERSs satellites were developed in a cooperative program between the Chinese Academy of Space Technology (CAST) of the People's Republic of China, and the Instituto Nacional de Pesquisas Espaciais (INPE) of Brazil. The program goal was to establish a complete remote sensing system to supply both countries with multispectral remotely sensed imagery Figure 4.1 depicts a commemorative mission patch for the CBERS mission stablished between China and Brazil.

Figure 4.1 – China-Brazil Earth Resources Satellite program patch.

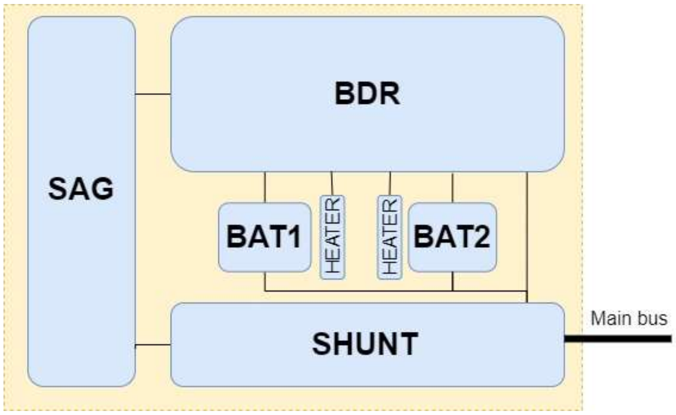
Source: INPE (2022).

CBERS-1 is the first satellite of its series, was launched in 1999 and kept itself operational for 45 months, being decommissioned in July of 2003, its orbit cycle was of 26 days, making 14 revolutions per day, having a nodal period of 100 min and an eclipse time of 35 min (BO, 2002).

The CBERS-1 satellite payload module accommodated an optical system comprising a High-Resolution CCD Camera, an Infrared Multispectral Scanner, and a Wide-Field Image, used for Earth observation and the Repeater for the Brazilian System of Environmental Data Collection. The service module contained the subsystems that ensures the satellite basic operation such as generating, conditioning and distributing power supply, the satellite's controls, the satellite's telecommunications and other functions.

The 1100 W of electrical power necessary for the operation of the on-board pieces of equipment was obtained through solar panels, which were the satellite's primary source of energy, these ones opened when the satellite was placed in orbit and were continuously oriented towards the Sun by automatic control. A block diagram representation of the power subsystem can be seen in Figure 4.2.

Figure 4.2 - Power Supply Subsystem block diagram.



This diagram depicts a simplified model of the CBERS-1 Power Supply Subsystem (PSS), where the Solar Array Generator (SAG), the SHUNT, the Battery Discharge Regulator (BDR), ant the two batteries are identified.

Source: Adapted from Azevedo (2012).

The energy obtained is stored on the batteries, a secondary source of energy for the satellite. Apart from generating and conditioning, the power subsystem was also responsible for distributed the energy, at different voltage levels, to the other subsystems (INPE, 2022).

The CBERS-1 satellite had more than 2000 different types of telemetry channels available to be analyzed by the operators. These represented values measured by sensors and equipment status. They were classified as analog measurements, representing analogical values such current, voltage level, temperature, and so on, or as binary, indicating, for example, an equipment state (ON/OFF).

This dissertation made usage of telemetry data related to the PSS. Usually, the satellite telemetry data is characterized by the presence of continuous (analog) and discrete (binaries) variables, however, in this case study, only the continuous values telemetries were used. Even though discrete values telemetries could give insight regarding the state of some mechanism of the subsystem, it is in the analogic value telemetries that the satellite power supply emerging behavior will be depicted, especially those associated with solar panels, SHUNT, batteries and BDR equipment.

The telemetry data used here corresponds to telemetry acquisitions made in the years 2000 and 2003. However, the available data was not continuous but segmented in some specific months. The available time-window were:

- From January 1st to January 30th of 2000;
- From March 1st to March 16th of 2000;
- From January 1st to January 30th of 2003;
- From March 1st to March 16th of 2003, and;
- From June 1st to June 30th of 2003.

Table 4.1 shows the list of telemetry channels that were analyzed. These telemetries were chosen because they are the most representative of the power subsystem behavior and thus have the greatest potential to identify problems and anomalies (AZEVEDO et al., 2012).

Table 4.1 - Telemetries used from CBERS-1.

Telemetry ID	Description	Limits
TM001	Main Bus Voltage	27 a 29V
TM002	Main Bus Current	0 a 36A
TM003	Main Error Amplifier (MEAS) Output Voltage	8 a 23.5V
TM013/017	BDR Input Current	0 a 13A
TM014/018	Battery voltage	43.2 a 56.5V
TM015/019	Battery Temperature	0 a 10°C
TM016/020	3-cell minimum group voltage	3.6 a 4.65V
TM021	BDR Output Current	0 a 36A
TM022/023	Solar Panel Current (SG1 and SG2)	0 a 7.2A

Source: Author's production.

The CBERS1 case study was divided in two parts. The first part was conducted over the data set taking in respect the following setup:

- Training and Test data set comprehending the observations made from January 1 st to January 30th of 2000, having a distribution of 75% of the data assigned for training the model, and the rest for testing;
- Validation data set comprehending all the observations made from March 1st to March 16th of 2000.
- Taking into analysis all the telemetries described in Table 4.1 but the TM001 due its lack of variability which have no relevance to modeling the power supply subsystem behavior.

The second part of the study case was conducted over the data set taking in respect the following setup:

- Training data set comprehending the observations made from January 1st to March 16th of 2003. Differently from the first part, the whole data set was used as "training".
- Validation data set comprehending all the observations made from June 1st to June 30th of 2003.

4.1.2 SCD2

The Data Collection Satellite Two (SCD2) has over 20 years of continuous operations by the Satellite Control Center at INPE. It was the second satellite designed, tested and assembled in Brazil, from the SCD family of data collection satellites (OLIVEIRA, 1996). The mission goal was to retransmit data obtained from a network of Automatic Environmental Data Collection Platforms (PCD) to assigned receiver stations Figure 4.3 depicts a commemorative mission patch for the satellite.

Figure 4.3 - SCD2 mission patch.

Source: INPE (2022).

The PCDs were distributed throughout the Brazilian territory, and those were composed of a UHF-band transmitter that collected environmental data that were continuously transmitted to the space (MIGUEZ et al., 1993).

The SCD2 satellite is composed of ten subsystems, including the Data Collecting Platform (DCP) payload. With over 20 years of operation, more than 135 telemetry data points and generating over 8GB of data per year, there is a lot to be analyzed from the housekeeping telemetry data alone, composed by 31 different telemetry signals. This work used data taken between 2014 and 2018 and amounts to about 23GB of CSV files.

4.1.3 Case studies planning considerations

The case study using telemetry data from the CBERS1 satellite was made in order to raise the key components associated with an anomaly detection technique, and from that, define the process (also considered as a data science strategy) of performing the anomaly detection task.

The CBERS1 case study was divided in two parts. The **first part of the study case** was conducted over the data set taking in respect the following setup:

- Training and Test data set comprehending the observations made from January 1st to January 30th of 2000, having a distribution of 75% of the data assigned for training the model, and the rest for testing;

- Validation data set comprehending all the observations made from March 1st to March 16th of 2000;
- Taking into analysis all the telemetries described in the Table 4.1 but the TM001 due its lack of variability which translate into not having relevance when comes to modeling the power supply subsystem behavior.

The **second part of the study case** was conducted over the data set taking in respect the following setup:

- Training data set comprehending the observations made from January 1st to March 16th of 2003. This time differently from the first part, the whole data set will be used as "training";
- Validation data set comprehending all the observations made from June 1st to June 30th of 2003.

The case study using telemetry data from the SCD-2 satellite had the same objective as the CBERS-1 case.

In both study cases, the application domain is the same, anomaly detection on telemetry data coming from an artificial satellite for Earth resources observation, which implies that the nature of the input data, availability or not of labels, as well as constraints and requirements shall be similar but not the same.

The input data from both study cases have the same nature and therefore present similar attributes, being those, a mixture of binary and continuous data in form of status indicator telemetries and measurement telemetries, i.e., output current of the solar panel array.

However, even though the nature of the data is the same an unexpected difference among the two data sets arose, the sampling rate, or number of observations for a given period of time is absurdly different, being the data from the SCD2 the one with the highest sampling, more than 100.000 samples.

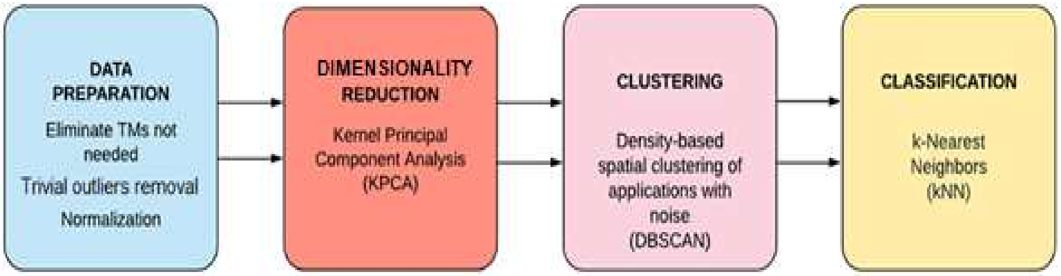
The resource constraint imposed by the limited computational resource, in personal notebooks, currently available for the research, limited the SCD2 dataset analysis. The process had to stop in the Data Preparation due to resources constraints and technical obstacles. The experiments made in the data from the SCD2 satellite could not be run during the machine learning model development. An analysis of alternative was evaluated taking into account current literature, which are discussed in Section 5.3.

4.2 The proposed process

The machine learning process proposed and applied in this research combines different approaches found in the literature.

The **process applied in the first part of the study**, at a high abstraction level, can be understood as the data science process proposed in Schutt and O'Neil (2013). Closely, the data-driven anomaly detection process is divided into four steps, namely, Data Preparation, Dimensionality Reduction, Clustering, and Classification, as depicted in Figure 4.4. The last two steps can be seen as a Machine Learning steps in charge to generate and verify the model, as the approach proposed in Yairi et al. (2017) and Zare (2018) and others mentioned by Taburoğlu (2019).

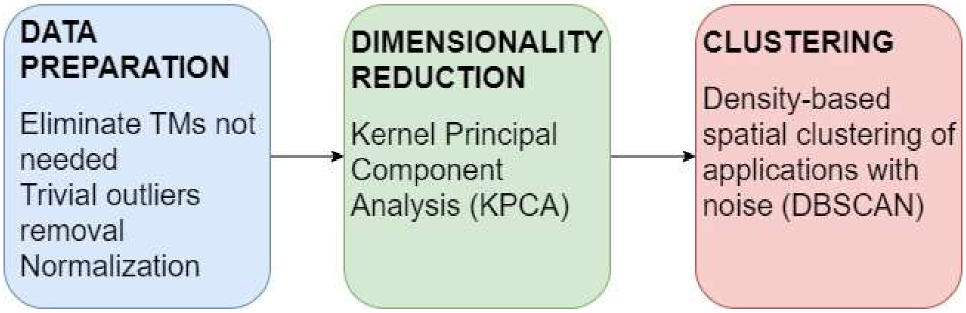
Figure 4.4 - Data-driven anomaly detection flow.


DATA PREPARATION
Eliminate TMs not needed
Trivial outliers removal
Normalization
DIMENSIONALITY REDUCTION
Kernel Principal Component Analysis (KPCA)
CLUSTERING
Density-based spatial clustering of applications with noise (DBSCAN)
CLASSIFICATION
k-Nearest Neighbors (kNN)

Flow of the mixed machine learning process presented in this dissertation, applied for anomalous behavior detections.
Source: Author's production.

The **process applied in the second part of the study** presents a similar flow of the one presented in Figure 4.4, having exactly the three first steps. The process applied in the case study part 2, is depicted in the **Figure 4.5.** Although, in the second part, the machine learning process finished with the clustering step, being the DBSCAN algorithm responsible for "classify" the data identifying patterns in the telemetry that are validate through visual analysis.

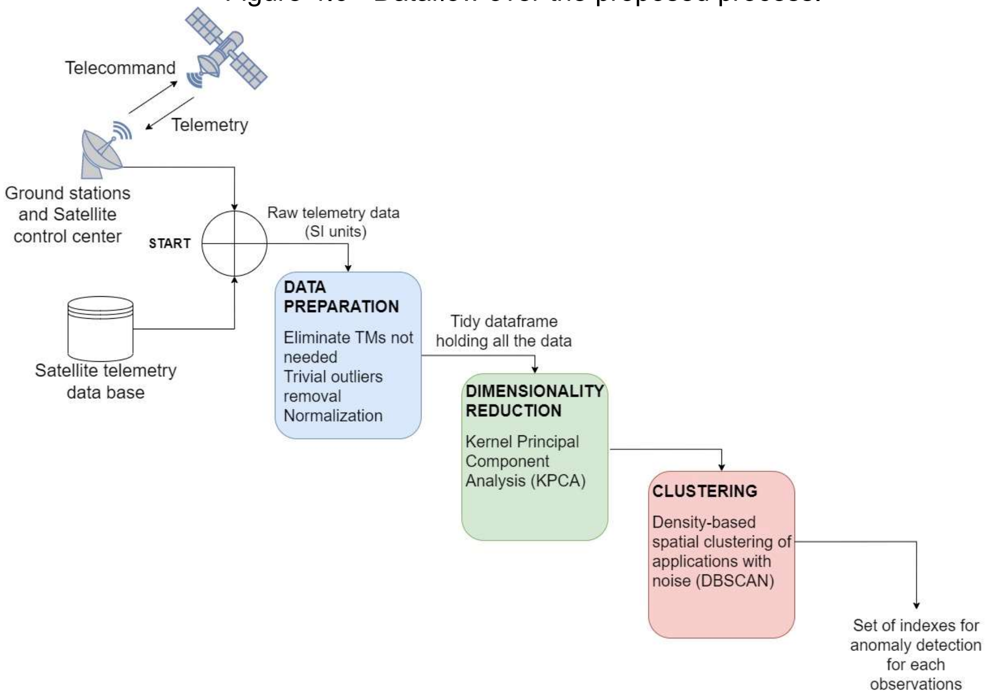
Figure 4.5 - Definitive machine learning process for anomaly detection.



Flow of the mixed machine learning process containing only three steps, applied for anomalous behavior detections in the case study part 2.
Source: Author's production.

The flow of information of the process is shown in the **Figure 4.6**. There, the start of the process has an input of raw data which may come either from a database or from an online source represented by the satellite control center.

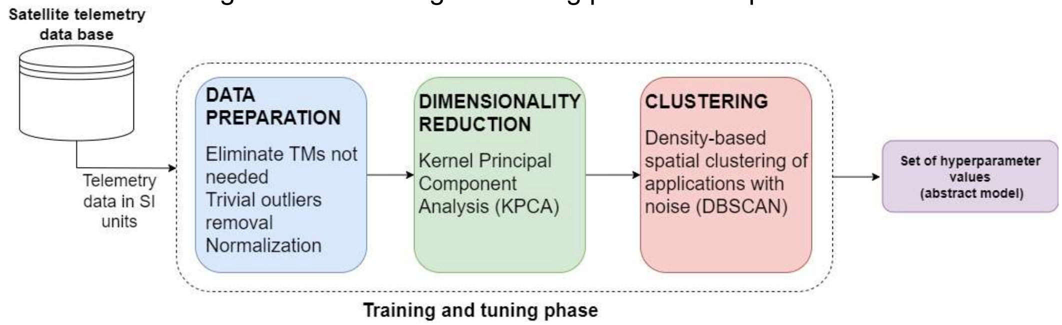
Figure 4.6 - Dataflow over the proposed process.


Telecommand
Telemetry
Ground stations and Satellite control center
START
Raw telemetry data (SI units)
Satellite telemetry data base
DATA PREPARATION
Eliminate TMs not needed
Trivial outliers removal
Normalization
Tidy dataframe holding all the data
DIMENSIONALITY REDUCTION
Kernel Principal Component Analysis (KPCA)
CLUSTERING
Density-based spatial clustering of applications with noise (DBSCAN)
Set of indexes for anomaly detection for each observations

Source: Author's production.

The process, to be able of detecting anomalous behavior, requires a pre-step which has the objective of training and tuning the hyperparameters used as input for the algorithms applied. The training and tuning step, depicted in **Figure 4.7**, provides a set of hyperparameters values as an input to the process when it is used for online anomaly detection. The set of hyperparameters is also referred as abstract model.

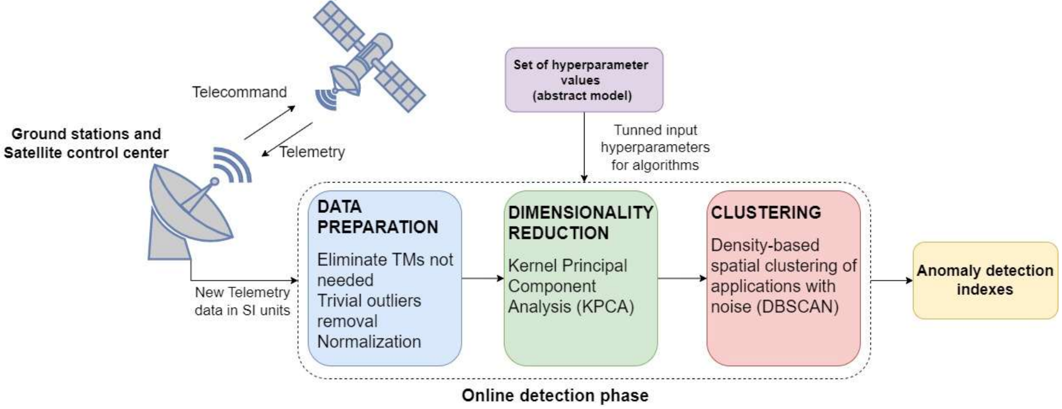
Figure 4.7 - Training and tuning phase of the process.


Satellite telemetry data base
Telemetry data in SI units
DATA PREPARATION
Eliminate TMs not needed
Trivial outliers removal
Normalization
DIMENSIONALITY REDUCTION
Kernel Principal Component Analysis (KPCA)
CLUSTERING
Density-based spatial clustering of applications with noise (DBSCAN)
Set of hyperparameter values (abstract model)
Training and tuning phase

Source: Author's production.

As shown in **Figure 4.8**, differently from the tuning phase which uses historical data from a database, the online anomaly detection phase uses data, which is online gathered, in other words, data coming from the satellite control center during a satellite pass. The online phase has two inputs, one is the telemetry data coming in, and the other one is the abstract model.

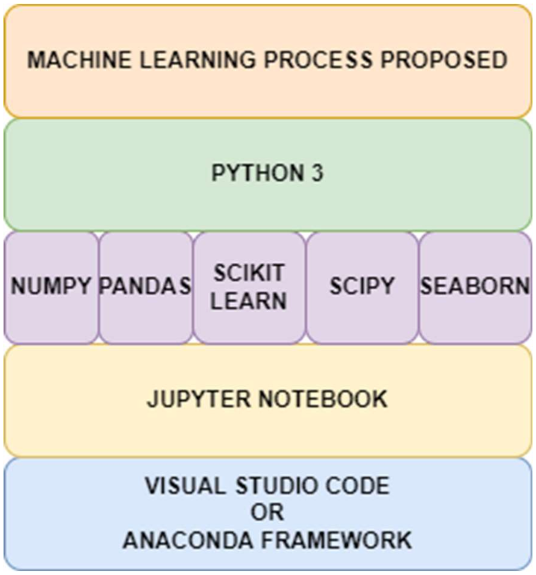
Figure 4.8 - Online detection phase.


Telecommand
Telemetry
Ground stations and Satellite control center
Set of hyperparameter values (abstract model)
Tunned input hyperparameters for algorithms
New Telemetry data in SI units
DATA PREPARATION
Eliminate TMs not needed
Trivial outliers removal
Normalization
DIMENSIONALITY REDUCTION
Kernel Principal Component Analysis (KPCA)
CLUSTERING
Density-based spatial clustering of applications with noise (DBSCAN)
Anomaly detection indexes
Online detection phase

Source: Author's production.

All the processes performed on both studies were made using Python as programming language, Jupyter Notebook as framework, Visual studio code as code editor (having Anaconda environment as an option), and free third-party libraries as Pandas, Scikit-learn, Scipy, Seaborn among others. Figure 4.9 depicts a layered-architecture diagram of the set of tools and solutions used to process the telemetry data. This way it is possible to see the different software entities that were used to achieve the outcome obtained with the demonstrated machine learning process.

Figure 4.9 - Layered- architecture diagram of the toolchain used during the studies.



Here is depicted the main parts used in this framework. Other libraries like matplotlib and plotly are not depicted. The code editor used is also not highlighted since it is irrelevant when comes to the outcome of this study.
Source: Author's production.

Make usage off-the-shelf solutions relies on the fact that the aim of the study was to study the possibility of making usage of machine learning algorithms to perform anomalous behavior detection, and not the development of tools or algorithms implementations.

4.2.1 Data preparation

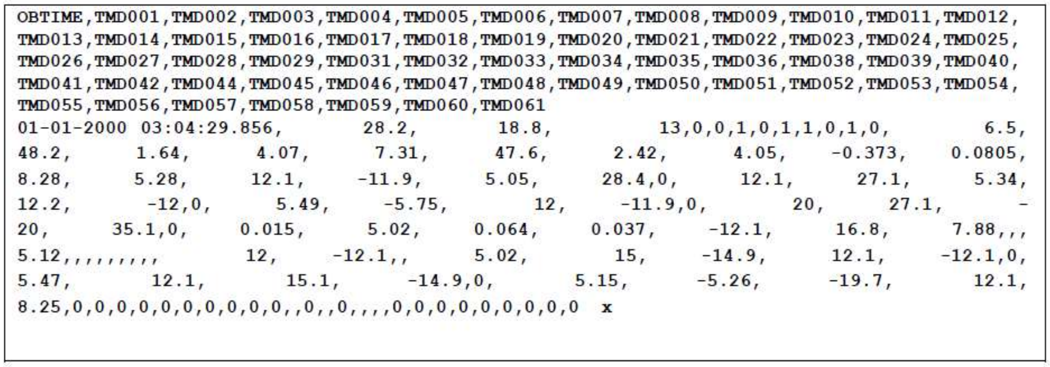
In a nutshell, the data preparation, or data processing, comprehends the following tasks: to import the raw data from the files, to clean the data making it tidy, to remove the trivial outliers and finally, to normalize the data. However, before that, the data shall be imported from where it is contained as detailed in the following section.

4.2.1.1 Importing data

The task of importing data is fully dependent of the data format and the storage. In the CBERS case study, the telemetry data was provided in .csv (comma separated value) format and divided into different files so that each file represents an observation window for a given month in a given year. When it is said that the mass of data was stored in .csv format it means that without the use of an interpreter of such format, the data looks as seen in

Figure 4.10.

Figure 4.10 - Raw set of archived TM data in *.csv format.

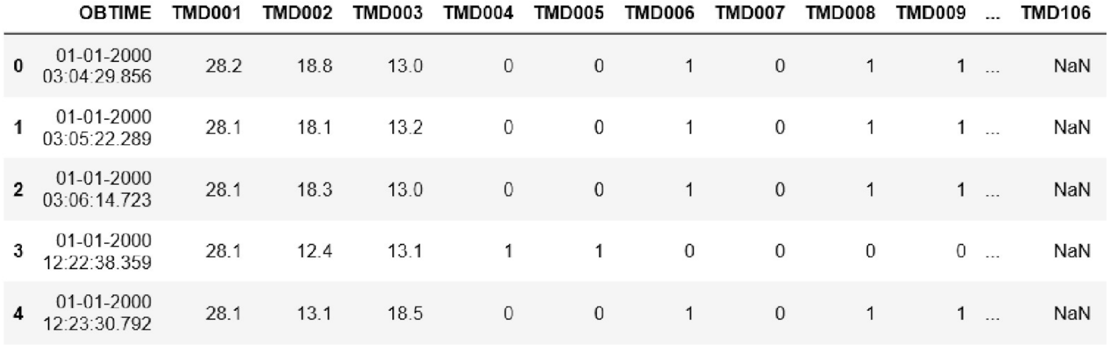


```
OBTIME,TMD001,TMD002,TMD003,TMD004,TMD005,TMD006,TMD007,TMD008,TMD009,TMD010,TMD011,TMD012,
TMD013,TMD014,TMD015,TMD016,TMD017,TMD018,TMD019,TMD020,TMD021,TMD022,TMD023,TMD024,TMD025,
TMD026,TMD027,TMD028,TMD029,TMD031,TMD032,TMD033,TMD034,TMD035,TMD036,TMD038,TMD039,TMD040,
TMD041,TMD042,TMD044,TMD045,TMD046,TMD047,TMD048,TMD049,TMD050,TMD051,TMD052,TMD053,TMD054,
TMD055,TMD056,TMD057,TMD058,TMD059,TMD060,TMD061
01-01-2000 03:04:29.856,       28.2,       18.8,          13,0,0,1,0,1,1,0,1,0,        6.5,
48.2,      1.64,      4.07,      7.31,      47.6,      2.42,      4.05,    -0.373,    0.0805,
8.28,      5.28,      12.1,     -11.9,      5.05,      28.4,0,      12.1,      27.1,      5.34,
12.2,       -12,0,      5.49,     -5.75,        12,     -11.9,0,        20,      27.1,         -
20,      35.1,0,     0.015,      5.02,     0.064,     0.037,     -12.1,      16.8,      7.88,,,
5.12,,,,,,,,,         12,     -12.1,,      5.02,        15,     -14.9,      12.1,     -12.1,0,
5.47,       12.1,       15.1,      -14.9,0,       5.15,      -5.26,      -19.7,         12.1,
8.25,0,0,0,0,0,0,0,0,0,0,,0,,0,,,,0,0,0,0,0,0,0,0,0  x
```

Source: Author's production.

To import the data from this source and place it in the Pandas DataFrame format, the 'read_csv' function was used, from the Pandas library, which provides a series of different functionalities, but here it was only used to export the mass of data found in a path inside the computer used in the study. The Pandas DataFrame is a two-dimensional, changeable in size and potentially heterogeneous, tabular data structure with labeled axes (rows and columns). Arithmetic operations are aligned on row and column labels. It can be thought of as a dictionary-like container for series objects. The function on data returns organized in a data-frame-tidy as seen in the Figure 4.11.

Figure 4.11 - First visualization of archived TM data using Pandas Library.



	OBTIME	TMD001	TMD002	TMD003	TMD004	TMD005	TMD006	TMD007	TMD008	TMD009	...	TMD106
0	01-01-2000 03:04:29.856	28.2	18.8	13.0	0	0	1	0	1	1	...	NaN
1	01-01-2000 03:05:22.289	28.1	18.1	13.2	0	0	1	0	1	1	...	NaN
2	01-01-2000 03:06:14.723	28.1	18.3	13.0	0	0	1	0	1	1	...	NaN
3	01-01-2000 12:22:38.359	28.1	12.4	13.1	1	1	0	0	0	0	...	NaN
4	01-01-2000 12:23:30.792	28.1	13.1	18.5	0	0	1	0	1	1	...	NaN

Source: Author's production.

The data is then extracted from a CSV file and shaped as a pandas data frame. This data frame still presents raw telemetry value that has to be processed to turn it into tidy before being used (WICKHAM, 2014). Each telemetry channel presents a variable with different engineering magnitudes represented by different measurements units. The telemetry data, now shaped as a data frame, present each telemetry variable place in a column, these can be referred as features or dimensions. Even though the data in Figure 4.11 data frame presents a better organization.

4.2.1.2 Data cleaning

The task of cleaning the data consist of identifying and correcting mistakes or errors in the data which can disturb the analysis to be performed by an algorithm. In the CBERS case, this task consists of removing telemetries signals which present the following characteristics:

a) Variables values which are not continuous.

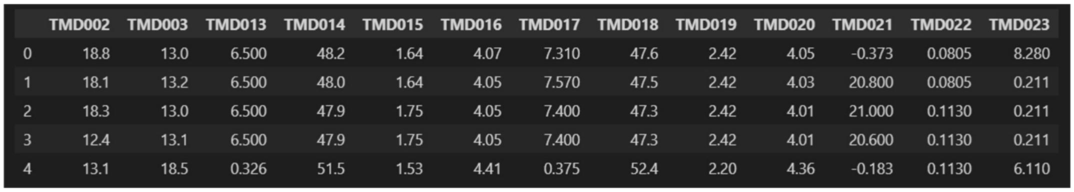
b) Time stamp variables.

c) Variables with low variation, which are nearly constant over time.

d) Variables which are not related to subsystem used as test case.

These characteristics were considered to be either irrelevant to the purpose of this study or would impose more complexity to the proposed process of analyzing multi type variables. The exclusion of such telemetries variables is easily achieved through the "drop" method from data-frame class contained at the pandas library. The results of the data-frame head, after executing the clearing task according to the established rules, can be seen in Figure 4.12.

Figure 4.12 - Dataframe after some clean up.



	TMD002	TMD003	TMD013	TMD014	TMD015	TMD016	TMD017	TMD018	TMD019	TMD020	TMD021	TMD022	TMD023
0	18.8	13.0	6.500	48.2	1.64	4.07	7.310	47.6	2.42	4.05	-0.373	0.0805	8.280
1	18.1	13.2	6.500	48.0	1.64	4.05	7.570	47.5	2.42	4.03	20.800	0.0805	0.211
2	18.3	13.0	6.500	47.9	1.75	4.05	7.400	47.3	2.42	4.01	21.000	0.1130	0.211
3	12.4	13.1	6.500	47.9	1.75	4.05	7.400	47.3	2.42	4.01	20.600	0.1130	0.211
4	13.1	18.5	0.326	51.5	1.53	4.41	0.375	52.4	2.20	4.36	-0.183	0.1130	6.110

Source: Author's production.

It is important to highlight that it is valuable to check whether there are NaN values within the data because those ones can jam the execution of some machine learnings algorithm applied here. These values when present, are replaced by an interpolation of the two values around the NaN value.

This way, at the end of this sub process, a data frame is ready to be taken to the next step of the data preparation.

4.2.1.3 Trivial outlier removal

The trivial outlier removal step may be optional depending on the clustering algorithm. In the proposed process of this research, it is used a mixture model approach which uses a dimensionality reduction algorithm to preprocess the data before using a clustering algorithm. The chosen clustering algorithm is the DBSCAN mainly due its ability to deal with noise data. This clustering algorithm could be enough to handle the kind of data available for this study, due its "noise resistant" characteristic (ESTER et al, 1996), therefore a trivial outlier removal would not be needed. However, this statement is only true if the dataset is given directly to the clustering algorithm, in other words, if the data is not preprocessed with a dimensionality reduction algorithm beforehand, as described in Yairi et al. (2017).

In Yairi et al. (2017) the authors discuss about an effect called "Trivial Outliers", which is defined as abnormal values caused by errors in data conversion or transmission. Another study found in the literature on how to handle outliers which are no related to failures or anomalous behavior is in Li et al. (2018) when talking about data preprocessing the authors proposed a "Singular points and Random fluctuations eliminations" through the use of arithmetic average and standard deviation.

The method presented in Yairi et al. (2017) was adopted in this study due its easy implementation. The authors start stating a common understanding that trivial outlier occur very rarely and abruptly, and furthermore, present values very different from those of the neighboring data points. In face of that, they calculate the upper and lower bounds of acceptable values, i.e., the thresholds values, presented in equation (4.1) and (4.2) by theta upper and theta lower.

$$\theta_{upper} = P_{100-\alpha} + \beta * (P_{100-\alpha} - P_{\alpha}) \tag{4.1}$$

$$\theta_{lower} = P_{\alpha} - \beta * (P_{100-\alpha} - P_{\alpha}) \tag{4.2}$$

Here are depicted the formulas used for calculating the upper and lower limits of a given signal, or feature.

Source: YAIRI, Takehisa et al. A Data-Driven Health Monitoring Method for Satellite Housekeeping Data Based on Probabilistic Clustering and Dimensionality Reduction. IEEE Transactions On Aerospace And Electronic Systems, [S.L.], v. 53, n. 3, p. 1384-1401, jun. 2017. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/taes.2017.2671247.

In order to calculate the boundaries, conservative values of alpha and beta were chosen. Being alpha equals to 1.0 and beta equals to 0.5 (YAIRI et al., 2017). With the threshold values in hands, every data point is evaluated according to the following rule:

- If a value y(t), at time step t, exceeds the one of the thresholds, but its previous and next values, i.e., y(t)−1 and y(t)+1, do not exceed the thresholds, then the current value, y(t), is considered as a trivial outlier and therefore, removed.

The considered trivial outliers are replaced by NaN values which in a further step of the implementation are replaced by a value which is the result of the interpolation of the data point values around the NaN value, i.e., y(t)−1 and y(t)+1. This way, with trivial outliers removed from the dataset, the normalization step can be taken.

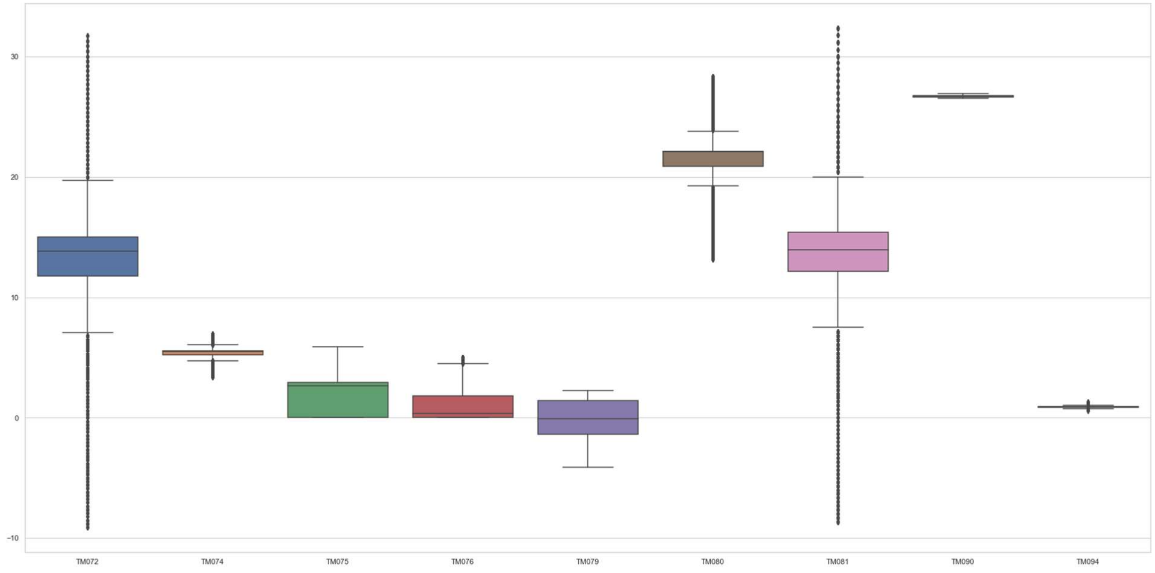
4.2.1.4 Normalization

A satellite system telemetry data has the main purpose of provide information from a variety of subsystems measurements. The transmitted data from the satellite to ground systems present the following features: diverse physical units, don´t share the same scale, different magnitude, or numerical range, and is multivariate data, as exemplified by Yairi et al. (2017).

Another point is that the feature scaling turns out to be an important step since the machine learning algorithms (the DBSCAN), used here, can have their output biased by the lack of normalization of the data features. The DBSCAN utilizes the distance (Euclidean) between points to determine similarity among then, unscaled data creates a problem. Usually, telemetry data that refers to the measurement of a given variable or signal have different orders of magnitude, or scale, due to the sensor used for such measurement or the way it was stored in the database. The lack of scaling, in principal component analysis, can also cause a problem making one feature mistakenly considered more important than another one, as highlighted by scikit-learn developers (PEDREGOSA et al., 2011).

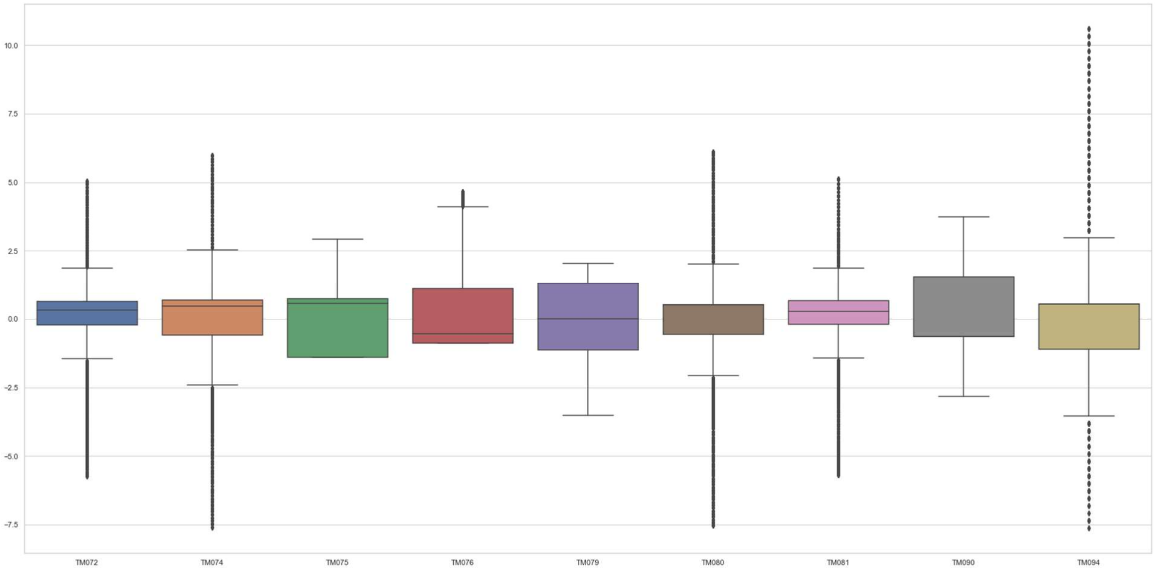
The normalization applied over each data feature is performed by the Standard Scaler method (SKLEARN.PREPROCESSING.STANDARDSCALER, 2022), through the scikitlearn preprocessing package (PEDREGOSA et al., 2011). In general terms, the data features are normalized through the performance of a standardization by removing the mean and scaling to unit variance. The outcome of normalization compared to a non-normalized data can be seen in Figure 4.13 and Figure 4.14.

Figure 4.13 - Non-normalized data.



Source: Author's production.

Figure 4.14 - Normalized data.



Source: Author's production.

This boxplot representation shows the comparison between a non-normalized and a normalized set of telemetry data. In Figure 4.13, the dataset before normalization presenting different magnitudes. In Figure 4.14, a normalized one, has the main characteristic of having a centralized mean.

4.2.2 Dimensionality reduction

The usage of the dimensionality reduction in this study was inspired by the lessons learned gotten from Azevedo et al. (2012) and supported by the idea that in machine learning, data with high dimensionality and multimodality, frequently are treated by dimensionality reduction as shown in Yairi et al. (2017). Furthermore, the clustering algorithms applied in Rasyid et al. (2018) and Molchanov and Linsen (2018), i.e., DBSCAN and K-Means, tends to present limitations to accurately define cluster when the data set has too many features, or presents high dimensionality.

It is important to highlight that through the research and development of the proposed method, different algorithms for dimensionality reduction were applied, namely, the t-Distributed Stochastic Neighbor Embedding (t-SNE), the Principal Component Analysis (PCA) and its Kernel variation. Among then, the Kernel Principal Component Analysis (KPCA) was chosen due its more adequate output.

The dimensionality reduction step applied here does not only performs a feature extraction but also enable the visualization of the data in three-dimensional space. These key arguments for the usage of this technique are elicited in Mohri, Rostamizadeh and Talwalkar (2018).

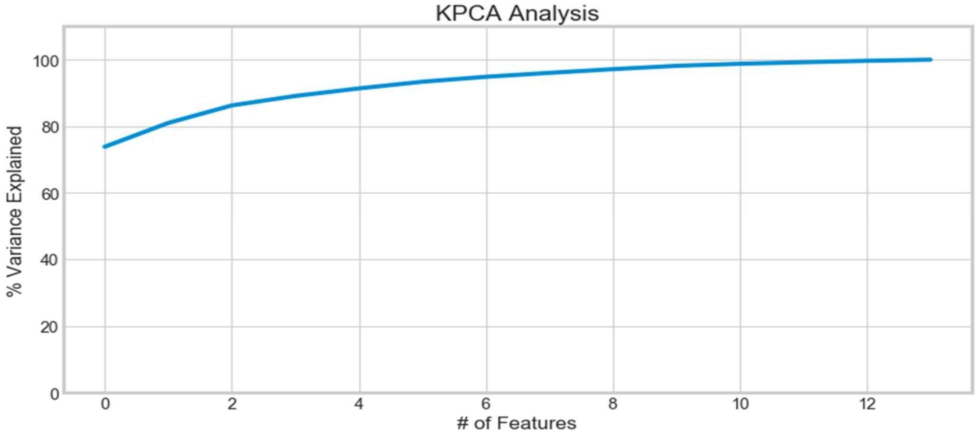
The Kernel Principal Component Analysis algorithm (KPCA) is applied to reduce the original dataset into a specified number of features called principal components. In the proposed process, the KPCA algorithm works as a preprocessing stage. In the early stages of the study, instead of KPCA, the simple PCA was used, however, as the process developed, the kernelized version algorithm delivered better results.

The implementation of the KPCA in this work was done by the Kernel-PCA method, part of the decomposition module from Scikit-learn library. This implementation of the KPCA requires only two hyperparameters as input, being , the gamma coefficient and a chosen kernel function. It is also required as input, the number of components to be used. This algorithm, in a high abstraction level, performs a non-linear dimensionality reduction through the use of kernels. Its definition was briefly described in 2.3.2.3 but may be found in Schölkopf *et al.* (1997) as well.

The choosing of the KPCA´s number of components, to be applied by the kernel function, and the gamma hyperparameters values were performed in a heuristic way, which can be considered a commonly used approach (ALAM, 2014).

To define how many components would be used to generate the output data set, the eigenvalues from the eigenvectors (principal components) were used as a data variance explaining measure (JOLLIFFE; CADIMA, 2016). In the first assessment execution of the algorithm, was generated a plot showing the relationship of number of eigenvectors, versus the absolute sum of the given eigenvalues, depicted as number (#) of features versus the percentage of variance explained, respectively, as shown in Figure 4.15.

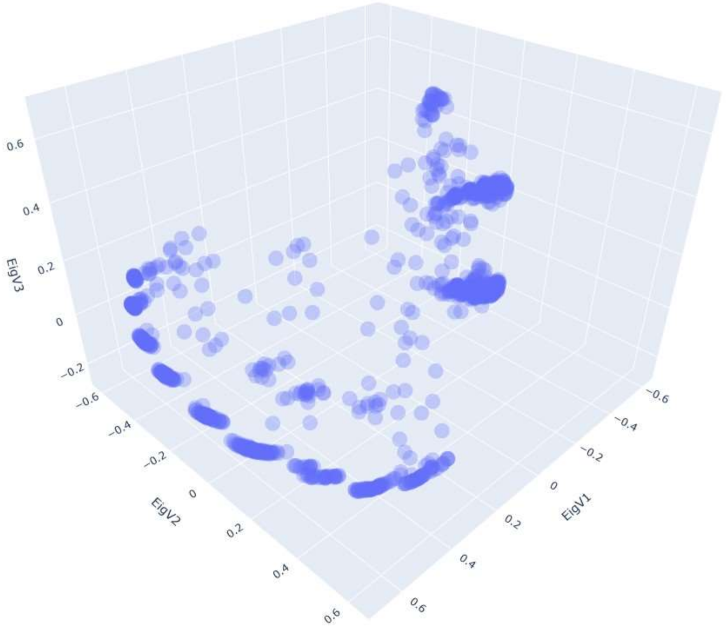
Figure 4.15 - Explained Variance percentage versus number of features used.



As observed, as the number of features (a.k.a. components) grow, the percentage of variance explained raises. The higher the percentages of explained variance better the predictions (ROSENTHAL;ROSENTHAL, 2011).Source: Author's production.

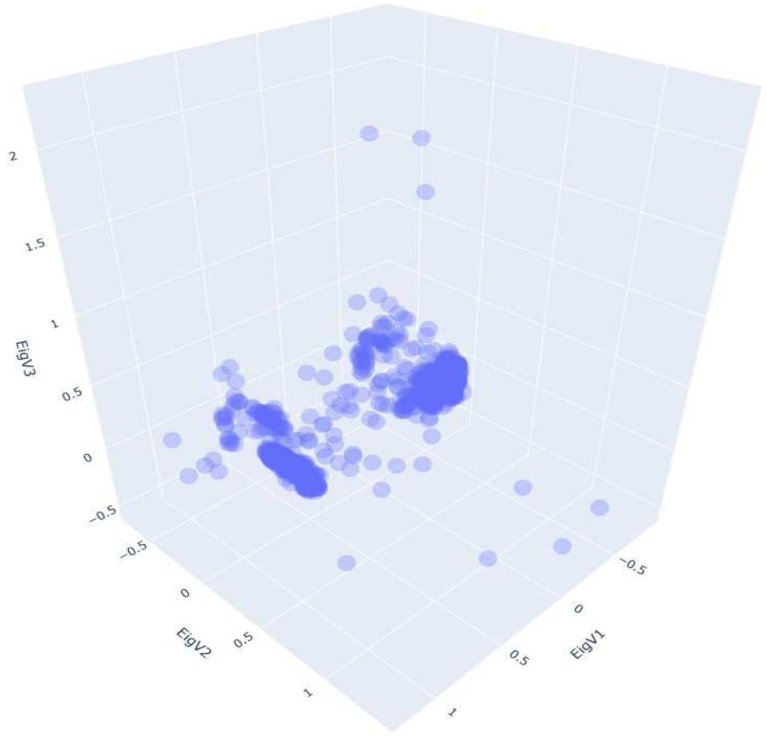
Although the task of defining the number of components which should be used by the KPCA algorithm is quite straightforward, the method to choose the kernel function and the gamma value was performed in an empirical way. Since the dimensionality reduction had a goal of generate a dataset that could be clustered, the kernel function and gamma value assessment were performed plotting the result of different permutation of kernel functions gamma value for the same number of components. This step took some time, as it is a manual fine-tuning step. Figure 4.16 illustrates an example of how the outputs of the process can vary, where the plots show the outcome of choosing the RBF kernel function. Figure 4.17 depicts the outcome of when the kernel function is Sigmoid. This step is repeated until that the desired output is obtained or the date present insights that can be understood as a representation of the satellite's circumstances during operation.

Figure 4.16 - KPCA output for RBF as the kernel function.



Source: Author's production.

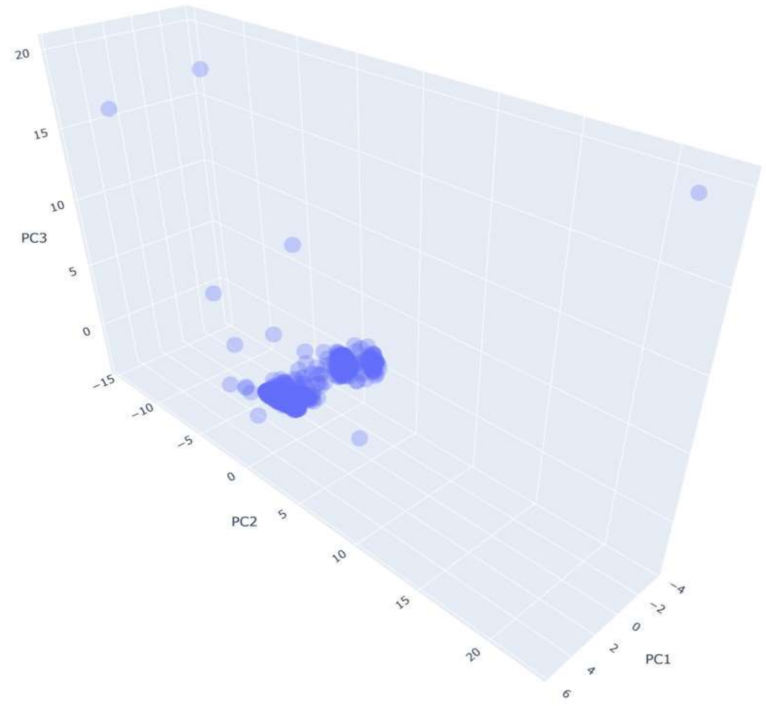
Figure 4.17 - KPCA output for Sigmoid as the kernel function.



Source: Author's production.

This study was conducted over the telemetries of the Power Supply Subsystem, so, it was expected that the data transmitted from it would depict a behavior or a tendency or a variance in the data resulted from the interaction of the parts and the elements surrounding it. This tendency or agglomeration was interpreted as an emerging attribute, or emerging behavior, which can be understand as defined by the system engineering, that the interaction of parts or subsystems always generates emerging attributes, that can only be achieved through this interaction and no other. Figure 4.18 shows the outcome of a PCA, ran in the early stages of this study. There, at first, two cluster are well defined and separated by a cloud of points. The two clusters were interpreted as two circumstances in which the satellite could be found. One when the satellite is receiving sunlight directly and another when the satellite is in the shadow and running with the energy stored in the batteries.

Figure 4.18 - Output of a PCA in the early stages of this study.



Source: Author's production.

The cloud of the data points, within the two agglomerations, are understood as a transition circumstance between the two major ones. There, can be also seen some sparse points that could have been turned to be anomalous behavior on the data.

4.2.3 Clustering

The clustering step is in charge of identifying the cluster to be used, or in other terms, it formalizes tendencies on the data set coming out from the dimensionality reduction step.

It is expected that the clustering algorithm be able to identify and define clusters that comprehend not only those agglomerated data, but also the sparse data. Since one of the main reasons to use clustering algorithm for anomaly detection on satellites systems comes from the fact that during the satellite's normal operation scenarios, the telemetry data tends to form clusters of characteristics (AZEVEDO et al., 2012).

It may be found in the literature, different algorithms being used for clustering purposes, applied in the space context, as K-Means, Expectation Maximizations in Ibrahim et al. (2019) and Azevedo et al. (2012) and Hierarchical Agglomerative Clustering in Mital et al. (2019). Each of them presenting pros, cons, and different performance depending on what are the characteristics of the data being analyzed and what is expected from their usage.

The Density-Based Spatial Clustering of Application with Noise, or simply DBSCAN, is a density-based clustering algorithm which needs minimal requirements of domain knowledge to determine the hyperparameter. It is capable of not only discover arbitrary-shaped clusters but also present good efficiency on large databases (ESTER et al., 1996).

The implementation used from the DBSCAN in this study was made possible through the DBSCAN class, part of the cluster module from the ScikitLearn library. This algorithm finds core samples of high density and expands clusters from them until it cannot classify points as part of a cluster. Its definition and formal explanation are given in Ester et al. (1996). A very educative and explanatory execution of the algorithm can be found in Harris (2015).

The DBSCAN, different from the K-Means, doesn´t require a definition of how many clusters one would like to have, instead, it is capable of provide how many clusters it can identify, followed by their identification, and also a singular cluster colleting those points which don´t fit any other cluster. This is performed through the hyperparameters epsilon (eps) and the minimum number of points in an eps-neighbourhood (minPts).

To find the most adequate values for both hyperparameters, two different methods can be used. The first one takes the outcome of the algorithm and evaluate it using a parameter called Silhouette Coefficient (SC) (KAUFMAN; ROUSSEEUW, 1990). The algorithm is executed with different values for epsilon, varying from 0.1 to 1.0 in steps of 0.1. For each epsilon increment step, the minimum number of points iterates from 2 to 8, and then it is tested. This procedure generates a list of different combinations resulting in a collection of number of found clusters and given SC pair, which can be evaluated using the rule of thumb shown in Table 4.2.

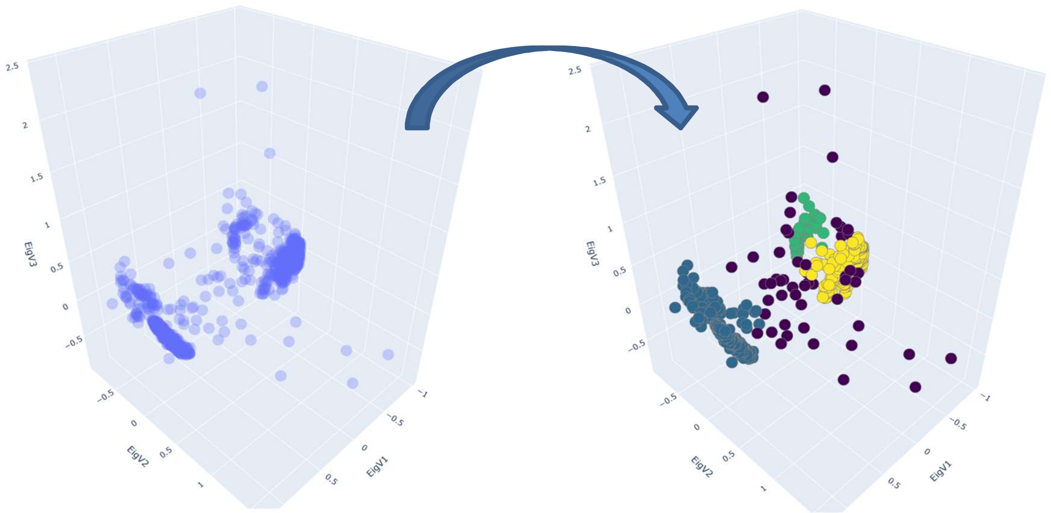
Table 4.2 - Subjective Interpretation of the Silhouette Coefficient (SC).

Silhouette Coeficient score	**Proposed Interpretation**
0.71 – 1.00	A strong structure has been found
0.51 – 0.70	A reasonable structure has been found
0.26 – 0.50	The structure is weak and could be artificial; try additional methods on this data set
≤ 0.25	No substantial structure has been found

Source: Kaufman (1990).

However, take only the SC to assess whether the fitted model fulfill the need to clusters identification is not enough. A heuristic evaluation had to be done with those hyperparameters which resulted in 5 clusters and a SC reasonable or better. This evaluation is made through a plot combining the dataset coming from the KPCA with a coloring scheme to represent the found different cluster, as presented in Figure 4.19.

Figure 4.19 -KPCA output data set before and after clustering.

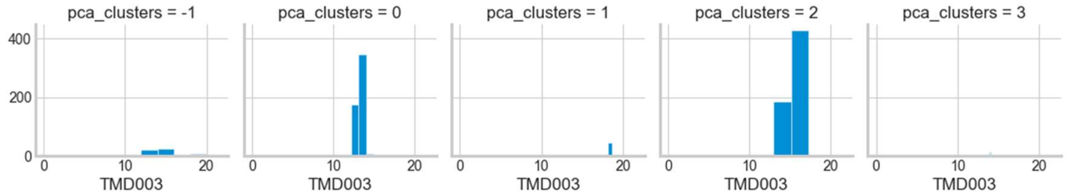


This plot shows the different clusters found by the DBSCAN algorithm. Each cluster has a representing color. The points in dark purple are part of no cluster. The visualization process allowed the assessment whether the found solution would fit or not the needs.

Source: Author's production.

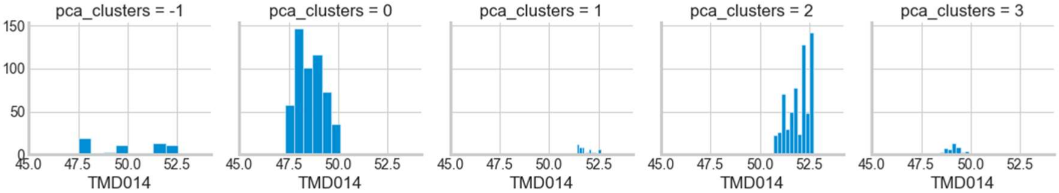
In the first method, after the hyperparameters being defined, the algorithm was executed once more and from the model fitted. Through an attribute called labels, it was possible to perform the retrieving of an array of integers labels corresponding to the different clusters. This array then is appended to the original data frame as a column of labels, designating a label to each observation of a group of telemetries. The relationship between each telemetry and the found clusters is plotted through a series of histograms. From the samples depicted from Figure 4.20 to Figure 4.23 it is possible to drawn conclusions on which cluster represents which satellite's circumstances, and how the clusters are distributed among the telemetries observations.

Figure 4.20 – Distribution of labels in the telemetry 003 – MEAS Output Voltage.



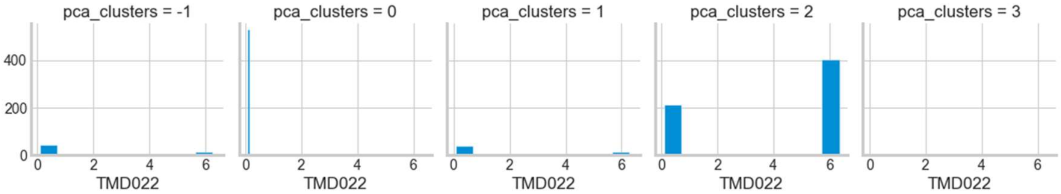
Source: Author's production.

Figure4.21 -- Distribution of labels in the telemetry 0014 – Battery Voltage.



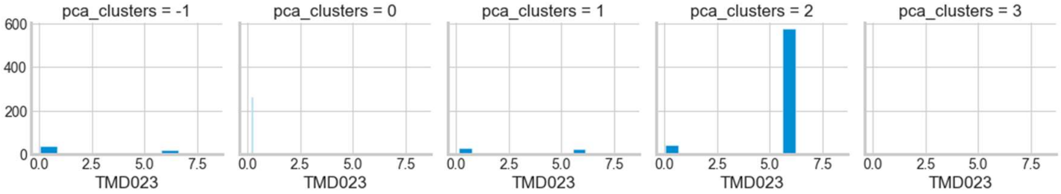
Source: Author's production.

Figure4.22 - Distribution of labels in the telemetry 0022 – Solar Panel Current.



Source: Author's production.

Figure 4.23 - Distribution of labels in the telemetry 0023 - Solar Panel Current.



Source: Author's production.

The TMD0023 value represents one of the solar panels output current, thus, it is possible to infer that cluster #2 is related to the circumstance of the satellite when sun is sight, on the other hand, cluster #0 represent the satellite in eclipse, while cluster #3 and #1 are related to the twilight situation and the clusters #-1 refers to the anomalous behavior points.

Till this point, only unsupervised learning was used due to the lack of labels on the dataset, although, throw the clustering step, the data observations could be separated in different operational scenarios and in one anomalous-behavior scenario. This categorization, in form of clusters, and furthermore as labeling dimension on the dataset, provides information for the next step of the proposed method.

The second method is presented in the paper (ESTER et al., 1996). In the Section 4.2 of that paper "*Determining the parameters eps and MinPts*” an adequate value for eps can be found by calculating the distance to the nearest K points for each point, sorting and plotting the results in order to asse where a point of inflection happens, or in other terms, the "elbow" or "knee" point (RAHMAH; SITANGGANG, 2016). That point will have the adequate value of eps to be taken. Additionally, the authors add and thumb rule to define the value of the K, being this equal to two times the amount of dimension of the input data, i.e., for inputs of two dimensions, K shall be set to 4. The calculation of the exact value of eps on the point of inflection was performed through the method described in Satopaa et al. (2011), available in the "kneed" open-source library.

4.2.4 Classification

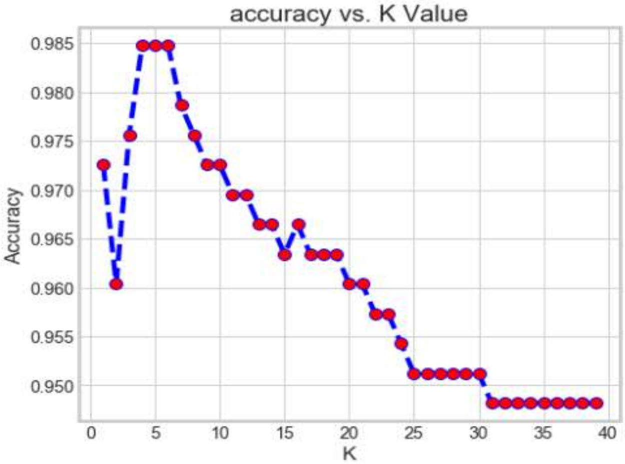
Among the classification-based algorithms found in the literature, as Support Vector Machine and Neural Network, the K-Nearest Neighbors was chosen to perform this Classification step. Being considered as a supervised learning algorithm since it requires labeled data to work, it is simple to understand, and easy to use.

The usage of the KNN algorithm in this study was, as the previous stages of this process, enabled by an implementation of it on ScikitLean. Through the class KNeighnorsClassifier, part of the neighbors module, the generation of classification model can be achieved. The implementation requires as input only the number of neighbors to be use as the ruler, thus, other parameters can be modified as the metric parameters, which is set to Minkowski Distance by default. It can also be modified to Euclidean or Manhattan, among other distance calculation methods. In this study, iterations of the KNN using the Manhattan and Euclidian distance metrics were performed.

The algorithm was, initially, executed with an arbitrary value for number of neighbors, six. Usually at this point, the KNN algorithm model is trained with a training data set, and then the model is evaluated against a testing data set. At the first iteration the telemetry data set was divided into two parts, being the first one corresponding to 75% of the data set, and used for training the model, and the rest to be used for validation. However, to determine a proper value for the number of neighbors, an empirical method together with a defined metric was applied. The method consists in training the algorithm and test it with different number of neighbors within a range of 1 to 40 neighbor points.

With the outcome of these iterations in hand, using the Accuracy function available in the ScikiLearn library, we can calculate the accuracy of the trained model. These accuracy values are obtained when comparing the classified data versus the real labels on the data frame. The outcome of this heuristic method was then used to create a plot that represents the accuracy of the model versus the value defined for k. The best accuracy value achieved, through a k value bigger than 1 is chosen. The outcome of this process is shown in Figure 4.24.

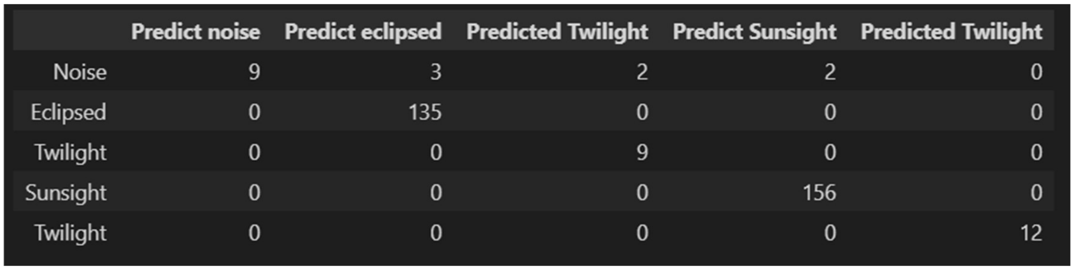
Figure 4.24 - Model accuracy versus the number of neighbors.



Source: Author's production.

After choosing the most adequate value for K, based on the highest accuracy value, the algorithm is executed, and a new model is trained. The model classification result then is assessed with the support of a confusion matrix. The confusion matrix is often applied to evaluate the result of classification algorithm. Confusion matrix depicts how confuse is the classification model when comes to its predictions (BROWNLEE, 2016), an example is shown in Figure 4.25.

Figure 4.25 - Confusion Matrix for a tryout of the KNN algorithm.



	Predict noise	Predict eclipsed	Predicted Twilight	Predict Sunsight	Predicted Twilight
Noise	9	3	2	2	0
Eclipsed	0	135	0	0	0
Twilight	0	0	9	0	0
Sunsight	0	0	0	156	0
Twilight	0	0	0	0	12

Confusion matrix of an early phase of the study. The label noise turned to be what is considered anomalous behavior on the data.
Source: Author's production.

In the confusion matrix depicted in Figure 4.25, using a K value of 5, the outcome of the KNN classifier algorithm to classify the data points, as one of the five different scenarios, can be clearly visualized, i.e., there were three times the model classified a point as Eclipsed scenario but in truth those points were noise.

The usage of such approach to depict the result of the classification model was made in order to have a different insight about how good the model is when comes to detect anomalous behavior despite its general accuracy.

5 ANALYSIS AND DISCUSSION

This Chapter discusses the results with 3 experiments, namely, CBERS1-Part1, CBERS1-Part2 and SCD2, which were ran over the dataset of the two satellites: CBERS1 and SCD2.

The telemetry data of the CBERS1 satellite was used in two experiments (CBERS1-Part1 case study and CBERS1-Part2 case study). At the first experiment all the effort and time spent was made in order to obtain knowledge not just about the inherent aspects of such application domain, but also to ramp up the knowledge regarding how to approach anomaly detection task making usage of off-the-shelf machine learning solutions. The CBERS1-Part1 case study outcome is presented in the Section 5.1.

The second experiment, the CBERS1-Part2 case study, was conducted in order to solve some limitations found in the machine learning process proposed, which were detected in the first experiment. Moreover, this experiment validated the anomalous behavior detection capabilities of the proposed process. The outcome of the second experiment is detailed on Section 5.2.

The third experiment was conducted with the telemetry data of the SCD2 satellite. Some problems arose in the handle of the data mass during the execution of this experiment did not allow the complete planed experiment, however some lessons were learned. These problems faced with the telemetry data of the SCD2 satellite are discussed in Section 5.3.

5.1 CBERS1- Part 1 case study

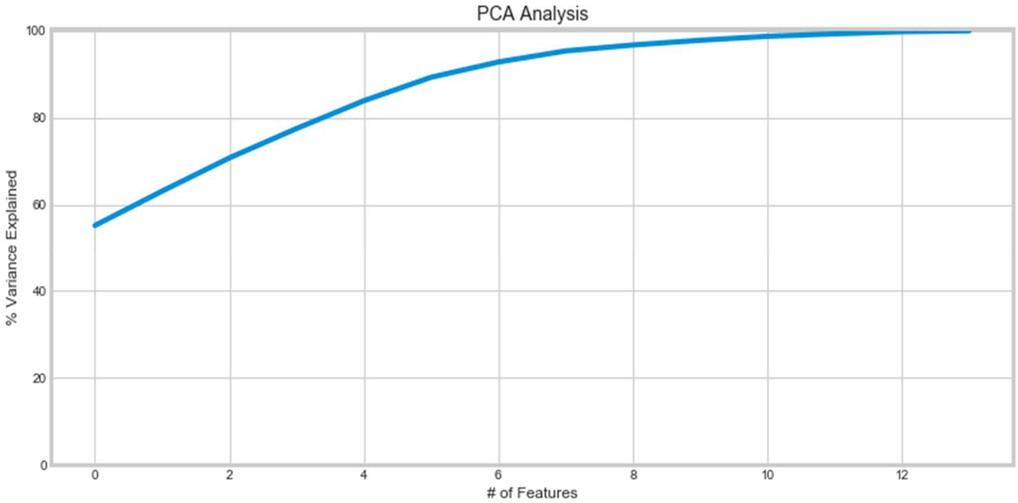
This session presents the first steps and analysis of the CBERS1 telemetry dataset and discusses the outcomes of such. The first part of the study case had the purpose of raise insights about the problem domain and about the machine learning algorithms that could be used for the steps described in the Section 4.2, it served as a feasibility study performed before performing the part 2 that is described in the Section 5.2.

5.1.1 A brief analysis with the use of PCA at first

Originally the Dimension Reduction step to be applied over the telemetry dataset was supposed to be performed making usage of the PCA algorithm. The dimensionality reduction had not just a pre-processing objective in this part of the case study, but also to be an enabler for visualizing the data before performing the clustering. For the sake of completeness, even though the KPCA turned to be the algorithm used for this purpose, the results coming from the first analysis made with the PCA are exposed in the following.

The outcome of the principal component analysis performed over the data coming from January of 2000, depicted by the Figure 5.1, demonstrates the amount of data variance in percentage can be explained by a given number of principal components (features) extracted from the data used as input.

Figure 5.1 - Data variance explained versus number of features.

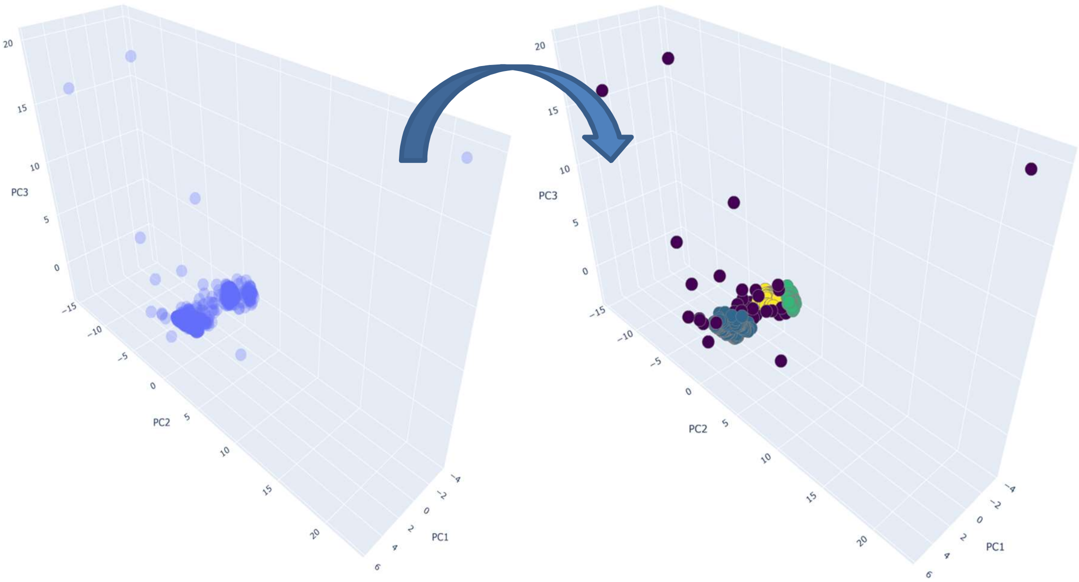


Source: Author's production.

The first principal component can explain almost 60% of the variance from the original dataset. The choice of having 3 principal components extracted from the original dataset would give an amount of 77.52% of the data variance represented, and at the same time, to enable the visualization of the new reduced data set in a 3 dimensional perspective, as depicted in the Figure 5.2 by the plot on the left side.

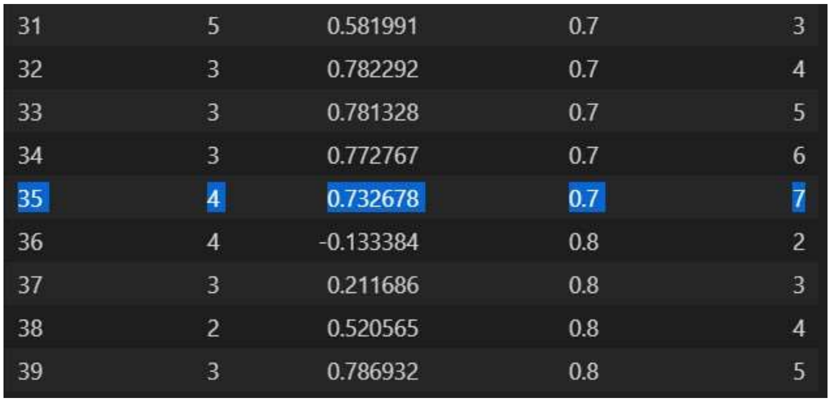
The plot on the right side is depicting the outcome of the clustering process performed with the DBSCAN algorithm. The clustering was executed having the hyperparameters values of epsilon and minPts set to 0.7 and 7 respectively. These values were obtained through a small experiment in which Eps values ranging from 0.2 to 1.5, which steps of 0.1, combined with minPts values ranging from 2 to 8, a table as the one depicted in Figure 5.3 was generated.

Figure 5.2 - New PCA data set clustering outcome.



Source: Author's production.

Figure 5.3 - Eps and minPts combination assessed by silhoute score.

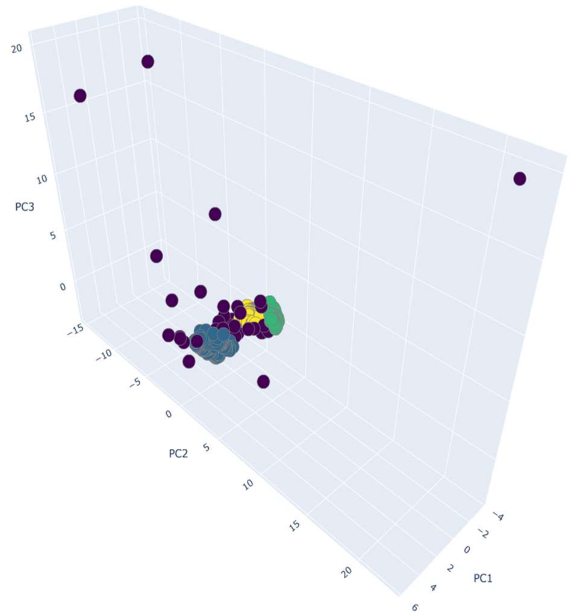


31	5	0.581991	0.7	3
32	3	0.782292	0.7	4
33	3	0.781328	0.7	5
34	3	0.772767	0.7	6
35	4	0.732678	0.7	7
36	4	-0.133384	0.8	2
37	3	0.211686	0.8	3
38	2	0.520565	0.8	4
39	3	0.786932	0.8	5

Source: Author's production.

The heuristic assessment made in this context was driven by finding a small eps value which, combined to a minimum number of points equal or bigger than 6, would result in a good enough silhouette score. The combination chosen, shown in Figure 5.3, is an eps value of 0.7 with a minimum number of points of 7, which resulted in 4 clusters, one of them containing the outliers. The result is presented in Figure 5.4. This result was considered good enough since through a visual assessment the clustering result seemed to present a better definition of cluster when comes to homogeneity and number, representing in a high level different main circumstances in which the satellite could be found. This can be seen from the manner in which the data is grouped and therefore clustered. These different scenarios can be understood as one for when the satellite is being hit by sunlight, another one when in the shadow, and a third one representing the twilight condition. All the other points left were outliers assigned to the "noise clusters". The definition of which cluster is representing which scenario was made through the assessment of the distribution of observation among the identified clusters.

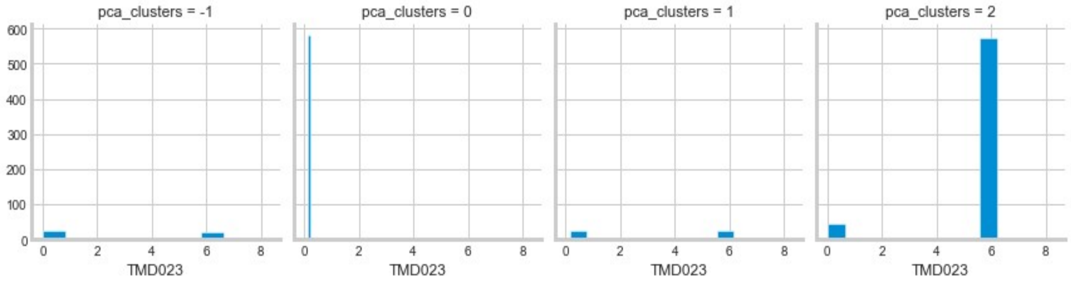
Figure 5.4 - Clustering outcome with hyperparameters set by heuristic evaluation.



Source: Author's production.

Figure 5.5 depicts the distribution of data points from the TMD023. These TM data correspond to current values in one of the solar array outputs. The data distribution indicates that the pca_cluster = 2 gathers the points related to the "hit by sunlight" scenario, and the pca_clusters = 0 gathers the points from the "in the shadow" scenario, while the pca_cluster = 1 holds the points for the "twilight" scenario, the pca_cluster = -1, as mentioned before, is the one which the outliers are assigned to belong.

Figure 5.5 - Data points of the TMD023 distribution among the found clusters.



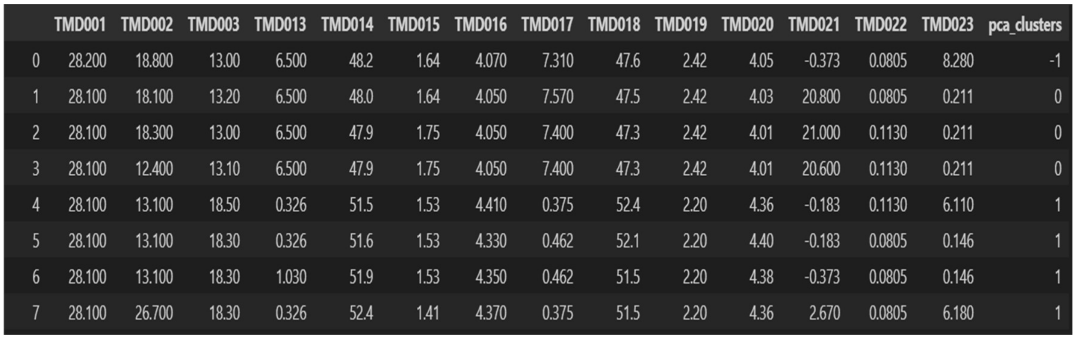
Source: Author's production.

These statements were made over the following assumptions:

- Cluster = -1: noise or outliers, because this is how the algorithm were implemented;
- Cluster = 0: Sun eclipsed, because this cluster holds most of the values for output current near to zero;
- Cluster = 1: Twilight, because this cluster holds values from both situations, sunlight and eclipse;
- Cluster = 2: Sun sight, because this cluster holds the most of the values for maximum output current.

The found clusters, given back by the DBSCAN implementation as an array of indexes were concatenated to the original data set (from January 2000), this way making the cluster indexes as labels for each observation in the data set, as shown Figure 5.6.

Figure 5.6 – Data frame having the clusters indexes as labels.



	TMD001	TMD002	TMD003	TMD013	TMD014	TMD015	TMD016	TMD017	TMD018	TMD019	TMD020	TMD021	TMD022	TMD023	pca_clusters
0	28.200	18.800	13.00	6.500	48.2	1.64	4.070	7.310	47.6	2.42	4.05	-0.373	0.0805	8.280	-1
1	28.100	18.100	13.20	6.500	48.0	1.64	4.050	7.570	47.5	2.42	4.03	20.800	0.0805	0.211	0
2	28.100	18.300	13.00	6.500	47.9	1.75	4.050	7.400	47.3	2.42	4.01	21.000	0.1130	0.211	0
3	28.100	12.400	13.10	6.500	47.9	1.75	4.050	7.400	47.3	2.42	4.01	20.600	0.1130	0.211	0
4	28.100	13.100	18.50	0.326	51.5	1.53	4.410	0.375	52.4	2.20	4.36	-0.183	0.1130	6.110	1
5	28.100	13.100	18.30	0.326	51.6	1.53	4.330	0.462	52.1	2.20	4.40	-0.183	0.0805	0.146	1
6	28.100	13.100	18.30	1.030	51.9	1.53	4.350	0.462	51.5	2.20	4.38	-0.373	0.0805	0.146	1
7	28.100	26.700	18.30	0.326	52.4	1.41	4.370	0.375	51.5	2.20	4.36	2.670	0.0805	6.180	1

Source: Author's production.

The given data set then was divided between a training and a testing data sets. The training set was used for training the KNN model using a k hyperparameter value of 6, following the same thumb rule used for calculating the minPts value. The KNN fitted model was tested using the testing data set and its result assessed with the help of the confusion matrix depicted by Table 5.1.

Table 5.1 - Confusion Matrix for the data set preprocessed by PCA.

	Predict Outlier	**Predict Eclipsed**	**Predicted Twilight**	**Predict Sunsight**
Outlier	7	4	1	2
Eclipsed	0	149	0	0
Twilight	0	0	8	0
Sunsight	0	0	2	155

Source: Author's production.

With the confusion matrix in hand, in order to calculate the accuracy of the generated model, the correct predicted values presented on the main diagonal are summed and then divided by the sum of all other elements. The obtained accuracy was 0.973 from a maximum score of 1.0. In a first moment the value seems pretty good; however, the purpose of this work is to be able to detect anomalous behavior, which, in the Table 5.1, is called as outlier. When calculating the outliers detection accuracy the found value was 0.5 from 1.0. As one can see, when evaluating only the noise accuracy of the model, it is possible to realize that in fact the model maybe not so good. The same assessment of KNN model was made with a data set containing not 3 but 5 feature, in the sense that it could generate a more accurate model, the values obtained for it were not better, being the total accuracy of 0.963, and the outlier detection accuracy of 0.368. This is summarized in the Table 5.2.

Table 5.2 - KNN model accuracy for different feature number-wise data sets.

Number of features	**Total accuracy (%)**	**Outlier detection acc.(5)**
3	97.3	50.0
5	96.3	36.8

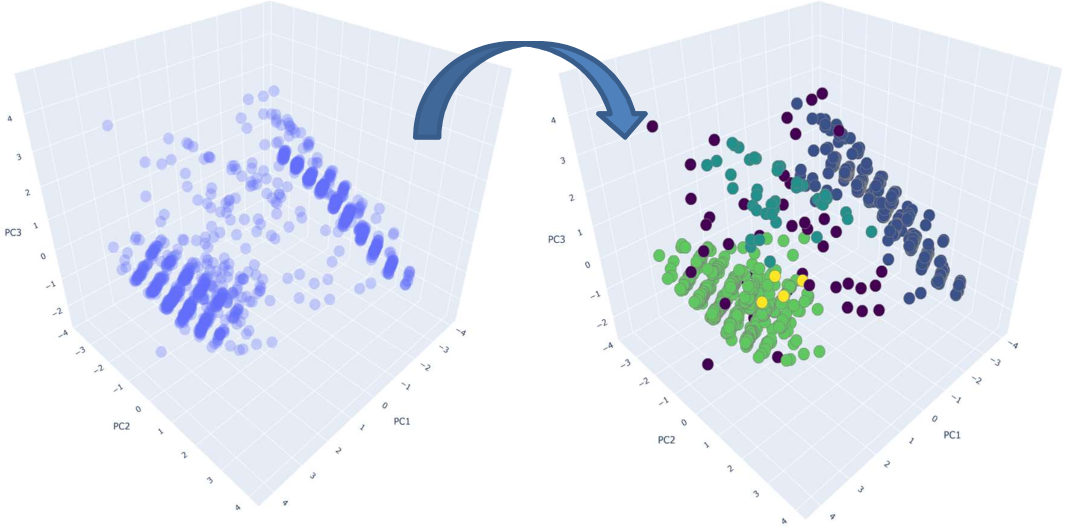
Source: Author's production

It is clear that at this point the performance of the classification model is not good enough for anomalous behavior detection. From this first result, some considerations were raised in order to assess a way to increase the anomaly detection ability of the proposed process.

This study proposes a mixture model approach which uses a dimensionality reduction algorithm to preprocess the data before use a clustering algorithm. The clustering algorithm used here is the DBSCAN, chosen mainly due its ability to deal with noise data. This "noise resistant" characteristic from the clustering algorithm would be enough to say that the trivial outliers removal techniques are not needed; however, this is true only if the dataset would be given to the clustering algorithm directly, in others words, if the data were not being preprocessed with a dimensionality reduction algorithm previously. This conclusion is confirmed with what is proposed by (Yairi et. al. 2017) and (LI et al., 2018). In face of that, the trivial outlier removal step, detailed in the 4.2.1.3 was performed in the data set before performing the dimensionality reduction. The clustering step was once more performed, and the outcome is depicted

Figure 5.7.

Figure 5.7 - Trivial outlier cleaned data set after clustering.



Source: Author's production.

After applying the trivial outlier’s removal process the data set obtained from the PCA algorithm presented a worse outcome. The result was considered to be not adequate because:

- The feature reduced data set do not present clearly separable data;
- As shown in
- Figure **5.7**, the number of clusters found didn't meet the assumption taken as requirement of 4 well-defined clusters interpreted as a representation the satellite's behavior due different circumstance.

The result obtained at this step showed that the PCA algorithm was not adequate for performing dimensionality reduction over the given data set. This can be explained by the non-linear nature of the data used in this study. Unfortunately, such characteristic cannot be handled by the classic PCA.

5.1.2 Data preparation using KPCA+DBSCAN+KNN

During the experiments performed in the Part 1, different Kernel functions were tested in order to raise the one which associated with the principal component analysis would perform better. Experiments were made using the linear kernel, radial basis function (rbf) kernel and the sigmoid kernel. For each function a visual assessment was made through the plot of the output data, in order to, from “trying and testing” steps (see Figure 4.7) to define a value for the hyperparameter gamma. The percentage of variance, explained by the number of eigenvectors composing the output of the KPCA data set, was assessed for the three different kernels experimented. The results are shown in Table 5.3.

Table 5.3 – Variance explained by the number of eigenvectors composing the output.

Kernel function	Number of eigenvectors	Variance explained (%)
Linear	3	93.15022787345725
	5	96.16924194850708
RBF	3	71.06849793183024
	5	82.9616942975014
Sigmoid	3	92.05005436958172
	5	94.81223076601363

Source: Author's production.

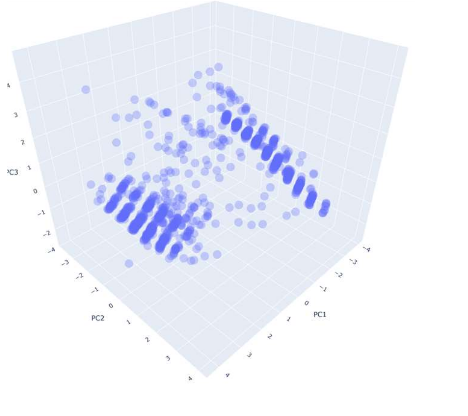
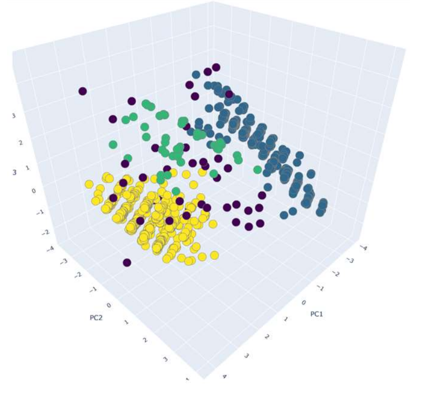
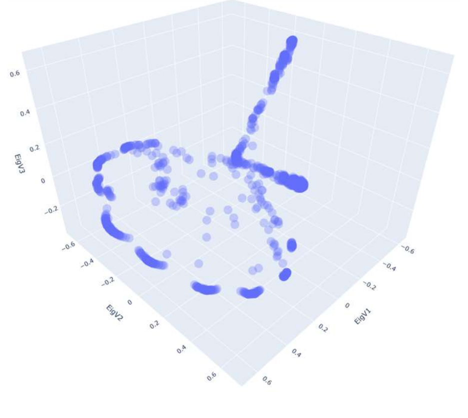
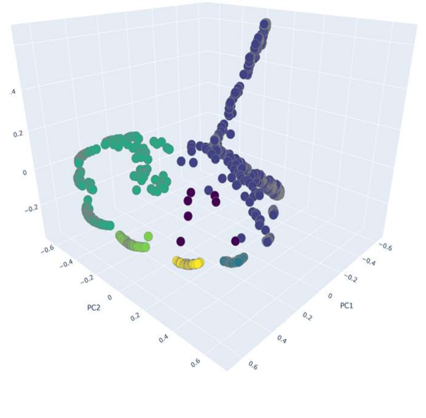
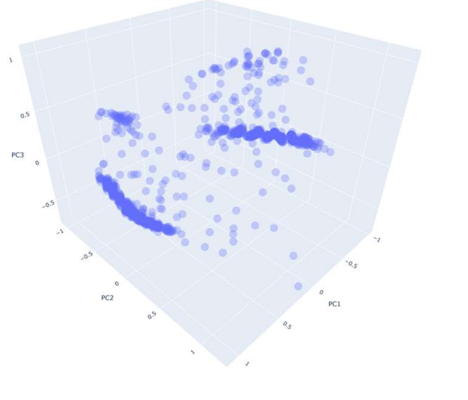
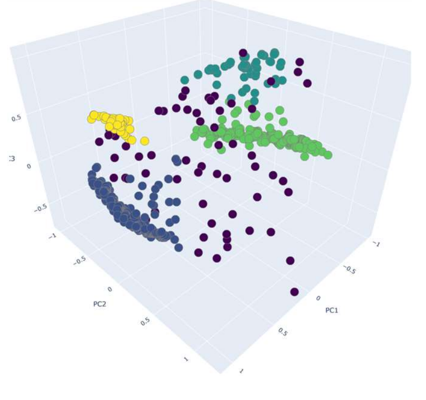
The usage of three components of the KPCA already produces a very good variance value, and as the data will be plot for visualization, chose three components among the other values turns to be the most adequate option. These outcomes from the KPCA were obtained setting the hyperparameters gamma to a value equal to 0.2.

For every different kernel experiment, the 3 dimensions visualization of the KPCA output data set was made as the outcome from the clustering performed by the DBSCAN. The clustering algorithm hyperparameters were set for the different kernel input data sets accordingly:

- Linear kernel-reduced data set as input – eps = 9 and minPts = 7;
- RBF kernel-reduced data set as input – eps = 0.1 and mintPts = 7;
- Sigmoid kernel-reduced data set as input – eps = 0.2 and mintPts = 7.

The visual analysis of the outcome from the clustering process, which was made over the plots depicted in Table 5.4, had the purpose of identify among the results which one would present the adequate characteristic of well separated and identifiable cluster that could represent the satellite's behavior in different circumstances.

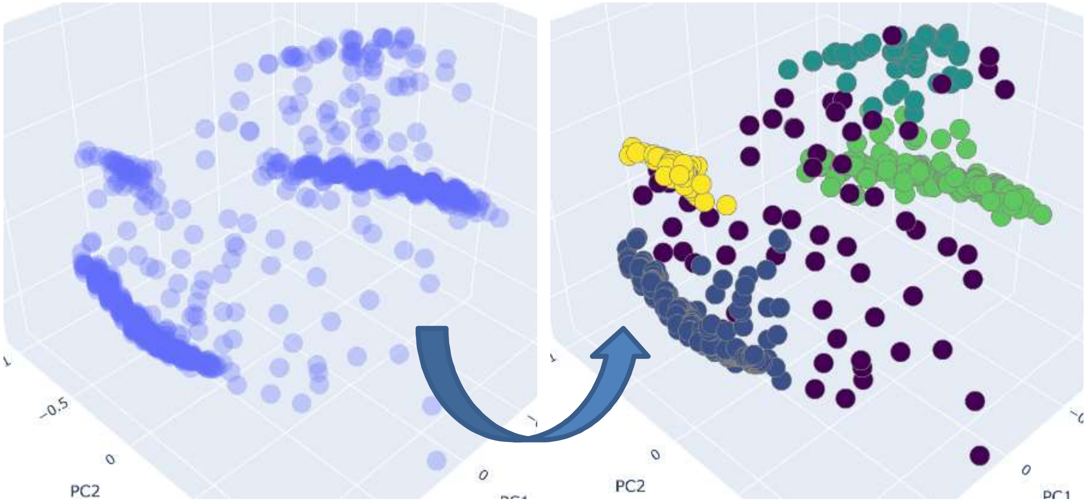
Table 5.4 – KPCA output before and after DBSCAN clustering process.

	KPCA output	**DBSCAN output**
LINEAR KERNEL		
RBF KERNEL		
SIGMOID KERNEL		

Source: Author's production.

The outcome dataset of the KPCA, showed in more detail in Figure 5.8, run a sigmoid kernel with the values of eigenvector and gamma hyperparameters of 3 and 0.2 respectively. They were used as input for a DBSCAN with epsilon equals to 0.2 and a minimum number of samples of 7, provided a silhouette score of 0.516, with 5 clusters identified. However, it is important to draw the attention that, after the trivial outliers were removed from the initial dataset, a new tunning of the KPCA hyperparameters had to be performed. In this case, the gamma had to be improved, and the hyperparameter “fit_inverse_transform” had to be assigned, which learn the inverse transform for non-precomputed kernels.

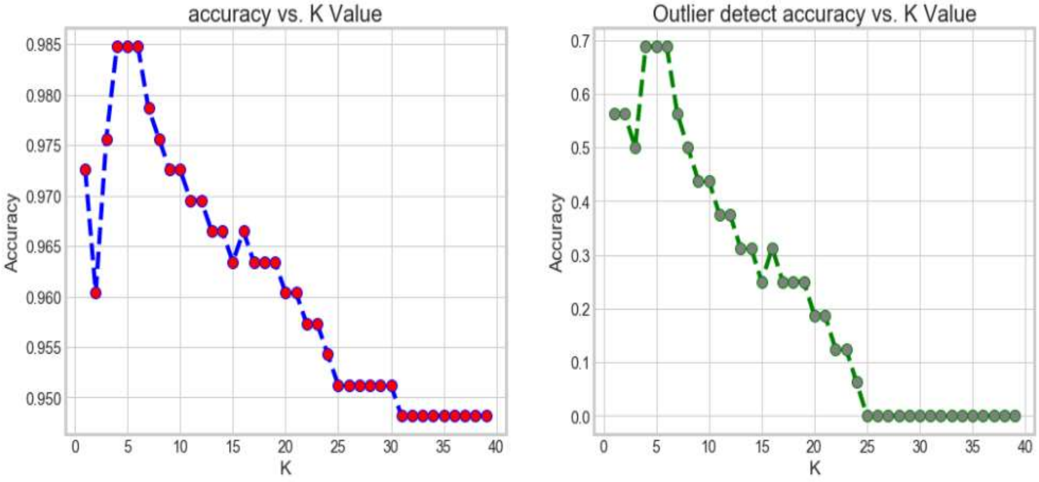
Figure 5.8 - Sigmoid KPCA output clustered with a trivial outlier cleared dataset.



Source: Author's production.

Different from the previous experiment, here, the outcome from both algorithms was quite satisfactory with 4 well defined clusters and a cloud of sparse outliers assigned to the -1 cluster. This result is adequate, and the indexes of the cluster can be used as labels to train a model with the kNN algorithm. The process to choose the k parameters was performed having the above result as input for the kNN algorithm. The result is shown in Figure 5.9. The best outlier detection accuracy value achieved was 0.6875 using K equals 5. After training the model, an evaluation was run over a test dataset, the outcome of this test is shown in Table 5.5.

Figure 5.9 - KNN model accuracy versus number of k.



Source: Author's production.

Table 5.5 - Confusion matrix for results obtained from KNN.

	Predicted noise	**Predict eclipsed**	**Predicted twilight**	**Predicted sun sight**	**Predicted twilight**
Noise	11	2	2	1	0
Eclipsed	0	135	0	0	0
Twilight	0	0	9	0	0
Sun sight	0	0	0	156	0
Twilight	0	0	0	0	12

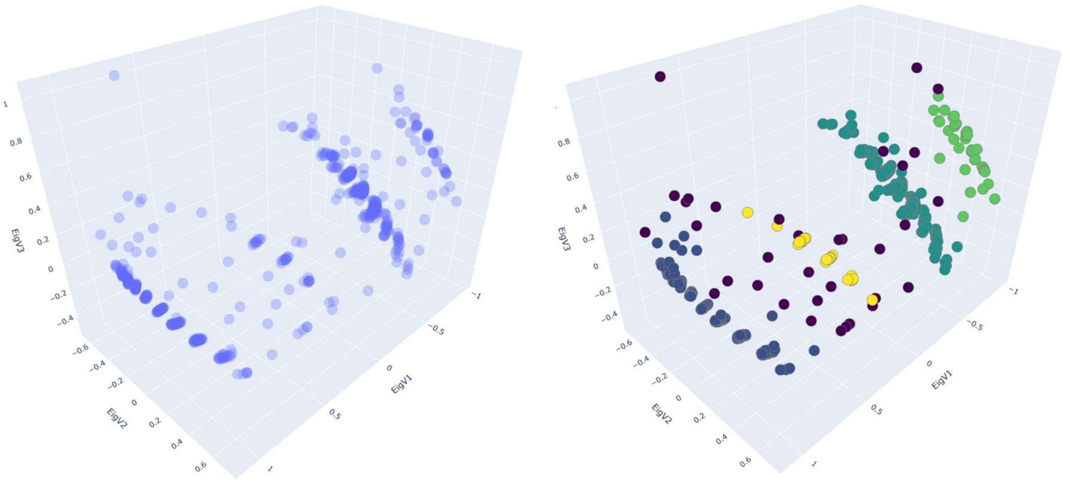
Source: Author's production.

The usage of the trivial outliers removal technique allied to the KPCA algorithm presented better result when come to anomalous behavior detection. Even though the KPCA output got improved by the removal of trivial outliers, the KPCA dataset generated at the end of the process presented clusters with differences in shape and density, and due to that, the work of the DBSCAN algorithm got difficult. However, we were able to identify clusters with a silhouette score of 0.52. Not the best, but good enough. Once more the general accuracy was a quite satisfactory value, around 0,979.

Among the kernel functions implemented in the KPCA sci-kit learn, the one which provided a more meaningful output was the kernel function sigmoid. Among the distance metrics implemented in the sci-kit learn for the KNN algorithm, two were tested, Euclidian and Manhattan. The Euclidian obtained a better accuracy when comes to anomalous behavior detection in the test model. Only the Ecludian-metric related results are presented here.

At this point, having a trained model presenting good results during the testing step, the only left step is to validate the model using a totally new and unseen data set. In order to perform the validation, the data set had to be firstly processed, then reduced, and finally clustered, this way, labels were identified for every observation on the given data. This process was conducted having the same setting of hyperparameters for the KPCA and DBSCAN. The outcome of it was plotted and is illustrated in Figure 5.10.

Figure 5.10 - KPCA-reduced validation data set before and after clustering.



Source: Author's production.

Having a labeled data set to be used as a base of comparison, the unlabeled version of the same validation data set was given as input to the KNN model obtained earlier and with the output of the KNN's classification another confusion matrix was built, which is depicted in Table 5.6. The calculated accuracy for general classification (considered in all the cases) was 0.44, meaning that the model misclassified more than the half of the observations from the data set, when compared to the label of the given observations. When comes to the anomaly (depicted as noise), the model was not capable of make an accurate classification.

Table 5.6 - Confusion matrix for results obtained from KNN predict over the validation data set.

	Predicted noise	**Predict eclipsed**	**Predicted twilight**	**Predicted sun sight**	**Predicted twilight**
Noise	0	14	3	10	8
Eclipsed	0	247	0	0	3
Twilight	0	0	0	266	0
Sunsight	0	1	45	0	0
Twilight	0	0	0	0	28

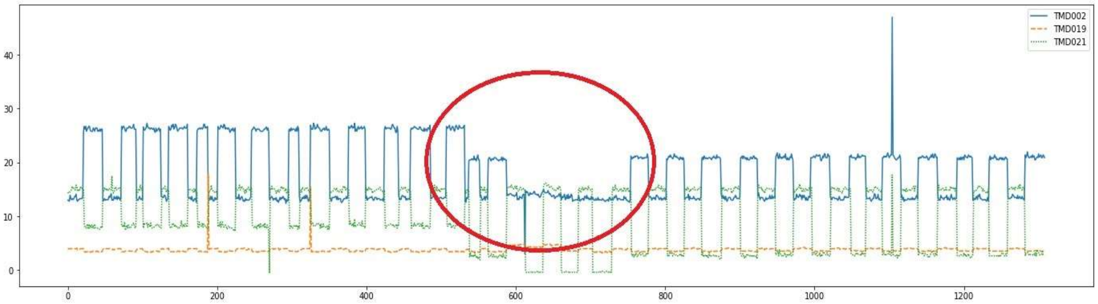
Source: Author's production.

5.2 CBERS1-Part 2 case study

5.2.1 Data preparation using KPCA+DBSCAN

The CBERS1-Part2 Case study had the goal of improve the accuracy of the anomaly detection as the accuracy of other behaviors in general, and also aimed to evaluate the capability of detecting an anomalous behavior in the telemetry data using one real case detected in CBERS1 operation. The telemetry signals from the month of June, as depicted in Figure 5.11, have anomalous behavior noticed more clearly over the TMD002 and TMD0019.

Figure 5.11 - Anomalous behavior present in the telemetry signals from the PSS during the month of June.

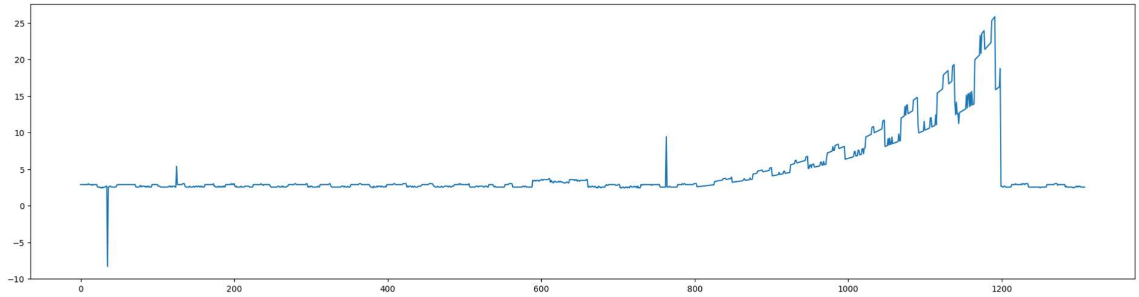


Source: Author's production.

It is known that between 15/06/2003 and 19/06/2003 the value of the TMD002 kept flat, within the expected limits, but out of the expected profile (behavior). The TMD0021 on the other hand, kept the profile but, by the same amount of time, had a negative offset which shifted the values out of the expected level.

Another anomalous behavior will be simulated according to the situation presented in (AZEVEDO et al., 2012). The simulated anomalous behavior concerns the TMD019 and follow the trend depicted in the Figure 5.12.

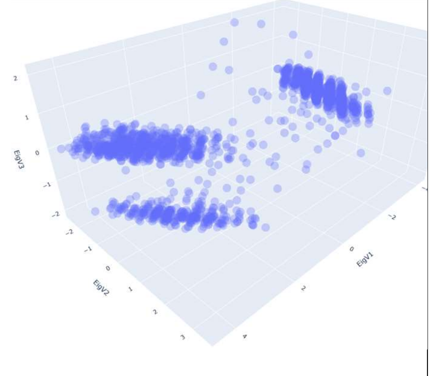
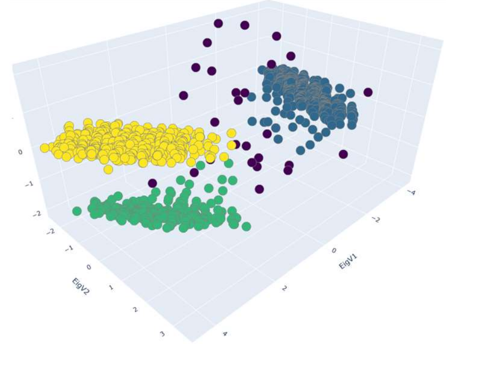
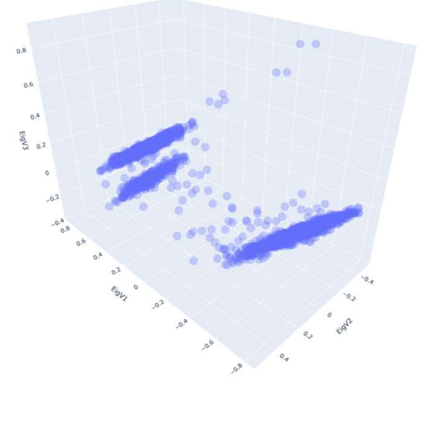
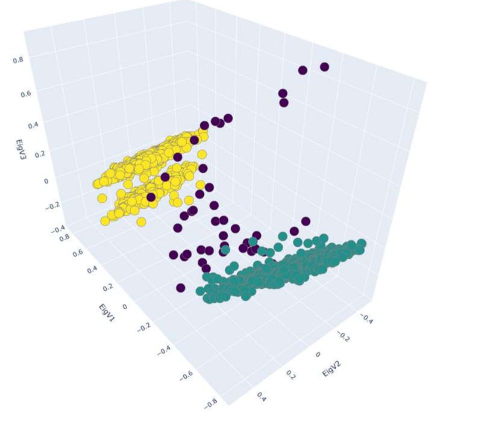
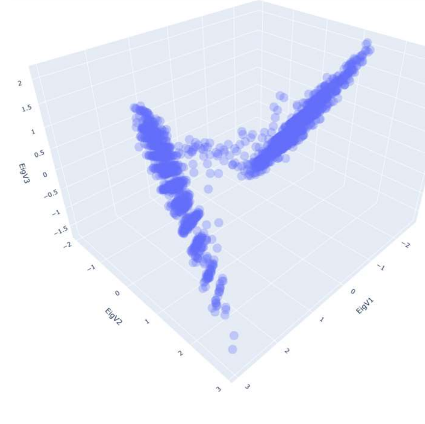
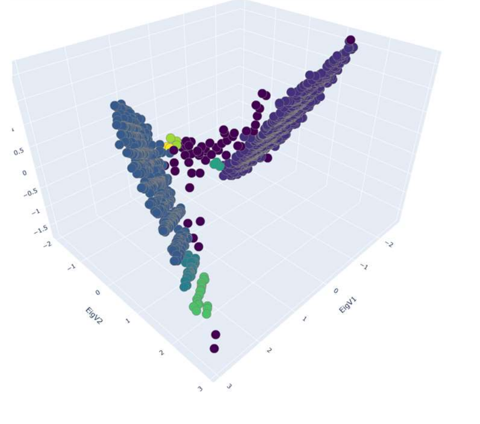
Figure 5.12 - Simulated anomalous behavior over one of the battery temperature measure telemetry (TMD019).



Source: Author's production.

In this step, as in the previous one, both data set were cleaned from trivial outlier, and normalized. The kernel functions assessed were the linear, sigmoid and poly. The outcome from such experiments is depicted in Table 5.7, where the result of the clustering can also be observed. The result presented was obtained from the processing of training data set.

Table 5.7 – KPCA output before and after DBSCAN clustering process for the CBERS1-Part2 case study.

	KPCA output	**DBSCAN output**
LINEAR KERNEL		
SIGMOID KERNEL		
POLY KERNEL		

Source: Author's production.

The hyperparameters settings made for the KPCA algorithm, apart from the kernel selection, was the number of feature in the output data set, making the gamma being equal to 1/N, where N is the number of features of the input data (default value from the algorithm implementation). The epsilon and minPts hyperparameters for the clustering algorithm were calculated using the "elbow method" (RAHMAH; SITANGGANG, 2016) together with the thumb rule for minPts definition. The values calculated for the eps are presented in Table 5.8.

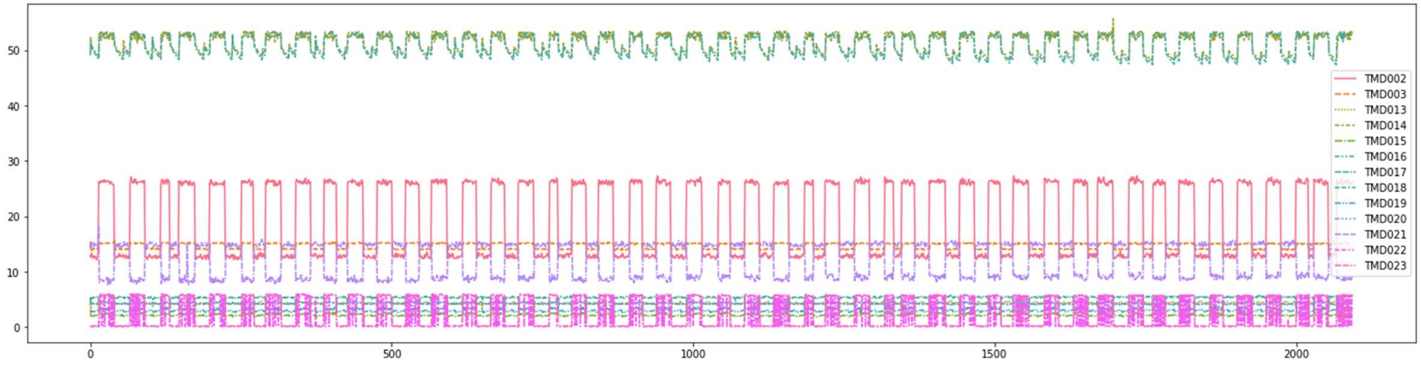
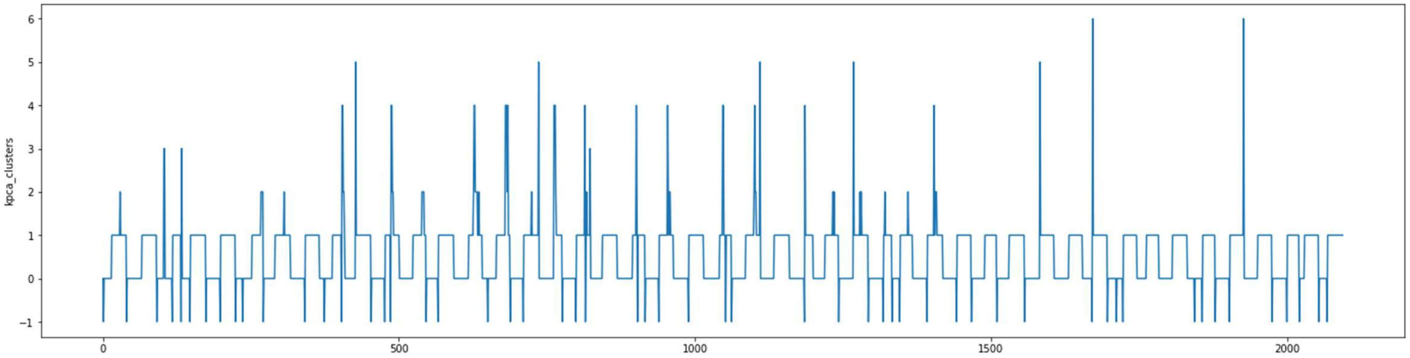
Table 5.8 – Epsilon and minPts values for the different clustering and different input data sets.

Kernel function used for dimensionality reduction	**Eps**	**MinPts**
Linear	0.7818109785966095	6
Sigmoid	0.122557620970987	
Poly	0.2169080273124639	

Source: Author's production.

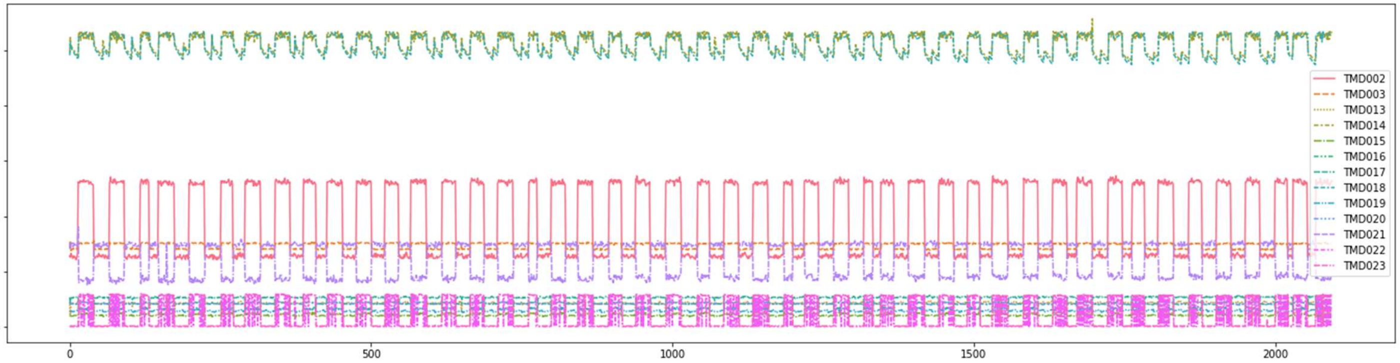
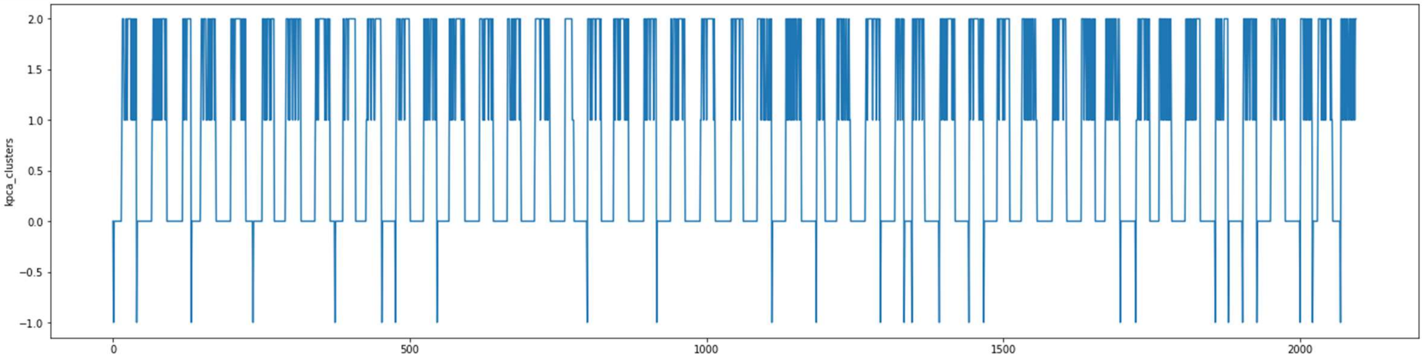
This time, there was no concern or requirements when comes to the number of clusters identified, there was also no assessment for identify which cluster would represent which behavior. The visual analysis and therefore, the assessment of the anomaly detection capabilities were made over the plot of the output of the DBSCAN as a line-plot, where the cluster-indexes are the y axis, and the observation indexes are in the x axis. The assumption made for performing such assessment was that the cyclic behavior profile observed in many telemetry signals coming from the PSS would be reflected in the clustering output, in form of different clusters for different profiles of behavior. In other words, the plot of the cluster-indexes is capable of reproduce the characteristic behavior of the satellite subsystem. Such assessment was made for all the kernel functions experimented, for the DBSCAN hyperparameters setting by elbow method and for the silhouette score method, however, since the result obtained using the elbow method was better, they are presented in Table 5.9.

Table 5.9 – Outcome of the normal behavior modeling.

Original Telemetry signal	
Poly Kernel+ DBSCAN	

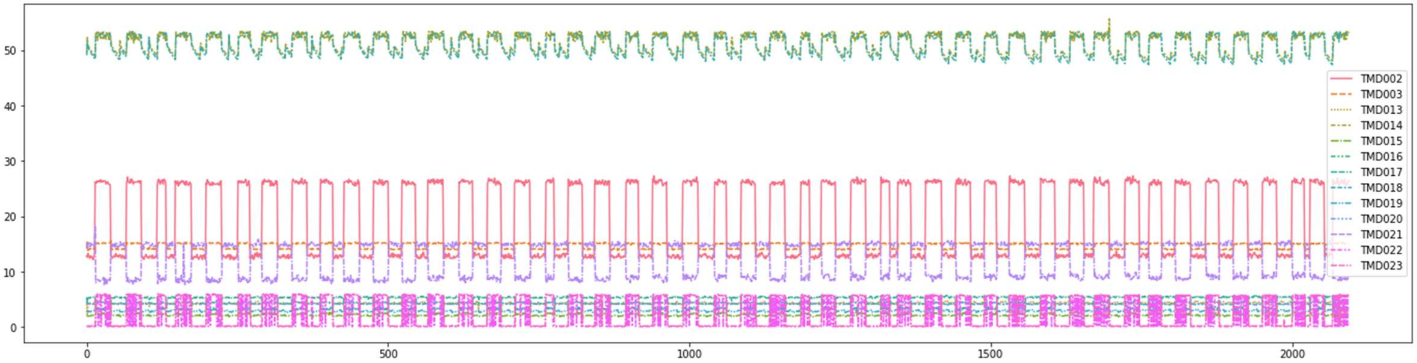
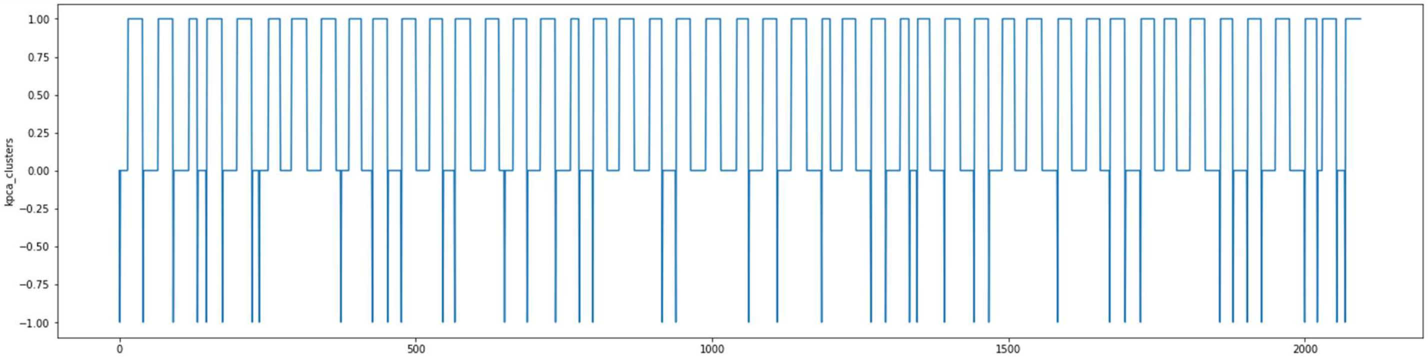
continue

Table 5.9 – Continuation.

Original Telemetry signal	
Linear Kernel+ DBSCAN	

continue

Table 5.9 – Conclusion.

Original Telemetry signal	
Sigmoid Kernel+ DBSCAN	

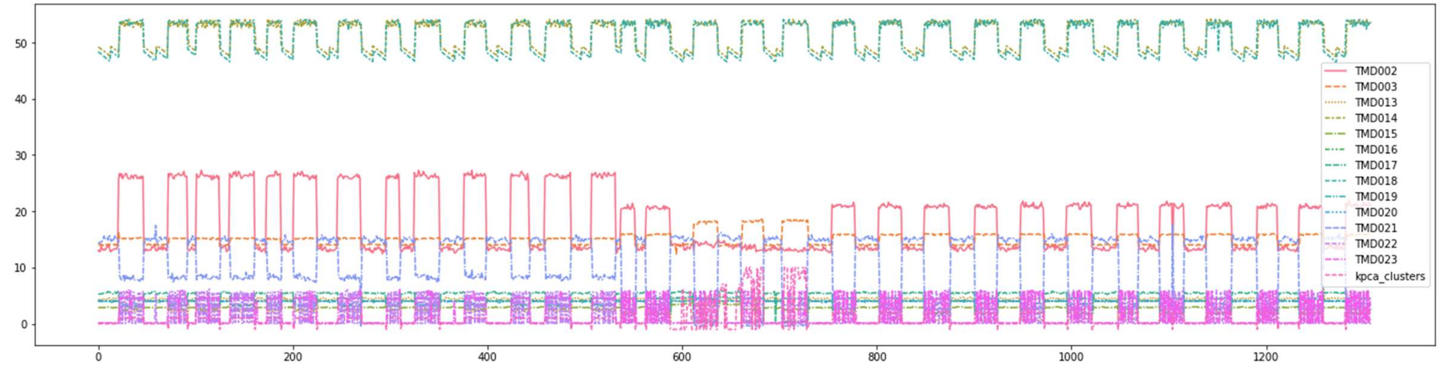
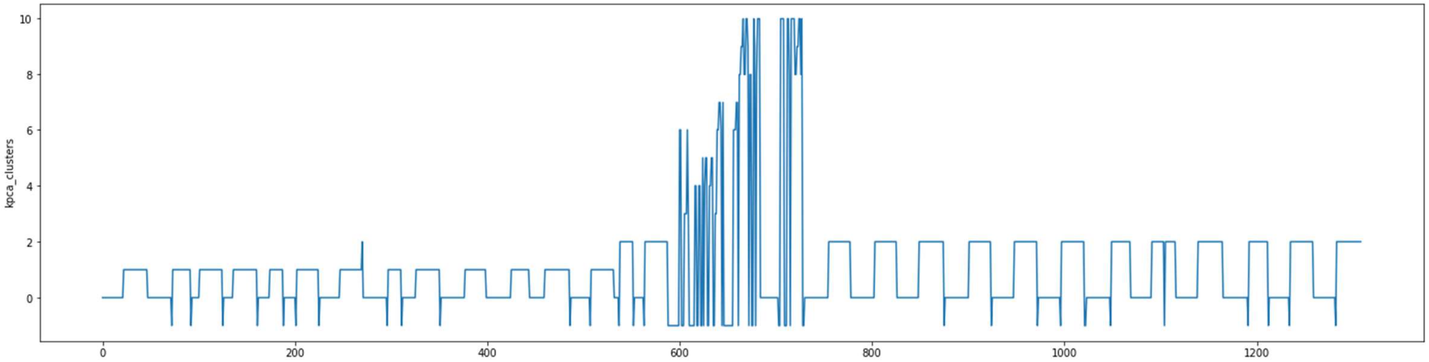
Source: Author's production.

The same dimensionality reduction algorithm drove by different kernel functions over the same input data resulted in different output data sets, these will be from now on referred as "kernel function" datasets, i.e., poly-kernel dataset. Given the difference between the kernel datasets, different eps values were found.

During the experiments with the DBSCAN algorithm, it was noticed that the algorithm is very sensitive when comes to eps value variations, so, it was expected as well that the outcome from the DBSCAN running with different eps values over different input kernel datasets would produce different outputs. This is confirmed when by the number of clusters identified in each experiment. While the poly-kenel dataset clustering resulted in 7 clusters, being one of them the cluster defined for noise or outlier observations. The sigmoid-kernel dataset clustering resulted in only 3 clusters, being one of them, the noise cluster. Such difference could also be noticed in the 3-dimensional plots in Table 5.7. However, even though the number of clusters found has a big variation among the experiments, looking at the plot of the outputs, it is possible to notice that the cyclic behavior from the satellite telemetries is presented in every each of those. As the training set used at the beginning of the process presents the considered normal satellites behavior, it can be said this outcome demonstrates that the clustering algorithm, tuned with the given hyperparameters was capable of provide an output signal modeled as the satellite's normal behavior.

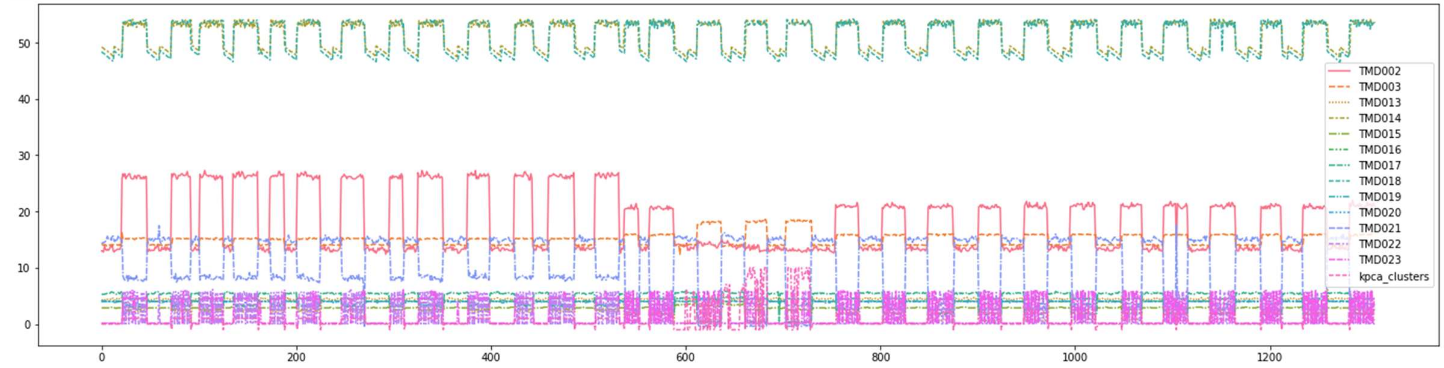
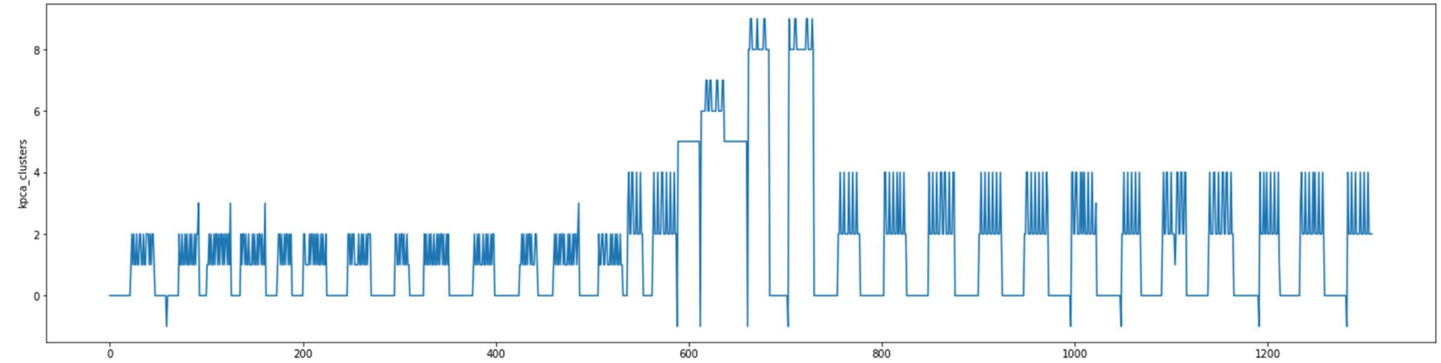
On achieving the above presented result, the next step is to validate the setup for the clustering algorithm, using the validation data set containing an "original" anomalous behavior. The outcome of this new step of the experiment is depicted in Table 5.10.

Table 5.10 - Outcome of the validation of the model against an original anomalous behavior.

Original Telemetry signal	
Poly Kernel+DBSCA N	

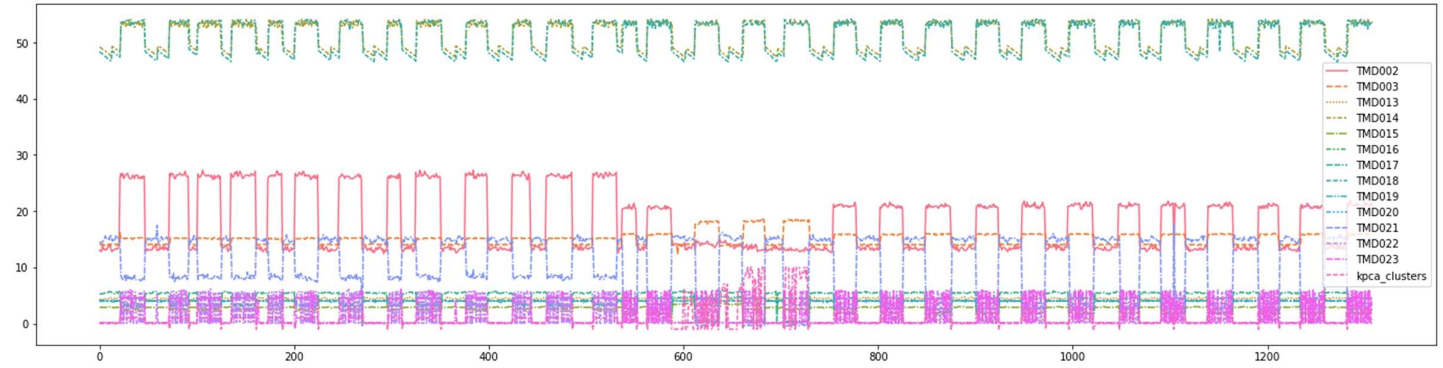
continue

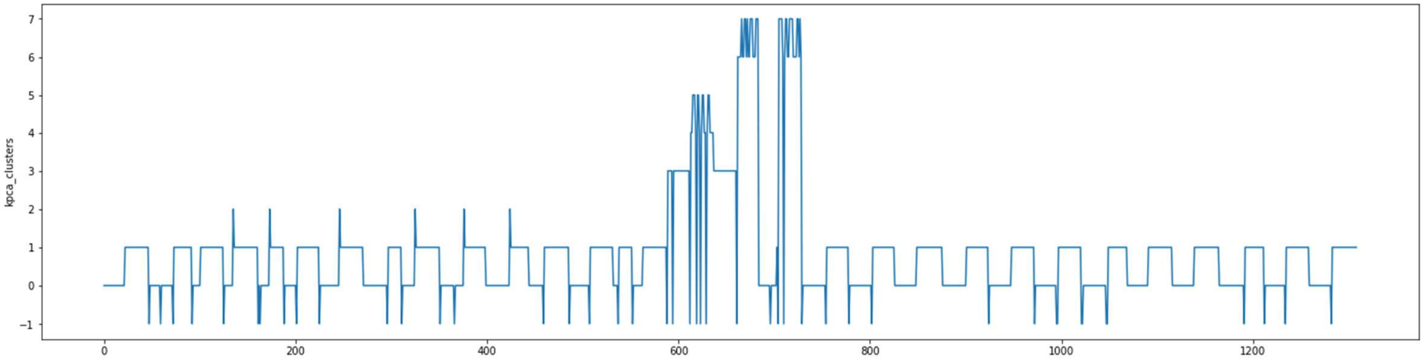
Table 5.10 – Continuation.

Original Telemetry signal	
Linear Kernel+ DBSCAN	

continue

Table 5.10 – Conclusion.

<table>
<tr>
<td>Original
Telemetry signal</td>
<td>


</td>
</tr>
<tr>
<td>Sigmoid
Kernel+
DBSCAN</td>
<td>


</td>
</tr>
</table>

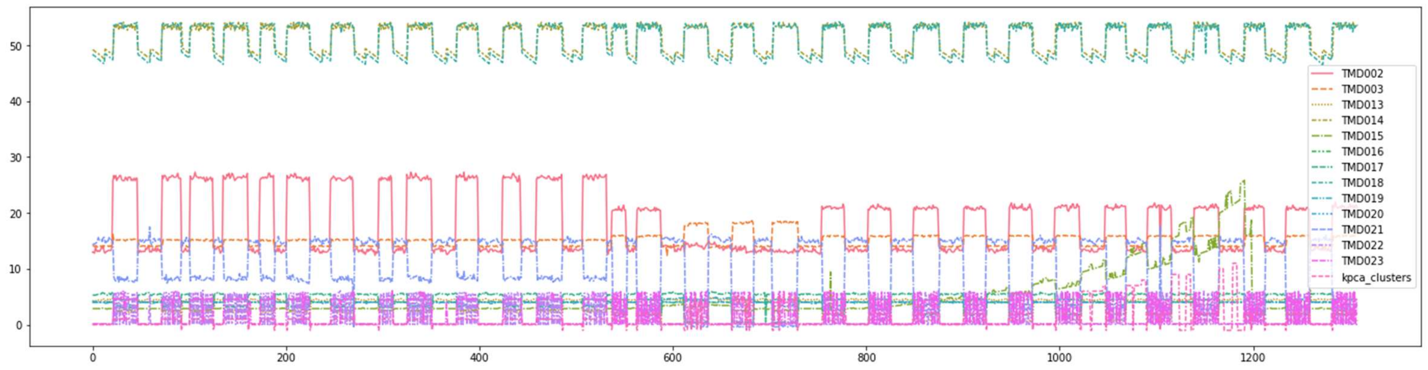
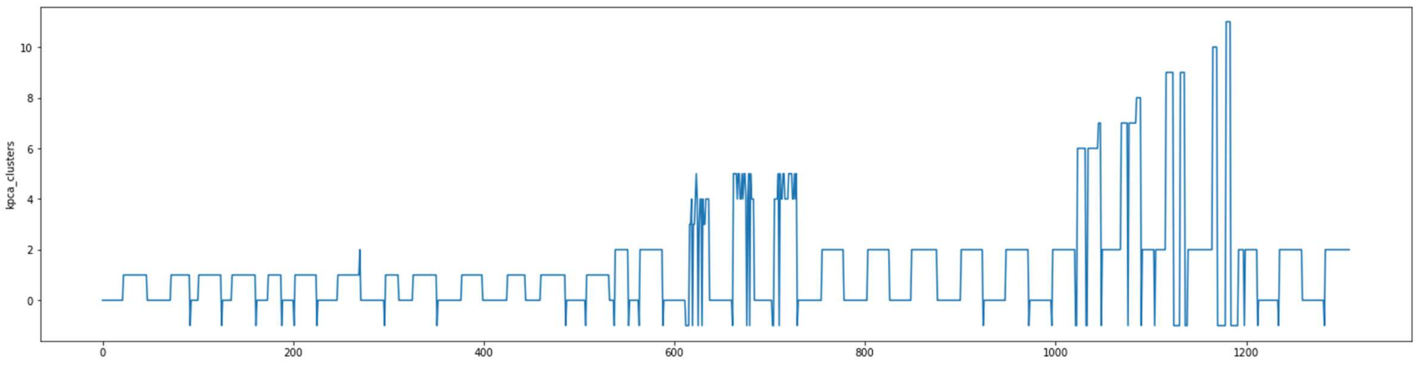
Source: Author's production.

The anomalous behavior presented in some telemetry signals from the validation dataset were detected by the DBSCAN. It is possible to identify in all the output plots depicted in Table 5.10, that in the same time window where the anomaly happens in the original dataset, the output of the clustering presents an increase in the cluster-index number, depicting where there are data points grouped together that form a cluster according to the setting; however, there are not part of cluster which groups the most of the data points. In other words, assuming that the normal behavior is the one found by the clustering in the modeling phase, the data points, which doesn't belong to the normal cluster and are assigns to higher indexes cluster, are by exclusion anomalous data points. However, the index value does not mean severity but a classified behavior or trend of the data, so the increasing of the cluster-index, making the curve gain some amplitude, means actually that, in that moment maybe another anomalous behavior was identified, pointing out the normal behavior spectrum was identified and therefore, had been assigned to a cluster.

All the results depicted in Table 5.10, obtained during the validation against the original anomaly on the telemetry, had a fairly good result. Although the poly-kernel and linear-kernel outcomes were more representative than the outcome from the sigmoid-kernel. On analyzing the original data, it is possible to identify that the amplitude of the signal suffered an attenuation, which is depicted by the two other process (poly-kernel and linear-kernel) results, when those assigned a different cluster-index for those observations after the index 800.

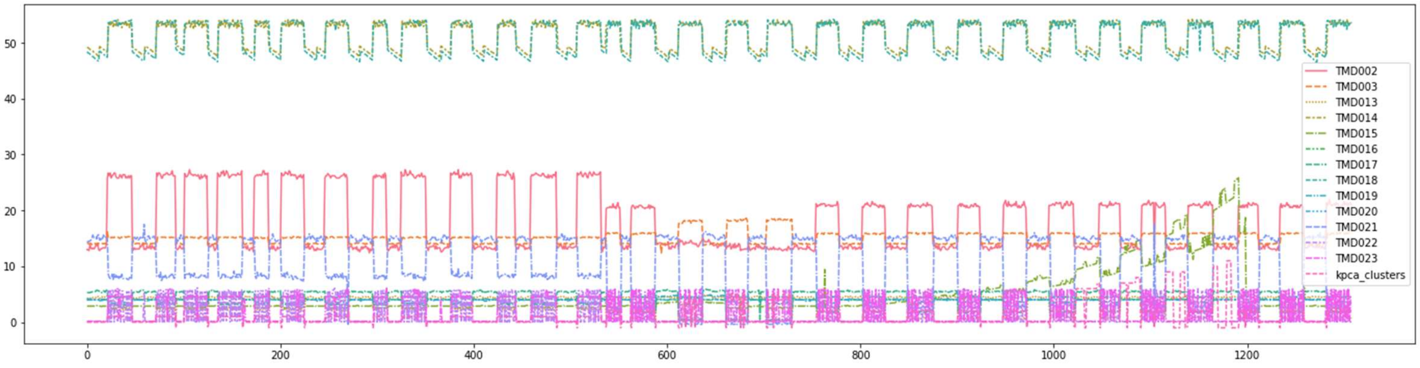
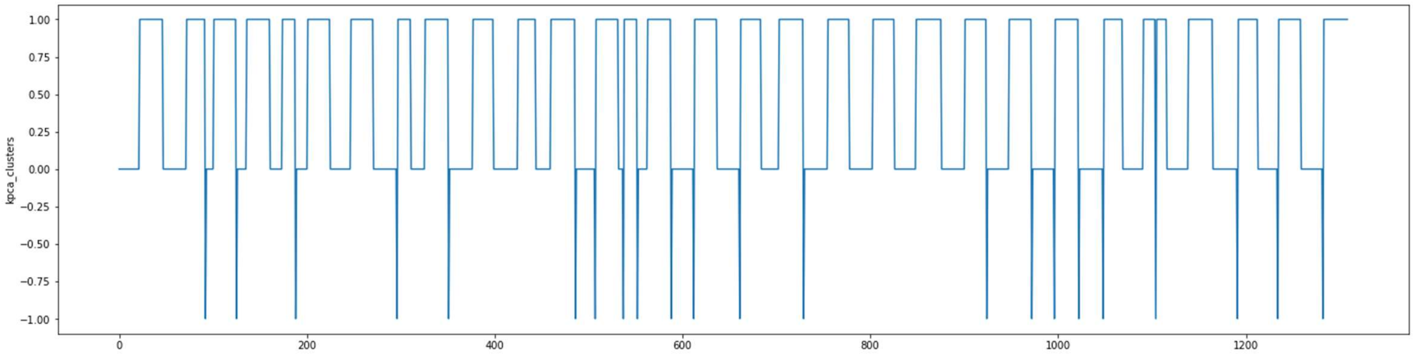
A last experiment was then performed in order to try out the obtained model against a simulated anomalous behavior injected in one of the telemetries holding the batteries temperature. This new validation set, made by the concatenation of the original validation set and the telemetry signal with the injected fault, was preprocessed by the KPCA and then by the DBSCAN algorithm. The outcome of the last experiment is depicted in Table 5.11.

Table 5.11 - Outcome from the second validation made against additional injected failure.

Original Telemetry signal	
Poly Kernel+DBSCAN	

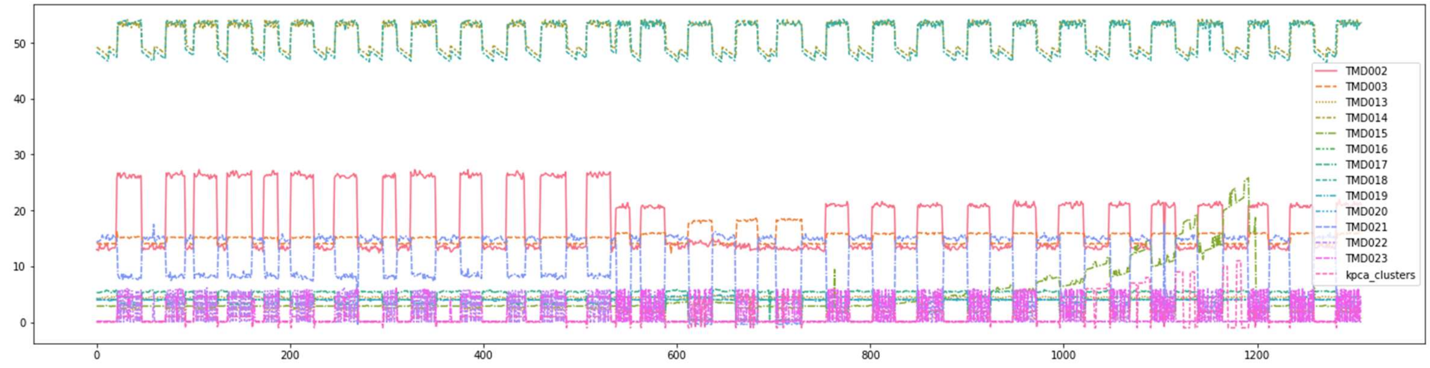
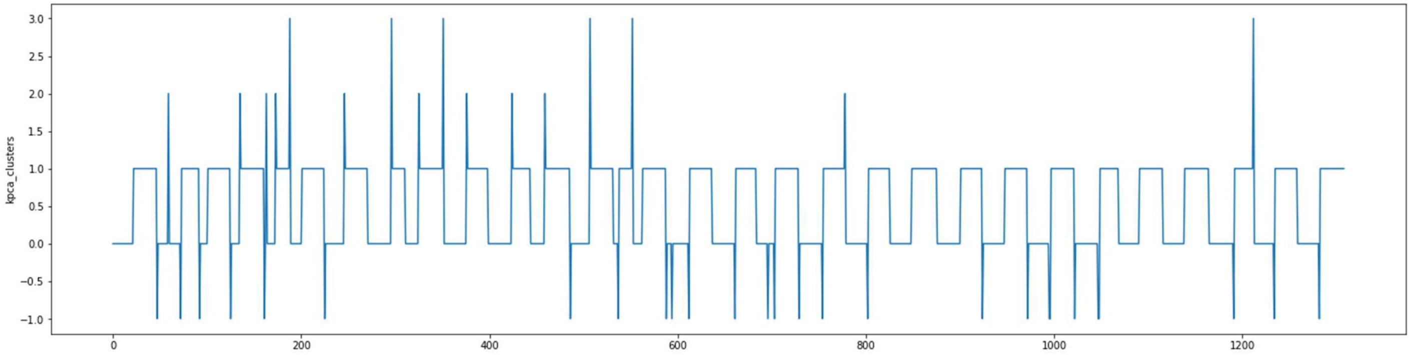
continue

Table 5.11 – Continuation.

Original Telemetry signal	
Linear Kernel+ DBSCAN	

continue

Table 5.11 – Conclusion.

Original Telemetry signal	
Sigmoid Kernel+ DBSCAN	

Source: Author's production.

This last experiment has shown that only the poly-kernel succeeded to identify the injected anomalous behavior. Matter of fact, this configuration was the only one capable of identifying all the anomalous behavior presented on to the given validation dataset. A reason that might explain such unexpected outcome from the two kernel approaches failed is shown in Table 5.7. In both cases, linear-kernel and sigmoid-kernel, there are just a few clusters identified, which means that, the DBSCAN algorithm assigned many points to the same group (cluster), this way, generating big groups that end up erroneously mimic the satellite's behavior. One of the reasons of the fail might come from the clustering algorithm was not able to identify differences in certain data. The other possible reason is that the linear and sigmoid kernel functions used with the KPCA were not adequate to well separate the input data, which translate in a hard work for the clustering and even bad clustering result.

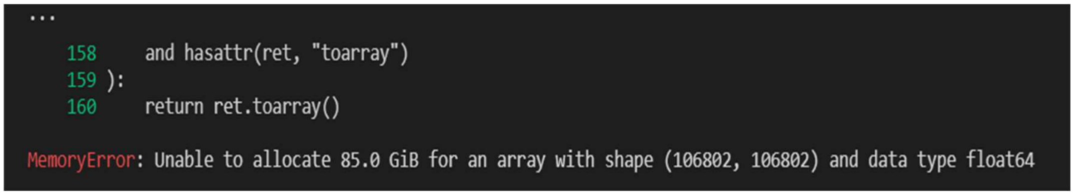
Another evidence is that there is no optimal hyperparameter that can be used for all types of datasets, as the results show that each dataset favors tuning, and this result complies with the "no-free-lunch theorem".

5.3 SCD2 case study

In the SCD2 case study, when taking the telemetry data for the Data Preparation step, the number of observations for a given period of time was absurdly high. For example, the number of observations made between 1/1/2015 and 1/7/2015 is more than 105.000 samples, against the 271 samples of a similar time window from the CBERS1 telemetry data set. This difference of around 400 times the amount of data imposes some constrains on the processing of such mass of data. KPCA algorithm is unable to manage the amount of data.

The resource constraint imposed in this situation comes from the limited computational resource available currently for running the research experiments. Personal notebooks, tends to not have a hardware specification needed to be a working station. For example, in order to perform a dimensionality reduction over an already reduced set of features from the SCD2 data set, reducing from 6 dimensions to 3 dimension, using the KPCA algorithm, in a time window of nearly one week, as depicted in the Figure 5.13, would need an amount of 85 GB memory for computing such task. Such capacity is usually not available in mid to high level personal notebooks. Figure 5.13 shows also the error message “Unable to allocate 85 GiB for an array” issued by the KPCA algorithm.

Figure 5.13 - Error message issued by the KPCA algorithm.



```
...
    158     and hasattr(ret, "toarray")
    159 ):
    160     return ret.toarray()

MemoryError: Unable to allocate 85.0 GiB for an array with shape (106802, 106802) and data type float64
```

Source: Author's production.

The memory resource request comes from an inherent characteristic of how the KPCA calculates the Kernel matrix, which is NxN matrix, being N the number of data points or observations, making the process very costly for big data sample as the one mentioned here. The data set of the SCD2 are in .CSV format and the process of loading it into the memory for performing the data science process also consumes a valuable amount of memory and sometimes the process is not possible due the size of the file.

Studies in the literature propose ways to overcome drawbacks from the KPCA algorithm through another training algorithm in order to have more computational efficiency (ZHENG; ZOU; ZHAO, 2005; WANG; HU; ZHAO, 2006), when comes to processing large data sets. However, the technical constraints regarding the algorithm’s implementation availability or their back-draws, such as the difficult of processing too big data set by KPCA algorithm, imposed limitations that were not planned to solve by this dissertation since developing or implementing tools for machine learning is out of the scope of this dissertation.

The experiments made with the data from the SCD2 satellite stopped during the machine learning model development phase due to resources constraints and technical obstacles. However, the process can be improved and modified in futures works to be capable of handle such kind of data characteristics.

6 CONCLUSIONS

The research work developed in this dissertation demonstrated that it is possible to detect anomalies in the behavior of a satellite subsystem using data-driven machine learning techniques to generate models from telemetry data. Moreover, all the results were obtained on making usage of off-the-shelf tools aiming to reduce cost with the development of such kind of tools. Another achievement was to show how a machine learning process enables improvement on the activities in a Satellite Control Central with tools to reduce operator's effort when comes to satellites telemetry data analysis.

6.1 Main contributions

The main contribution of this study was to propose a data science approach to provide support to artificial satellites operation in anomaly detection task.

Even though, the study does not cover how the data science approach will be implemented or used by the end users, it indicated that different instances of the present approach running from different machines could be capable of assess the behavior of different subsystem of the satellite, triggering alarms when the behavior of the given subsystem would not conform the normal.

For INPE’s Satellite Control Center (CCS), one of the main stakeholders from this work, the machine learning process proposed showcases an approach that can be implemented alongside other existing system to provide support and facilitate the work of those involved in satellite operations related activities.

For the academia, the proposed process adds one more brick of knowledge to the hall of anomaly detection proposed frameworks, demonstrating the capabilities and constraints achieved when using a density-based clustering algorithm associated to preprocessing methods for satellite telemetry data analysis. Furthermore, this work contributes with information and knowledge in the anomaly detection area of study for the space systems application domain.

For the correlated works, the results present pathways, possibilities and demonstrates as well possible pitfalls to be avoided in the application of certain techniques in this application domain.

Furthermore, Table 6.1 shows the summary of the published, and currently expecting to be published articles that resulted either directly from this work or from the wider master’s effort.

Table 6.1 - Resulting published work.

Name	QUALIS	SCOPUS Percentile	Status
2nd IAA LASSS 2019	Conference	-	Published
WETE 2021	Conference	-	Published
IEEE Latin America Transactions	B2	61%	Submitted
IEEE Systems Journal	A2	88%	Writing

Source: Author's production.

6.2 Future work

As future work it is recommended the performance of more tests and validation of the proposed process against other failure scenarios, in order to raise the capabilities and constraints of it. Define the use cases for it and optimize the algorithms.

Investigations on designing an architecture or a framework where the proposed process would fit the current system under usage by the operators would add value to satellite operation processes. Moreover, it would be important turn the process into a tool with a well-defined interface for users and other systems, so it would be a natural step to make this work more tangible to the CCS’s operation team.

Researches at INPE on model-based system design and model-based tests fields can combine solutions to propose a model-based failure diagnosis approach to have a complete health monitoring system for the operated satellites

REFERENCES

ALAM, A.; FUKUMIZU, K. Hyperparameter selection in kernel principal component analysis. **Journal Of Computer Science**, v. 10, n. 7, p. 1139-1150, 2014. Available from: http://dx.doi.org/10.3844/jcssp.2014.1139.1150.

ALFEILAT, H. A. A.; HASSANAT, A. B.A.; .LASASSMEH, O.; TARAWNEH, A. S.; ALHASANAT, M. B.; SALMAN, H. S. E.; PRASATH, V. B. S. Effects of distance measure choice on K-nearest neighbor classifier performance: a review. **Big Data**, v. 7, n. 4, p. 221-248, 2019. Available from: http://dx.doi.org/10.1089/big.2018.0175.

AZEVEDO, D. R. et al. Applying data mining for detecting anomalies in satellites. In: EUROPEAN DEPENDABLE COMPUTING CONFERENCE, 9., 2012, Sibiu. **Proceeding...** IEEE, 2012. p. 212-217.

BARNETT, V.; LEWIS, T. **Outliers in statistical data**. 3. ed. [S.l.]: Wiley, 1994.

BISWAS, G.; KHORASGANI, H.; STANJE, G.; DUBEY, A.; DEB, S.; HOSHAL, S. An application of data driven anomaly identification to spacecraft telemetry data. In: ANNUAL CONFERENCE OF THE PHM SOCIETY, 8., 2016. **Proceedings…** 2016. DOI: 10.36001/phmconf.2016.v8i1.2551.

BO, H. CBERS-1 power supply subsystem and in-orbit performance. In: SPACE POWER EUROPEAN CONFERENCE, 6., 2002, Porto. **Proceedings...** European Space Agency, 2002. p. 369-373.

BROWNLEE, J. **Dimensionality reduction algorithms with Python**. 2020. Available from: https://machinelearningmastery.com/dimensionality-reduction-algorithms-with-python/. Access on: 25 Aug. 2022.

BROWNLEE, J. **Distance measures for machine learning**. 2020. Available from: https://machinelearningmastery.com/distance-measures-for-machine-learning/. Access on: 12 Aug. 2022.

BROWNLEE, J. **What is a confusion matrix in machine learning**. 2016. Available from: https://machinelearningmastery.com/confusion-matrix-machine-learning/. Access on: 09 July 2022.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection. **ACM Computing Surveys**, v. 41, n. 3, p. 1-58, 2009. Available from: http://dx.doi.org/10.1145/1541880.1541882.

DHALLA, A. **The math of Principal Component Analysis (PCA)**: using two different strategies rooted in linear algebra to understand the most important formula in dimensionality reduction. Analytics Vidhya, 2021. Available from: https://medium.com/analytics-vidhya/the-math-of-principal-component-analysis-pca-bf7da48247fc. Access on: 26 Aug. 2022.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2., 1996, **Proceedings...** AAAI Press, 1996. p. 226-231. Available from: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf. Access on: 13 Aug. 2022.

FUJIMAKI, R.; YAIRI, T.; MACHIDA, K. An approach to spacecraft anomaly detection problem using Kernel feature space. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY IN DATA MINING, 11., 2005, Chicago. **Proceedings...** Chicago: ACM Digital Library, 2005. p. 401-410.

GAO, Y.; YANG, T.; XING, N.; XU, M. Fault detection and diagnosis for spacecraft using principal component analysis and support vector machines. In: IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS (ICIEA), 7., 2012. **Proeceedings…** IEEE, 2012. p. 1983-1988.

GLEN, S. **Hierarchical clustering / dendrogram: simple definition, examples**. Available from: https://www.statisticshowto.com/hierarchical-clustering/. Access on: 16 Aug. 2022.

HARRIS, N. **Visualizing DBSCAN clustering**. 2015. Available from: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/. Access on: 06 July 2022.

HODGE, V.; AUSTIN, J. A survey of outlier detection methodologies. **Artificial Intelligence Review**, v. 22, n. 2, p. 85-126, 2004. Available from: http://dx.doi.org/10.1023/b:aire.0000045502.10941.a9.

HURWITZ, J.; KIRSCH, D. **Machine learning for dummies**. Hoboken: John Wiley & Sons, 2018.

IBM. **Data science**: learn how data science can unlock business insights and accelerate digital transformation and enable data-driven decision making. 2022. Available from: https://www.ibm.com/cloud/learn/data-science-introduction. Access on: 01 Sept. 2022.

IBRAHIM, S. K. M. **Spacecraft performance analysis and fault diagnosis using telemetry-mining**. 2018. 132 f. Thesis (Master in Engineering, Science in Computer and Systems) - Zagazig University, Zagazig, 2018.

IBRAHIM, S. K. et al. Machine learning techniques for satellite fault diagnosis. **Ain Shams Engineering Journal**, p. 45-54., 2019.
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **CBERS-1, 2 e 2B**. Available from: http://www.cbers.inpe.br/sobre/cbers1-2-2b.php. Access on: 11 July 2022.

JOLLIFFE, I. T.; CADIMA, J. Principal component analysis: a review and recent developments. **Philosophical Transactions of the Royal Society A**: **Mathematical, Physical and Engineering Sciences**, v. 374, n. 2065, e 20150202, 2016. Available from: http://dx.doi.org/10.1098/rsta.2015.0202.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding groups in data**: an introduction to cluster analysis. Brussels: Wiley-Interscience, 1990.

KUBAT, M. **An introduction to machine learning**. 2.ed. [S.l.]: Springer, 2017.

LI, W.; PENG, M.; LIU, Y.; JIANG, N.; WANG, H.; DUAN, Z. Fault detection, identification and reconstruction of sensors in nuclear power plant with optimized PCA method. **Annals of Nuclear Energy**, v. 113, p. 105-117, 2018. Available from: http://dx.doi.org/10.1016/j.anucene.2017.11.009.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. 2.ed. Cambridge: The MIT Press, 2018.

MIGUEZ, R. R. B.; SILVA, M. M. Q.; KONO, J. **SCD2 operation handbook**. [S.l.: s.n.], 1993.

MITAL, R.; CATES, K.; COUGHLIN, J.; GANJI, G. A machine learning approach to modeling satellite behavior. In: IEEE INTERNATIONAL CONFERENCE ON SPACE MISSION CHALLENGES FOR INFORMATION TECHNOLOGY, 2019. **Proceedings…** IEEE, 2019. p. 62-69. Available from: http://dx.doi.org/10.1109/smc-it.2019.00013.

MOLCHANOV, V.; LINSEN, L. Overcoming the curse of dimensionality when clustering multivariate volume data. In: INTERNATIONAL CONFERENCE ON INFORMATION VISUALIZATION THEORY AND APPLICATIONS, 13., 2018, Madeira. **Proceedings...** Scitepress, 2018. v. 3, p. 29-39.

OLIVEIRA, F. **O Brasil chega ao espaço: SCD 1 Satelite de Coleta de Dados**. São Paulo: [s.n.], 1996. 972 p. Available from: http://urlib.net/rep/6qtX3pFwXQZ3r59YCT/GUJxJ.

ORLANDO, V.; KUGA, H. K. (Org.). **Os satélites SCD1 e SCD2 da missão espacial completa brasileira - MECB**. Available from: http://200.144.244.96/cda/oba/aeb/a-conquista-do-espaco/Capitulo-5.pdf. Access on: 10 July 2022.

PEDREGOSA, F. *et al*. Scikit-learn: machine learning in Python. **Journal of Machine Learning Research**, p. 2825-2830, 2011. Available from: https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf. Access on: 07 July 2022.

PURARJOMANDLANGRUDI, A.; GHAPANCHI, A. H.; ESMALIFALAK, M. A data mining approach for fault diagnosis: An application of anomaly detection algorithm. **Measurement**, p. 343-352, 2014.

RAHMAH, N.; SITANGGANG, I. S. Determination of optimal Epsilon (Eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra. **Earth and Environmental Science**, v. 31, e 012012, 2016. Available from: http://dx.doi.org/10.1088/1755-1315/31/1/012012.

RASCHKA, S. **Kernel tricks and nonlinear dimensionality reduction via RBF kernel PCA.** 2014. Available from: https://sebastianraschka.com/Articles/2014_kernel_pca.html. Access on: 30 Aug. 2022.

RASYID, L. A. *et al*. Review on clustering algorithms based on data type: towards the method for data combined of numeric-fuzzy linguistics. **Journal of Physics: Conference Series**, p. 1-10, 2018. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/1097/1/012082. Access on: 07 May 2022.

RIBEIRO, E. M S. **A1 – análise de componentes principais**. Available from: https://edisciplinas.usp.br/pluginfile.php/3381446/mod_resource/content/2/03_Componentes_Principais.pdf.

RODRIGUES, I. P. *et al*. Modeling satellite battery aging for an operational satellite simulator. **Advances in Space Research**, v.67, n. 6, p. 1981-1999, 2021.

ROSENTHAL, G.; ROSENTHAL, J. **Statistics and data interpretation for social work**. [S.l.]: Springer, 2011

SAFIZADEH, M. S.; LATIFI, S. K. Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. **Information Fusion**, v. 18, p. 1-8, 2014. Available from: http://dx.doi.org/10.1016/j.inffus.2013.10.002.

SATOPAA, V.; ALBRECHT, J.; IRWIN, D.; RAGHAVAN, B. Finding a "Kneedle" in a Haystack: detecting knee points in system behavior. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS WORKSHOPS, 31., 2011. **Proceedings…** IEEE, 2011. p. 166-171. Available from: http://dx.doi.org/10.1109/icdcsw.2011.20.

SCHÖLKOPF, B. *et al*. Kernel principal component analysis. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS, 7., 1997, Lausanne. **Proceedings...** Springer, 1997. p. 583-588. Available from: https://people.eecs.berkeley.edu/~wainwrig/stat241b/scholkopf_kernel.pdf. Access on: 06 July 2022.

SCHÖLKOPF, B.; SMOLA, A. J. **Learning with Kernels**: support vector machines, regularization, optimization, and beyond. Cambridge: The Mit Press, 2002.

SCHUTT, R.; O'NEIL, C. **Doing data science**. Sebastopol: O'reilly Media, 2013.

SCIKIT-LEARN. **Nearest neighbors**: nearest neighbors classification. Available from: https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbor-algorithms. Access on: 13 Aug. 2022.

SCIKIT LEARN. **Sklearn preprocessing standardscaler**. 2022. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Access on: 07 July 2022.

SHAWE-TAYLOR, J.; CRISTIANINI, N. **Kernel methods for pattern analysis**. Cambridge: Cambridge University Press, 2004.

SKIENA, S. S. **The data science design manual**. [S.l.]: Springer Cham, 2017.

SOLBERG, H. E.; LAHTI, A. Detection of outliers in reference distributions: performance of horns algorithm. **Clinical Chemistry**, v. 51, n. 12, p. 2326-2332, 2005. Available from: http://dx.doi.org/10.1373/clinchem.2005.058339.

SOUZA, P. B. **Uma estratégia baseada em algoritmos de mineração de dados para validar plano de operação de voo a partir de predições de estados dos satélites do INPE**. 2011. 171p. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011. Available from: http://urlib.net/ibi/8JMKD3MGP7W/39GL532.

TABUROğLU, S. A survey on anomaly detection and diagnosis problem in the space system operation. **Journal of Intelligent Systems: Theory and Applications**, v. 2., p. 13-17., 2019.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. [S.l.]: Pearson Education Limited, 2005.

TUKEY, J. W. **Exploratory data analysis**. Reading: Addison-Wesley, 1977.

WANG, H. *et al*. **Fault identification and diagnosis based on KPCA and similarity clustering for nuclear power plants**. Elsevier: Annals of Nuclear Energy, 2021.

WANG, H.; HU, Z.; ZHAO, Y. Kernel principal component analysis for large scale data set. In: HUANG, D. S.; LI, K.; IRWIN, G. W. (Ed.). **Intelligent computing**. Berlin: Springer, 2006. p. 745-756. Available from: http://dx.doi.org/10.1007/11816157_91.

WARNER, R. **Applied statistics: from bivariate through multivariate techniques**. [S.l.]: SAGE, 2013.

WICKHAM, H. Tidy data. **Journal of Statistical Software**, v. 59, n. 10, p. 1-23, 2014. Available from: http://dx.doi.org/10.18637/jss.v059.i10.

WERTZ, J.R.; EVERETT, D. F.; PUSCHELL, J.J. Space Mission Engineering: The New SMAD, Microcosm Press, 2011.

WERTZ, J.; LARSON, W. **Space mission analysis and design**. [S.l.]: Springer, 1992.

YAIRI, T.; TAKEISHI, N.; ODA, T.; NAKAJIMA, Y.; NISHIMURA, N.; TAKATA, N. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. **IEEE Transactions on Aerospace and Electronic Systems**, v. 53, n. 3, p. 1384-1401, 2017. Available from: http://dx.doi.org/10.1109/taes.2017.2671247.

YANG, T.; CHEN, B.; GAO, Y.; FENG, J.; ZHANG, H.; WANG, X. Data mining-based fault detection and prediction methods for in-orbit satellite. In: INTERNATIONAL CONFERENCE ON MEASUREMENT, INFORMATION AND CONTROL, 2., 2013, Harbin. **Proceedings...** IEEE, 2013. p. 805-808.

ZHENG, W.; ZOU, C.; ZHAO, L. An improved algorithm for Kernel principal component analysis. **Neural Processing Letters**, v. 22, n. 1, p. 49-56, 2005. Available from: http://dx.doi.org/10.1007/s11063-004-0036-x.

ZHANG, L. **Big data analytics for fault detection and its application in maintenance**. 2016. 148 f. Thesis (PhD in Operation and Maintenance Engineering) - Luleå University of Technology, Luleå, 2016.

ZARE, S. **Fault detection and diagnosis of electric drives using intelligent machine learning approaches.** 2018. Available from: https://scholar.uwindsor.ca/etd/7436

APPENDIX A – USED LIBRARIES

This study made usage from the following open-source python libraries:

- General data manipulation
    - Numpy;
    - Pandas;
    - Scipy;
    - Random, and;
    - Embedded python functions.
- Plots and graphs generation
    - Matplotlib;
    - Seaborn;
    - Plotly.
- Data science processing and algorithm
    - ScikitLearn, and following modules:
        - DBSCAN from cluster
        - NearestNeighbors and KNeighborsClassifier from neighbors
        - Confusion_matrix and silhouette_score from metrics
        - Train_test_split from model_selection
        - StandardScaler from preprocessing, and
        - PCA and KernelPCA from decomposition